参考資料５

経済産業省委託調査

平成１６年度ＥＣ技術基盤の相互運用性に関する調査研究

（取引相手先の属性認証技術等の調査）

属性認証ハンドブック

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

この報告書は、平成１６年度受託事業として（財）日本情報処理開発協会電子商取引推進センターが経済産業省から委託を受けて、電子商取引推進協議会（ECOM）の協力を得て実施した「平成１６年度ＥＣ技術基盤の相互運用性に関する調査研究（取引相手先の属性認証技術等の調査）」の成果を取りまとめたものです。

序文

本ハンドブックは、電子認証における資格や権限の情報として利用される属性情報について、これまで認証公証ワーキンググループで行ってきた、実社会での利用場面の分析や利用者からの要求要件の整理等利用する側からの検討と、属性証明書の活用やSAMLによる属性認証の実現方式等の技術的な検討の結果を踏まえて、今年度は合同の検討チームを組んで関連する情報の調査及び整理を行い、ハンドブックとしてまとめたものである。

すでに、公開鍵証明書を用いることによりインターネット上で本人確認を行う仕組みができている。しかしながら、公開鍵証明書により本人性は証明されるものの、その本人の資格や権限の情報については統一的な記述方法はなく、多くの場合、公開鍵証明書の拡張領域に利用するサービス毎にそれぞれの形式で属性情報を記述する方法が取られ、相互運用性が課題として残っている。

この属性情報の利用に関して、公開鍵証明書とは別に属性証明書を発行する方法、信頼性の高いデータベースに属性情報を登録管理し公開鍵証明書にリンクさせておくことにより属性情報を取り出す方法などが検討されている。

このハンドブックは、属性情報の特性、実現方式の特長、利用者からの要求などの情報を総合的に検討して、属性情報を活用するシステムの構築の際の参考になることを目的に作成されたものである。

本報告書が、電子署名の利用を検討している企業、機関の方々にとって一助になることができれば幸いである。

平成17年2月

財団法人日本情報処理開発協会
電子商取引推進センター
電子商取引推進協議会

目　次

1. まえがき

個人の属性情報とは、個人に関連する様々な情報である。ネットワークビジネスのアプリケーションは、個人の識別に加えて、そのアプリケーションが利用する幾つかの属性を必要とする。例えばネットワークでのショッピングでは、支払いのためのクレジットカード番号や商品の送付のために氏名や住所などの属性情報を与えなければ商取引が成立しないことがある。ネットワークアプリケーションは属性情報を有効に活用することでビジネスを遂行できる。しかし、これらの属性情報は場合によっては極めてセンシティブで保護されなければならない情報であって、とりわけインターネットで行うトランザクションのセキュリティが重要である。

本ハンドブックは、属性情報を活用するにあたって考慮すべき本人の認証方式や属性情報の扱いについて、技術の現状や標準化の動向および実例をふまえて出来るだけ広く概観するものである。また本ハンドブックは、個人情報の保護が強く求められる時代にあって、ネットワークアプリケーションの技術者や企画者や管理者に対して、認証や属性情報を扱う技術について、どの技術がどのようなセキュリティの特徴を持っているのか、あるアプリケーションにはどのようなセキュリティ対策が必要かのガイドを与えるものとして企画された。

属性情報を扱うためには、まず本人の認証(Authentication)が必要である。ネットワークビジネスの初めに認証在りきである。属性情報の活用は安全な認証が完了して始まる。インターネットの活用はいまやネットワークビジネスに欠かせないものとなっている。しかし、そこではセキュリティ情報の盗聴、漏洩、改ざんなどの脅威が蔓延しており、この脅威からの保護が必須である。ネットワークに無防備にID、パスワードが流れることの危険性が叫ばれているが、セキュリティが弱いとされるID、パスワードの世界からネットワークビジネスは未だに抜けだせないでいる。安全なネットワークビジネスのためには、より強いセキュリティ認証技術が求められているのである。本ハンドブックではこのような視点から、現在使われている各種の認証技術についてまず概観し、その上で属性情報をどのように扱うのかを見ることにする。

本ハンドブックの2章では、アイデンティティ認証として、現在多くの関心事である各種バイオメトリック認証技術を概観し、さらにより強い認証技術に向けたワンタイムパスワード認証、共通鍵暗号技術を用いたケルベロス認証技術、強力なセキュリティを与えるPKI 認証技術を述べる。

3章では、属性認証として、属性の定義や考え方を述べ、属性を証明する方法としてPKI 証明書に記載する方法、X.509 標準に定められた属性証明書を用いる方法、また従来から行われてきた属性データベースによる方法をのべる。属性データベースによる属性情報の扱いは、今まで各ベンダによる独自の実装方法がとられてきた。最近OASIS で標準化されたSAML は分散したWebシステムへのシングルサインオンや属性の証明、これらの情報を元にポリシーに従ったアクセス制御を行う仕様として注目を集めている。さらにSAML をベースに個人情報保護の観点から分散個

人情報を連携させる Liberty の技術についても注目されている。ここではこの SAML や Liberty の解説を行った。また、属性を使って認可を行うためのポリシー記述言語である XACML についても述べる。

4 章では、海外動向として現在米国政府が企画している、電子政府のなかで各政府機関が行うサービスをシングルサインオンで利用するための認証(Authentication)基盤について述べる。これは各政府機関が e-Authentication イニシアチブとして推進しているものである。この基盤では、属性の扱いは各サービスに任されるとして基盤としては直接扱わないが、セキュリティの最初の関門としての認証を SAML によるシングルサインオンで実現しようとしている。

またこの章では、標準化の推進に係っている各種標準化団体についても紹介した。

付録では、これらの認証や属性の証明のためのセキュリティのベースとして用いられる XML 署名について解説しておく。PKI は強いセキュリティを提供するが、公開鍵の検証が難しい。この公開鍵の検証をサーバで行い、PKI アプリケーションから公開鍵の検証を開放する XKMS についても注目すべき技術として解説した。

本ハンドブックは、各章、各節のそれぞれの内容が相互に独立した記述になっているので、読者は章立ての順序に関係なく、興味ある部分を見ていただければよい構成となっている。

2. アイデンティティ認証

アイデンティティ認証の技術は大きく分けると次の 3 つに分類できる。

・本人の生体情報に基づくもの

顔、虹彩、指紋、網膜、静脈紋、DNA など、不所持は生じない方式。体調、怪我など本人の状態が変わることで、本人であるにも係らず本人が拒否される場合もある。他の認証方法と違い、判定基準が統計的手法に基づいている。

・本人の記憶に基づくもの

本人が記憶している暗証番号／PIN、パスワード、パスフレーズなどが該当する。本人がそれらを忘却した場合や、入力ミスの場合には本人拒絶される。また、他人に容易に類推されるデータを利用した場合や卓上にメモを貼り付けた場合に他人により成済ましされる可能性があるため、利用者への暗証番号などの管理の徹底などを図る必要がある。

・本人が所持しているものに基づくもの

鍵、IC カード、ハードトークンなど、本人が所持している物により本人を認証する方式。他人による所持物の盗用によりなりすましが出来るため、利用者による所持物の管理が重要である。

(公開鍵方式の秘密鍵は IC カード等演算機能をもった物に格納して利用する場合が一般的なため、本人の「記憶に基づく」ものではなく、ここでは「所持するものに基づく」ものに分類する)

上記の認証方式の実例を表 2-1 に示す。以下の各章でこれらの主なものを紹介してゆく。

表 2-1 認証方式例

項番	分類	認証方式例
1	生体情報	生体認証
2	記憶	ID／パスワード認証
3		ケルベロス認証
4		ワンタイムパスワード(カウンタ同期型)
5	所持	ワンタイムパスワード(時刻同期型)
6		PKI による認証

各方式毎に得失があるため、実際には上記の各方式を組み合わせて利用することが一般的である。例えば、IC カードを利用する場合には毎回 PIN の入力を求めるなどの利用方法がある。

以下、各認証方式について解説してゆく。なお、上記では自然人に対する認証を中心に分類しているが、認証の対象としては「機器」や「メッセージ」などについて行う場合もあるため、これらについても示すこととする。

2.1　生体認証

生体認証とは身体認証またはバイオメトリクス認証とも呼ばれ、人の個体毎に異なる生体情報(特徴・特性)に基づく認証である。生体情報は、要件として「普遍性」(誰もが所持)、「唯一性」(同一所持者なし)、「永続性」(経年変化なし)が求められる。また指紋や虹彩など身体的特徴と、音声や署名など行動的特徴に分類される。

生体認証の特徴としては、生体情報は体調や天候などにより常に変化し登録時と同じ状態にはならないため、判定基準が統計的手法に基づく点が挙げられる(DNA 認証を除く)。

判定基準としては、「本人拒否率」と「他人受入率」の兼合いにより認証する。

また標準化に関しては、ローカル環境でのクローズな利用から、公的利用など広域におけるオープン利用に向け、互換性や連携を取るアプリケーションインタフェースや評価基準の統一を図る方向で進められている。[bio01]

2.1.1　生体認証の種類

生体認証に用いられている生体情報には、次のものがある。

＜身体的特徴＞

指紋	・・・	指紋の特徴点(分岐・端点など)で判別
掌形	・・・	掌(てのひら)の大きさ、形状などで判別
静脈	・・・	指・掌の静脈の形状で判別
虹彩	・・・	目の虹彩(アイリス)の紋様で判別
顔	・・・	顔の輪郭、目・鼻の位置で班別
網膜	・・・	眼底の網膜の形状で判別
DNA	・・・	人体の設計図である DNA で判別

＜行動的特徴＞

音声	・・・	音声波形、発声速度などで判別
筆跡	・・・	署名の形状、筆順、筆の運び方、筆圧などで判別

■指紋(Fingerprint)

指紋隆線の特徴点等を用いて認証する。長年の実績があり、入退出管理やパソコンや携帯電話の認証など、最も普及している。

機器も各社から様々な物が出されており、認証操作は機器に指を載せるだけで、照合時間も短く、安全性も高い。短所としては、犯罪捜査利用などの経緯より指紋登録に抵抗感を持つ人が多い点。

情報対象：両手の 10 指

エラー要因：気候(乾燥)、年齢、汚れ、傷など

＜適用事例＞

- ビル・マンションの入退出管理
- パソコン・携帯電話の組込みによるアクセス管理

■掌形(Handgeometry)

親指を除いた 4 指の長さ、掌の幅や厚さ、指の関節の幅と高さ。

登録および認証の操作性が容易。

情報対象：2 箇所(左右の掌)

エラー要因：怪我、年齢など

＜適用事例＞

- 米国の主要空港における入国審査のための無人入国審査
- 大学の食堂

■静脈(Veincheck)

指・掌の静脈形状のパターンを特徴点として認証する比較的新しい技術。他人受け入れ率が低く、登録および認証の操作性が容易で非接触型の機器も開発されている。ただし新しい技術のため、精度など未知の部分が有る。

情報対象：10 指、または両方の掌

＜適用事例＞

- 銀行 ATM
- 入退出管理

■虹彩(Iris)

黒目の内側で瞳孔より外側のドーナツ状の筋肉組織の薄膜の紋様である虹彩を用いて認証する。虹彩は生後2年程度で成長が止まり、指紋同様に個人特有のパターンとなる。

他人受け入れ率が低く、非接触での認識が可能。

情報対象：両眼

エラー要因：外部照明

＜適用事例＞
- 米国・カナダの入国審査
- 入退出管理

■顔(Face)

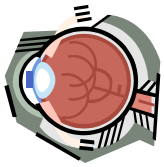
顔の輪郭、目・鼻・口など顔器官の位置・特徴点・立体形状を用いて認証する。人同士が普段行っている認証に近いため心理的抵抗が少なく、親しみやすい認証方式。またカメラを利用するため、他の認証と比較して距離が有っても認証可能。

情報対象：1箇所

エラー要因：証明、年齢、メガネ、マスク、一卵性双生児

＜適用事例＞
- 韓国・オーストラリアの入国審査
- 携帯電話の本人認証

■網膜(Retina)



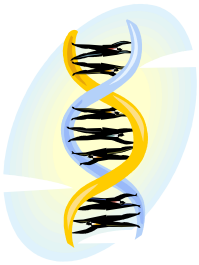
網膜上の毛細血管のパターンを用いて認証する。指紋同様に個人特有のパターンとなる。眼の中に保持されているため不安定要素も低い。

情報対象：2箇所

エラー要因：メガネ

＜適用事例＞
- 銀行ATM
- 入退出管理

■DNA(DNA)



人体の設計図ともいわれている DNA 遺伝子を用いて認証する。人間の DNA は約 30 億個の塩基配列から成り個人により異なる部分が有る。

DNA はデジタル情報のため同値認証が可能で、他のアナログ情報を元にした認証と異なり照合アルゴリズムが不要で認証精度も高い。

しかし現状では、抽出・分析に要する時間と機器費用が課題となり認証用途としては普及しておらず、ブランド商品やグッズの真贋識別のための DNA 認証マークなどに利用されている。

情報対象：複数(粘膜、毛髪など)

＜適用事例＞認証以外での利用
- DNA インキによる、有名人のサイン、ブランド商品やグッズの真贋識別
- 犯罪捜査

■音声(Voi cepri nt)

個人の音声波形や発生速度による声紋の相違を用いて認証する。標準的なパソコンが有れば、音声認証処理が可能なため特別の認識装置は不要。また電話などを用いた遠隔での認証も可能で、心理的な負担が一番軽いと言われている。

情報対象：音声

エラー要因：体調、変声期、外部の雑音

＜適用事例＞
- テレホンバンキングにおける認証

■筆跡(Si gnat ure)

記述する場合の筆記運動を用いて認証する。書体の形状に加え、筆順、筆の運び方、筆圧などを総合的に判別する。日本よりもサイン文化である欧米での利用実績が高い。

情報対象：利き手(1 箇所)

エラー要因：利き手の怪我、字体の変化

＜適用事例＞
- タブレット PC や PDA などの機器認証
- 入退出管理

2.1.2　生体認証の特徴

■判定基準は統計的手法による

生体情報はDNAを除き体調や天候などに左右されるため、登録データとの同値による判定は難しく、統計的手法による判定を用いる。判定には「本人拒否率」(FRP)と「他人受入率」(FAR)を用い、閾値の調整により認証精度を調整する。他の認証と異なり、100%の一致は逆に疑わしいと判定する。

本人拒否率	FRR：False Rejection Rate 本人の生体情報を不一致と認識する確率
他人受入率	FAR：False Acceptance Rate 他人の生体情報を一致と認識する確率

本人拒否率と他人受入率は相反する関係にあり、他人による「なりすまし」を厳しく防ごうと「他人受入率」を小さくすると「本人拒否率」が大きくなり、本来合うべき本人も拒否される傾向が有る。各種生体認証の精度やコストの比較について表 2-2 に示す。

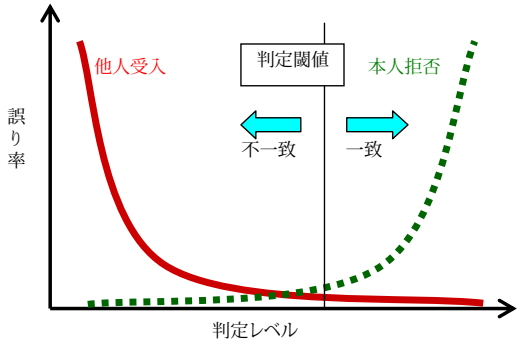


図 2-1 判定基準の概念図

表 2-2 認証技術の比較

生体情報	精度	安全性	受容性	コスト	その他
指紋	○	○	△	◎	真皮検証による生体確認
掌形	○	△	△	○	
静脈	○	◎	○	○	
虹彩	◎	◎	△	△	
顔	△	△	◎	○	三次元解析による精度向上
網膜	◎	○	△	△	
ＤＮＡ	◎	○	△	△	
音声	△	△	◎	◎	
筆跡	△	△	◎	◎	

◎：良い　○：普通　△：低い
安全性：偽造のしにくさ
受容性：馴染み易さ(受け入れ易さ)

評価基準は、文献[bio02]～[bio05]を参考に一般的な判断に基づいた。

■盗まれた(偽造された)場合について

生体情報はパスワードのように容易に変更ができない。そのため盗用された場合は、指紋認証の場合は人差し指から薬指に変えるなど、認証対象の部位を変更するなどの対応となる。

また生体認証技術の安全面での評価に向け、偽造に関する研究も進められている[bio08]。

表 2-3 偽造に関する技術

認証方式	偽造技術	内容
指紋	グミ指など	ゼラチンを用いて偽造した人工指紋を使用。ガラスなどに付着した遺留指紋からの偽造も可能
虹彩	印刷画像など	登録時に表示される画像を印刷し、瞳孔部分だけをくりぬいた紙を使用 (画像は登録時の一瞬に表示されるだけなので、盗用は難しい)

■IC カードと組合せた利用について[bio09]

IC カードは生体認証との相性が良いと言われており、今後の普及が期待される。利用形態としては IC カードのセキュア領域に生体情報を格納し照合に利用する。照合方式には下表に示す幾つかの方式がある。

表 2-4 IC カードを利用する場合の各種方式

方式	生体情報入力	照合ロジック	概要	生体情報の安全性
All On Card (AOC)	IC カード	IC カード	入力、照合の全てをカード上で実施(マスターの生体情報はカードから出ない)	高 ↕ 低
Match On Card (MOC)	入力装置	IC カード	照合についてカード上で実施(マスターの生体情報はカードから出ない)	
Store On Card (SOC)	入力装置	PC 等	マスターの生体情報を認証装置または PC に取出して照合実施	

2.1.3 課題と標準化

これまでは重要施設における入退出管理など特殊用途に応じたクローズなシステムを、特定ベンダの機器等で構築してきた。今後、オープンな環境における公的利用などを普及するには、機器等の互換性・連携やネットワークでの安全性などが重要となる。

そのための標準化やガイドライン策定などについて、国際的に検討が進められている。

■国際標準化の動向

国際的な標準化については、国際標準化機構(ISO)と国際電気標準会議(IEC)の第 1 合同専

門委員会(ISO/IEC/JTC1)内に設置された、第 37 分科委員会(SC37)が中心となって策定が進められている。

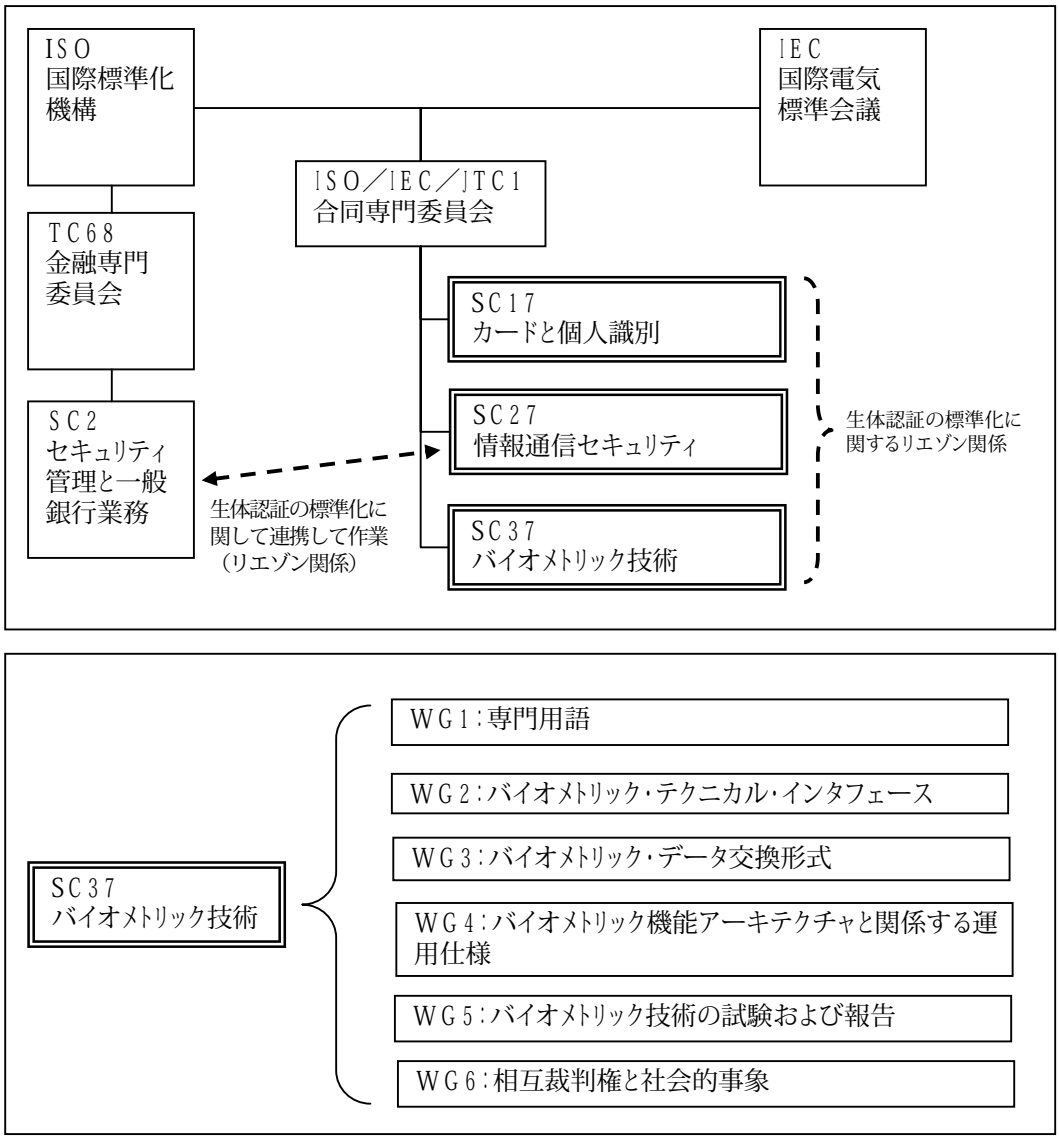


図 2-2 国際標準化の組織と関係［bio09］

■互換性確保のための標準［bio01］［bio06］

互換性向上や開発コスト削減に向けた標準。アプリケーションとのインタフェースやデータ形式を規定。

① BioAPI

プラットフォームに依存しない認証技術部分とアプリケーション間のインタフェース(API ：Application Program Interface)仕様に関する規定。BAPI(Biometric API)や HA-API(Human Authentication API)を統合している。

2000 年 3 月に BioAPI バージョン 1.0、2001 年 3 月にバージョン 1.1 を公開した。バージョン 1.1 は 2002 年 2 月に米国標準規格(ANSI/INCITS 358-2002)となり、現在 国際標準化

に向け活動中。

＜策定団体＞BioAPI Consortium(1998年4月発足)

② Java Card Biometric API

Java Card(ICカード)における認証技術部分とJavaアプレット間インタフェースの規定。(1)オンカードマッチング(MOC)、(2)複数バイオメトリクス技術のサポート、(3)Global Platformの尊重、(4)CBEFFの推進、(5)コンパクト、を要件として制定。

③ ISO/IEC 7816-11

IC カード内に格納するバイオメトリクスデータ形式の標準化とオンカードマッチング(MOC)に関する規格。認証について静的(指紋、顔など)と動的(署名、音声など)に分類して定義。

④ Common Biometric Exchange File Format(CBEFF)

バイオメトリクスデータのファイル形式に関する規格。NISTIR6529(NIST Interagency Reports)として公開するとともに、国際標準化に向け活動中。BioAPIやANSI X9.84などで扱うデータ形式は、CBEFFに準拠している。

＜策定団体＞NIST(米国標準技術局)

⑤ XML Common Biometric Format(XCBF)

XCBF委員会によりOASIS規格として策定している標準XMLスキーマ。バイオメトリクス情報のインターネット転送を行う規格。

＜策定団体＞XCBF委員会(関係団体：OASIS)

⑥ ANSI B10.8

運転免許証のデータフォーマットに関する規格。運転免許書が含む指紋データのフォーマットを規定。

⑦ ANSI/NIST-CSL 1-1993, ANSI/NIST-ITL 1-2000

警察における指紋や顔などのデータの交換を目的としたフォーマット。

■精度評価のための標準［bio01］［bio06］

認証製品の客観的で信頼性の高い評価値の算出を目的とする標準。

① Best Practice

実運用環境での評価について、ベンダー・インテグレーター・ユーザーの全て立場で利用可能な最良の試験実施方法を示すことを目的としたガイドライン。

＜策定団体＞英国 CESG/BWG(Communication-Electronics Security Group, Biometric Working Group)

② JIS-TR

過去のECOMやIPAにおける検討を基に策定された国内規格で、JIS規格化を目的としたガイドライン。

＜策定団体＞情報技術標準化研究センター(INSTAC)

■セキュリティのための標準［bio01］［bio06］

認証製品をセキュリティシステムの一部として捉え、一定のレベル以上のセキュリティを確保するとともに、保証に向けた評価の策定を目的とする標準。

① Biometric Device Protection Profile(BDPP)

情報セキュリティ国債評価基準(CC：Common Criteria)/ISO15408 に基づく生体認証装置の安全性評価を目的とした、セキュリティ要求仕様書(PP：Protection Profile)。

＜策定団体＞英国 CESG/BWG(Communication-Electronics Security Group, Biometric Working Group)

■設計運用のための標準［bio01］［bio06］

アプリケーションより認証装置に求められる要求仕様の標準。

① ANSI X9.84

米国における銀行アプリケーションを対象とした、生体認証システムにおけるバイオメトリクス情報の管理とセキュリティに関する規格。

銀行の顧客及び従業員の識別と認証についてセキュリティ要件や技術を規定。

② 運用要求策定ガイドライン

アプリケーションへの適用に関し、認証精度要件を定量的に作成するとともに認証装置の評価・選定するための指針。

2.2 ID／パスワード認証

(1) 基本的なID／パスワード認証

顔が見えず声も聞こえないネットワーク上では、相手が誰なのか、どんな人なのかを見きわめることは困難である。ネットワーク上でサービス提供者側が相手を特定する際に、ユーザの記憶を頼りに認証する方法としてID／パスワードによる認証方法がある。図 2-3 に示すとおり、個人に配布したIDとパスワードをサーバ上に保存しておき、サービスを受ける際に入力したIDとパスワードがサーバに保管されているそれと一致すれば本人とみなす仕組みである。

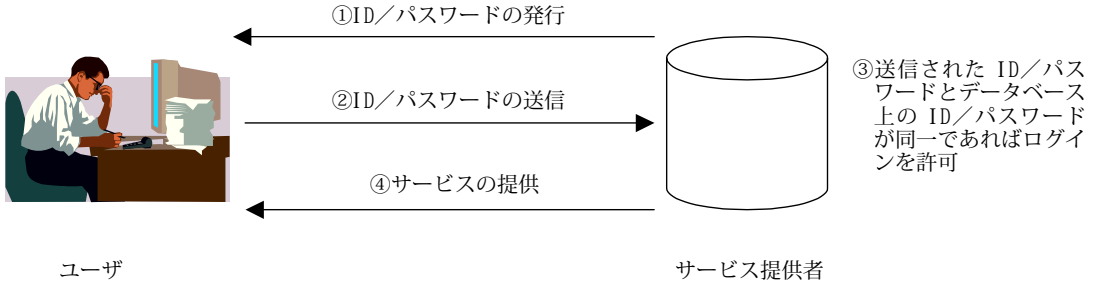


図 2-3 単純なID／パスワードの仕組み

(2) 暗号化通信型ID／パスワード認証

基本的なID／パスワード認証のように単純な仕組みでユーザ側が最初に抱く不安は「通信経路

上からIDとパスワードを盗聴されないか、ということであろう。そこで、図 2-4 に示すとおり、インターネット上では暗号化通信を行うことが多い。

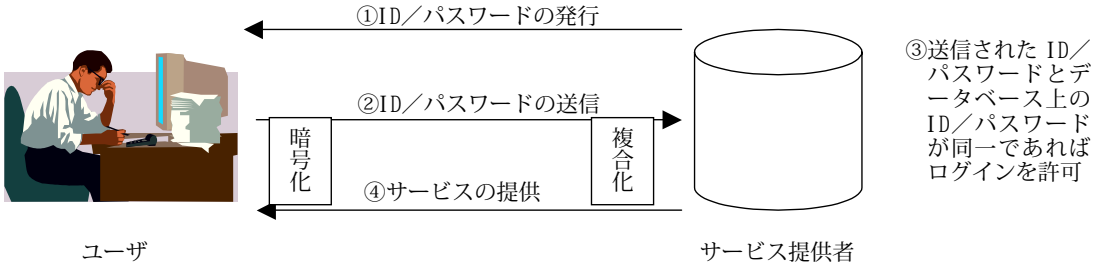


図 2-4 暗号化を施す場合

(3) サーバ証明書型ID／パスワード認証

ユーザ側から見たもう一つの不安として「自分がログインしようとしている相手は本物か」「相手方はどのような会社か」ということがあるだろう。金融機関などを装ったサイトを構築して、ユーザの ID／パスワードを盗むといった手口(Phi shi ng－フィッシングという) による被害も報告されており、ユーザの不安を取り除くことが必要である。

この課題は、図 2-5 に示すとおり、第三者機関によるサーバ証明書の付与などによって解決することができる。これはサイト運営者が自らの「氏素性」を明確にすることでユーザの不安を取り除く効果がある。但し、サイト運営者側からみてログインしようとしているユーザが本人であるかどうかは判断できない。

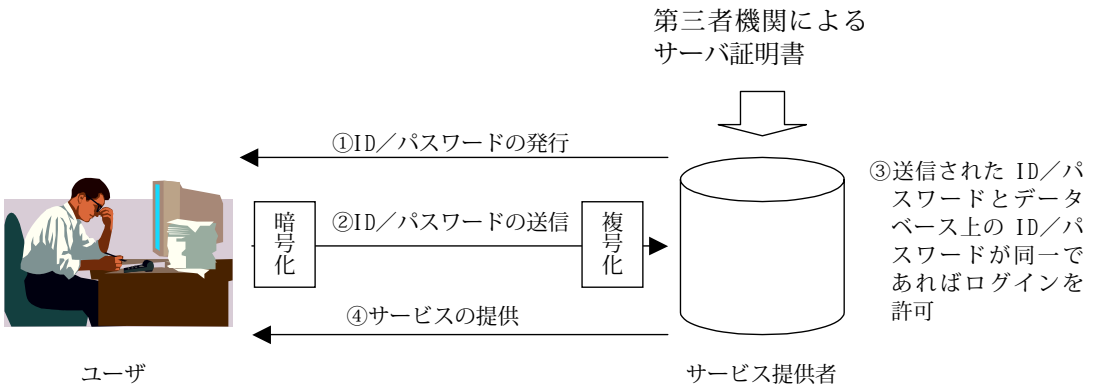


図 2-5 サーバ証明書によりサービス提供者の信頼性を保証

2.3 ワンタイムパスワード

ここまで述べてきた原始的な ID／パスワード認証の場合、第三者に ID／パスワードの情報が漏れた場合は、本人かサービス提供者がその事実を認識して何らかの対抗措置を講じるまで、「永遠に」「何回でも」本人になりすましてアクセスできてしまう。そこで、パスワードの利用回数や有効期限を設定する方法が考案された。これを一般に「ワンタイムパスワード」と呼び、「チャレンジレスポンス型」と「同期型」の二種類がある。

(1) チャレンジレスポンス型

「チャレンジレスポンス型」ではユーザが入力した ID をもとにサーバ側でパスフレーズとシーケンス番号を基に算出した「チャレンジ」という文字列をユーザ側に送信する。ユーザが入力したパスフレーズとチャレンジから計算で得られるレスポンスをサーバに送り返す。サーバでは受け取ったレスポンスとシーケンス番号から計算を行い、この結果がひとつ前のシーケンス番号に対応したパスフレーズに等しい場合のみログインできる仕組みである。この方式では、クライアントとサーバの双方に専用のソフトウェアが必要である。(図 2-6)

［2-3-a］［2-3-b］

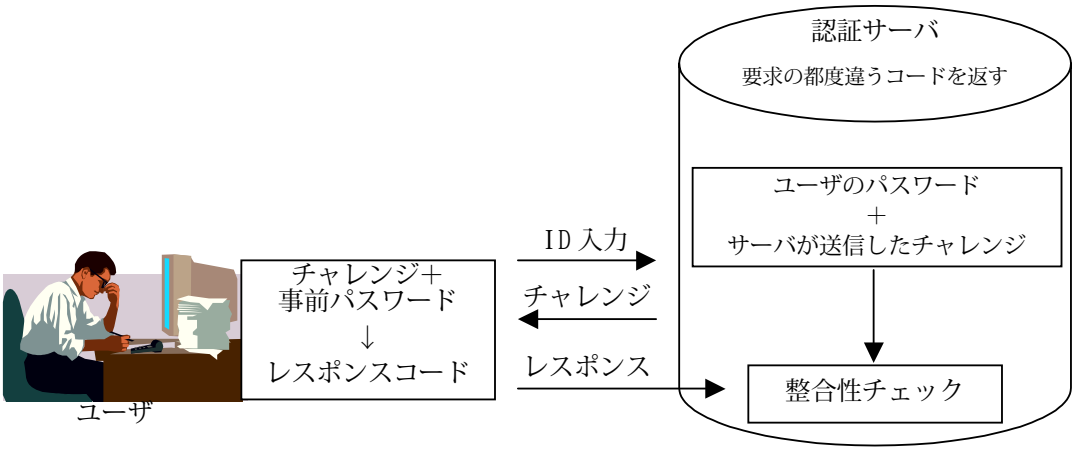


図 2-6 チャレンジレスポンス型ワンタイムパスワードの仕組み［2-3-a］

(2) 同期型

「同期型」には「時間同期型」と「カウンタ同期型」の二種類があり、前者はトークンと呼ばれるワンタムパスワード生成器(認証サーバと時刻を合わせておく)を利用し、時刻をもとに計算したパスコードにより認証を行うものである。後者は、ワンタイムパスコードの生成回数などから同期をとることでログインを許可する方法である。

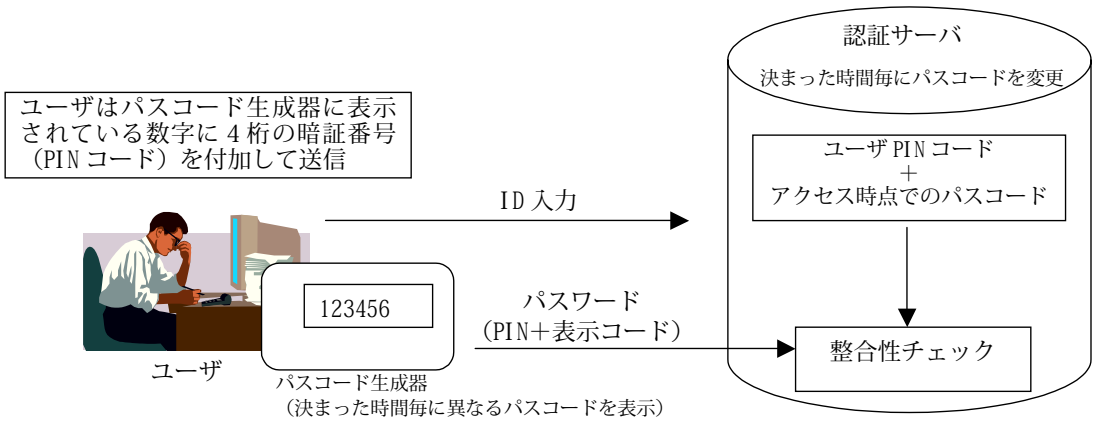
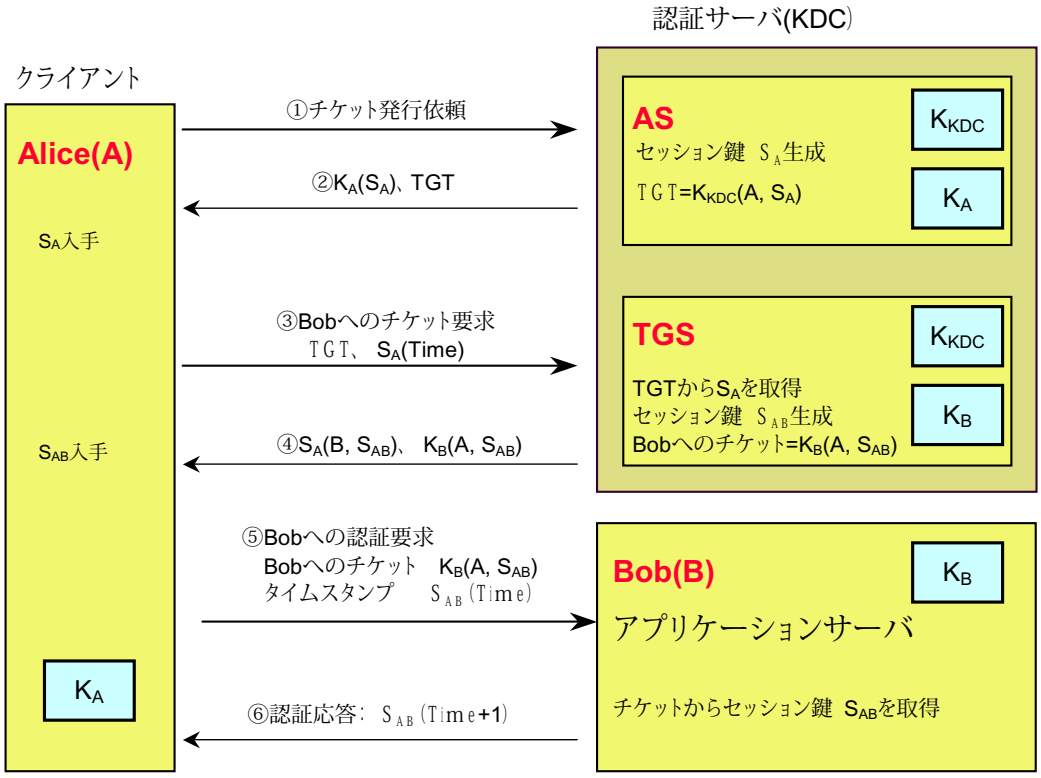


図 2-7 時間同期型ワンタイムパスワードの仕組み［2-3-a］

2.4　ケルベロス認証

ケルベロス(Kerberos)認証は、通信の秘匿やユーザ認証などをすべて秘密鍵(共通鍵)を用いることと、チケットセンターで発行するログインのためのチケットを用いることによって一回 ID／パスワードを入力するだけで複数のシステムにログインできる(シングルサインオン)ことを特徴とする、認証方式である。

現在のバージョンは 5 であり、Windows2000 におけるユーザ認証とアクセス制御に採用された［2-4-a］［bio01］。図 2-8 にケルベロス認証の概要を示す。



AS(Authentication Server)：認証サーバ
KDC(Key Distribution Center)：鍵配布センター
TGS(Ticket-Granting Server)：チケット発行サーバ
TGT(Ticket-Granting Ticket)：TGS とクライアントの通信チケット(チケット交付チケット)

図 2-8 ケルベロス認証の仕組み

① クライアント(Alice)が認証サーバにチケット交付チケットを要求する。
② 認証サーバはセッション鍵(S_A)を生成し、クライアント(Alice)の識別子と S_A をチケット発行サーバとあらかじめ共有している鍵 K_{KDC} で暗号化して TGT を生成する。さらに S_A をクライアント(Alice)のパスワードから生成した鍵 K_A で暗号化し、TGT と共にクライアント(Alice)に送り返す。
③ クライアント(Alice)はユーザにパスワードを入力させて、そのパスワードから生成した鍵 K_A で S_A を復号し、S_A で暗号化した時刻情報などと TGT をチケット発行サーバに送付して、アプリケーションサーバ(Bob)向けのチケット発行を要求する。
④ チケット発行サーバはあらかじめ認証サーバと共有している鍵 K_{KDC} で TGT を復号して、S_A

を入手し、さらに S_A で暗号化されている時刻情報などを復号してユーザを確認する(パスワードを知っている事を確認)。さらにそのユーザがアプリケーションサーバ(Bob)へのアクセス権があることを確認してクライアント(Alicce)とアプリケーションサーバ(Bob)間のセッション鍵(S_{AB})を生成する。さらに、クライアント(Alice)の識別子と S_{AB} を、アプリケーションサーバ(Bob)とあらかじめ共有している鍵 K_B で暗号化してアプリケーションサーバ(Bob)向けのチケットを生成し、S_A で暗号化した S_{AB} と共にクライアント(Alice)に送り返す。

⑤ クライアント(Alice)は S_A で S_{AB} を復号し、S_{AB} で暗号化した時刻情報などとアプリケーションサーバ(Bob)向けのチケットをアプリケーションサーバ(Bob)に送付して、認証とサービスの提供を要求する。

⑥ アプリケーションサーバ(Bob)はあらかじめチケット発行サーバと共有している鍵 K_B でチケットを復号して S_{AB} を入手し、さらに S_{AB} で暗号化されている時刻情報などを復号してユーザを確認し、問題が無ければ認証応答を返してクライアントに対してサービスの提供を開始する。S_{AB} はクライアントとの間で秘密情報を通信するために引き続き利用可能である。

クライアントが別のサーバと通信を行いたい場合は、TGS にその要求を出す。TGT が有効である限り、AS とやりとりを行う必要はない。

2.5 PKI による認証

2.5.1 PKI

PKI による認証では、公開鍵方式に基づくデジタル署名の技術を用いる。このデジタル署名で利用した鍵とその所有者とを結びつけるために認証局を用いている。ここではこれらの技術を順次解説することとする。

(1) 公開鍵暗号方式

PKI(Public Key Infrastructure)は直訳すると、「公開鍵基盤」を意味し、「公開鍵暗号方式」を用いた「基盤(インフラ)」を表している。「公開鍵暗号方式」とは、公開鍵、秘密鍵の対を使用した暗号技術を指す。この技術を用いることで、暗号化、デジタル署名、認証といった様々なセキュリティ対策が実現できる。「基盤(インフラ)」とは組織や社会の土台のことを指す。

公開鍵暗号方式では、「公開鍵」、「秘密鍵」と呼ばれる、対になっている 2 つの暗号用の鍵を用いるが、これら 2 つの鍵の組合せを「鍵ペア」と呼ぶ。公開鍵暗号方式では、これら 2 つの鍵の一方で暗号化した情報はもう一方の鍵でないと復号できないという性質を持つ。この性質を用いて、ネットワーク上の離れた相手に安全に情報を送信することができる。

公開鍵暗号方式を用いた暗号通信の方法例を図 2-9 に示す。

① 利用者 B の鍵ペアのうち、「公開鍵」を公開する。「秘密鍵」は誰にも知られないように保管する。

② 利用者 A は公開されている利用者 B の「公開鍵」を入手する。

③ 利用者 A は入手した「公開鍵」を使用して平文を暗号化し、利用者 B に転送する。

④ 利用者 B は自分の「秘密鍵」を用いて暗号文を復号する。

「公開鍵」で暗号化した暗号文は対応する秘密鍵のみで復号できる。第三者が、公開されている利用者Bの「公開鍵」を使用しても対応する「秘密鍵」を計算などで求めることは事実上不可能なため、暗号文を解読することは出来ない。「秘密鍵」が外部に漏れない限り、暗号文は「秘密鍵」を持つ利用者Bだけが復号出来る。

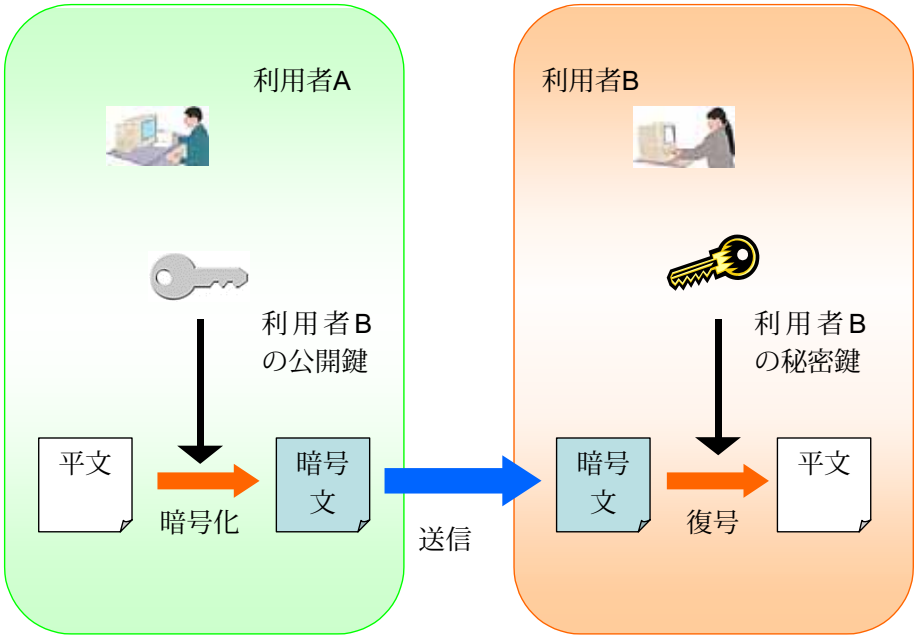


図 2-9 公開鍵暗号方式によるメッセージの送信

公開鍵暗号方式は共通鍵方式より複雑な演算処理を行うため、より多くの計算量を必要とする。そのため、暗号通信を行うときは、一般的には高速な処理を実現するため、公開鍵暗号方式と共通鍵暗号方式を組み合わせて使用する。電文を送信する際には公開鍵暗号方式を用いて共通鍵を安全に送信し、次に送信した共通鍵を使用してメッセージを暗号化する。図 2-10 に公開鍵暗号方式を用いた共通鍵の配送例を示す。

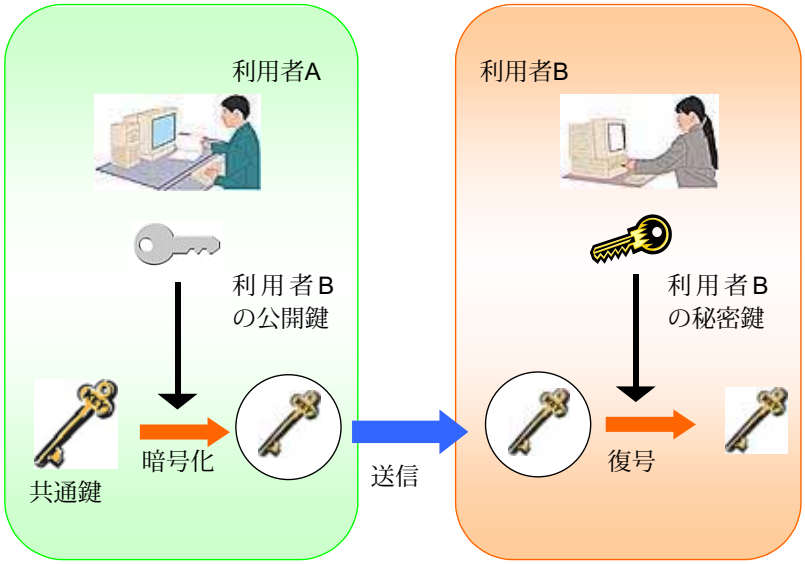


図 2-10 公開鍵暗号方式を用いた共通鍵の配送

図 2-11 に上記の公開鍵暗号方式により配送された共通鍵を用いた共通鍵による通信路の暗号化の例を示す。

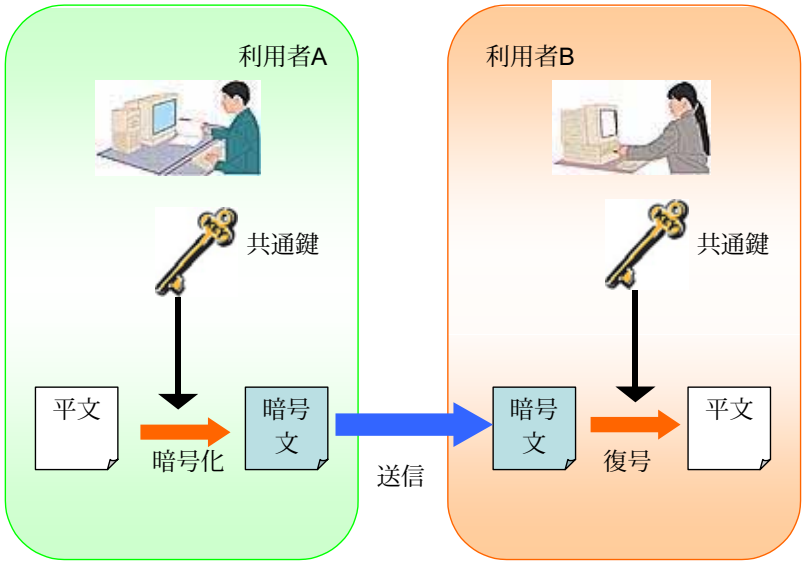


図 2-11 共通鍵による通信路の暗号化

(2) デジタル署名

上記では、「公開鍵」で暗号化し、「秘密鍵」で復号する方法を示した。これとは逆に、「秘密鍵」で暗号化して「公開鍵」で復号することもできる。「公開鍵」は誰にでも入手できるため、電文の復号化はどの利用者でも行える。このため電文の秘匿は行えないが、秘密鍵が特定の者のみ所有していることから、電子文書の発信者を特定することが可能となる。

公開鍵暗号方式を用いたデジタル署名の実現のために必要な技術として、ハッシュ関数が

ある。この関数は、可変長の入力データから固定長のデータを生成する。生成されたデータをダイジェストと呼ぶ。

ハッシュ関数には次の性質がある。

- 入力データの長さが異なっていても、生成されるダイジェストの長さは一定
- 入力データが少しでも異なれば、生成されるダイジェストは大きく異なる
- ダイジェストから元のメッセージを再生することは出来ない
- 同じダイジェストを出力する 2 つの入力データを見つけるのは困難

電文のダイジェストを作成し、これを秘密鍵で暗号化すると、秘密鍵の所有者しか作成できないデータが生成できる。これにより、電文の改ざんを検出することができる。これをデジタル署名と呼ぶ。デジタル署名を用いると、メッセージの完全性の確保と作成者の認証とが同時に可能となる。

図 2-12 にデジタル署名の生成と検証の例を示す。ここで、利用者 B は利用者 A の公開鍵を予め安全な方法で入手しているものとする。この入手の方式については(3) 認証局の節で詳しく述べる。

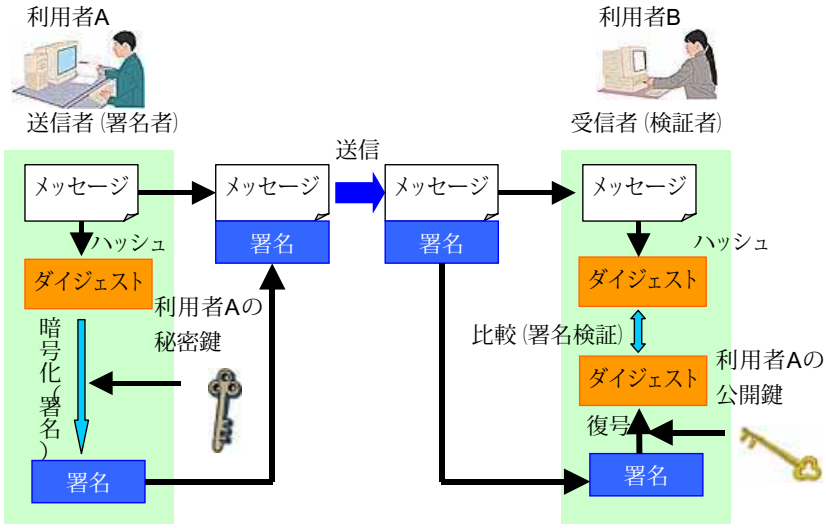


図 2-12 デジタル署名の生成と検証

デジタル署名の生成(利用者 A)

(a) 署名したい電文から、ハッシュ関数を用いてダイジェストを生成

(b) 生成したダイジェストを利用者 A の秘密鍵で暗号化

(c) 電文と生成した署名とを利用者 B に送信

デジタル署名の検証(利用者 B)

(d) 受診した電文から、ハッシュ関数を使ってダイジェストを生成

(e) 受診したデジタル署名を利用者 A の公開鍵を用いて復号

(f) (d) で生成したダイジェストと(e) で復号したダイジェストを比較し、完全に一致することを確認

デジタル署名は、電文のダイジェストを送信者の秘密鍵で暗号化したものである。受信者は、このデジタル署名を送信者の公開鍵を用いて復号し、受信した電文のダイジェストと比較することにより、電文が改ざんされていないことがわかる。また、送信者の公開鍵と対応する秘密鍵で作成されたデジタル署名であることが確認できる。

(3) 認証局

公開鍵暗号方式では、はじめに通信相手に公開鍵を安全な方法で渡しておく必要がある。このやり方として、媒体に公開鍵を格納し、相手と面と向かって確認しあい、手渡す方法等が確実である。しかしながら、ネットワークを経由しての通信では通信相手が見えないため、第三者が通信相手になりすまして不正な公開鍵を送信してくる可能性がある。そのため、公開鍵方式を用いる場合には、使用する公開鍵が本当に正しい相手のものであるかを確認する必要がある。

このとき、通信相手の双方が信頼できる第三者機関(TTP) に公開鍵の所有者を保証する電子証明書を発行してもらう方法がある。この電子証明書を保証してくれる第三者機関を認証局(CA) と呼ぶ。認証局は、公開鍵の所有者の本人確認(実在性、本人性) を行い、公開鍵とその所有者とを保証する電子証明書を発行する。電子証明書には、公開鍵とその所有者を証明する情報が記載され、改ざんを防ぐために認証局の署名が付与される。図 2-13 に PKI における公開鍵の交換に CA を用いて行った例を示す。

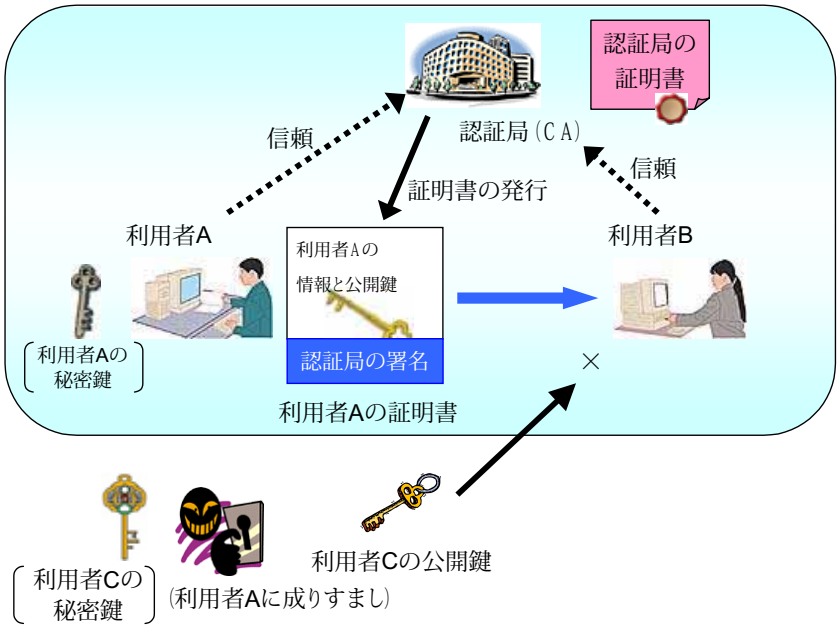


図 2-13 PKI における公開鍵の交換

(a) 利用者 A は CA に公開鍵を提出する
(b) CA は窓口での対面などで、利用者 A が実在する人であること(実在性)、窓口に来ている人が利用者 A であること(本人性)を確認(本人確認)し、利用者 A の電子証明書を発行する
(c) 利用者 A は発行された電子証明書を利用者 B に送信する
(d) 利用者 B は、電子証明書に記載されている利用者 A の情報と CA のデジタル署名とを確認し、電子証明書内の利用者 A の公開鍵を入手する

利用者 C が利用者 A になりすまして公開鍵を送信しても、CA のデジタル署名を偽造できないため、CA のデジタル署名が記載された電子証明書を作ることが出来ず、利用者 B はなりすましされていることを容易に検出できる。

CA は自分の公開鍵を証明するための電子証明書を持つ。CA の電子証明書にはその CA 自身がデジタル署名を付与している。このような電子証明書を自己署名証明書と呼ぶ。

CA は、その電子証明書の発行方法について、利用者から信頼されるものである必要がある。証明書の発行時に遵守している事柄などを証明書発行ポリシーとして「認証局運用規程(CPS)」を定めて公開する。また、国が CA を認定する認定認証業務の制度も法律で規定されている。

(4) 電子認証

PKI での電子認証では、上記で説明した電子証明書とデジタル署名技術を利用する。相手を認証しようとする側(A)と認証される側(B)の間の電子認証について説明する。A は B の電子証明書を入手しておく。

認証の流れは次のとおりである。

(a) A が B にランダムな値(これをチャレンジ値という)を送信する。
(b) チャレンジ値に B がデジタル署名を作成(B の秘密鍵でチャレンジ値を暗号化)し、A に送信する。
(c) A は、B から送られたデジタル署名を B の電子証明書から取得した B の公開鍵を用いて復号する。
(d) A は(c)で復号した値と、(a)で送信した値とを比較する。

上記で比較した結果が一致すれば B しか所持していないはずの「B の秘密鍵」を持っていることが確認できることから、認証される側は正しく B であることが確認できる。

電子署名法［HOUMU-SHOMEI］に規定する自然人の実印と同様の電子署名と、上記の電子認証とは同様なデジタル署名技術を利用している。電子署名法に規定する電子署名はその署名された文書が署名した人が作成したものであること、及びその文書が改ざんされていないことを示すために利用される。このため、その電子署名は実印を押印した紙の文書のように比較的長期にわたり保存され、後日その真正性の確認に用いられることがある。これに対し、

上記の電子認証は、デジタル署名を行うデータはランダムな値(タイムスタンプなどを加えることもある)であり、署名したデータ自体は意味を持たない。また、通信の相手方が正しく認証されればその署名データを保存する必要は無い。

2.5.2　国際標準・業界標準

(1) X.509 / RFC 3280

X.509 はITU(国際電気通信連合)が1988年に第1版を勧告した、電子証明書に関する標準仕様である。認証局の発行する電子証明書及び、それの失効に関する情報を記載する証明書失効リストの標準仕様が規定されている。

現在広く用いられているのは 1997 年に勧告され、その改訂版が 2000 年に勧告された X.509 第3版(X.509v3)である。以前の版と比べ、電子証明書に拡張領域が設けられ、発行者が独自の情報を追加できる。

X.509v3 の電子証明書は公開鍵のバージョン番号、電子証明書のシリアル番号、公開鍵情報、電子証明書を発行した認証局情報、電子証明書の有効期間、証明される主体者の情報、拡張領域等で構成されている。拡張領域には電子メールアドレスや IP アドレスなどのアプリケーションに依存する情報を記載できる。

この X.509 は主に汎用的な枠組みが規定されている。この利用方法については 1995 年に設立されたIETFの PKIX(Public-Key Infrastructure(X.509))ワーキンググループにて規定している。1997年版X.509に対応して1999年にはRFC 2459が策定され、X.509電子証明書と電子証明書失効リストのプロファイルが策定された。2000年版X.509に対応して2002年にはRFC 3280が規定され、相互認証や、証明書の状態等についての規定が追加された。

RFC 3280にて定義されたX.509証明書の基本領域のプロファイルを表 2-5 に示す。

表 2-5 X.509 証明書の RFC3280 基本プロファイル

領域名	説明
tbsCertificate (署名前証明書)	電子証明書の基本的な情報と公開鍵を示す。
version (バージョン)	X.509 証明書のバージョンを示す。
serialNumber (シリアル番号)	電子証明書を一意に識別するための番号。発行した CA が割り当てる。
signature (アルゴリズム識別子)	発行する CA が電子証明書に署名する際に用いる書名アルゴリズム。OID で指定する。
issuer (発行者)	電子証明書を発行した CA の名前。
validity (有効期限)	電子証明書の有効期限を表す。
notBefore (開始時刻)	電子証明書が有効となる時刻。
notAfter (終了時刻)	電子証明書が無効となる時刻。
subject (主体者)	証明書の所有者の名前。ユーザの名前やサーバ名などが記述される。
subjectPublicKeyInfo (主体者公開鍵情報)	証明書所有者(主体者) の公開鍵に関する情報。
algorithm (アルゴリズム)	公開鍵のアルゴリズム名。OID で指定する。
subjectPublicKey (主体者公開鍵)	主体者が所有している公開鍵。
issuerUniqueID (発行者ユニーク識別子)	発行者名を再利用した際に、発行者を識別するために使用される。(発行者名は再利用しないこと、本識別子を使用しないことが推奨されている)
subjectUniquiID (主体者ユニーク識別子)	主体者名を再利用した際に、主体者を識別するために使用される。(主体者名は再利用しないこと、本識別子を使用しないことが推奨されている)
extensions (拡張領域)	電子証明書の拡張領域。
signatureAlgorithm (署名アルゴリズム)	発行者が電子証明書に署名する際のアルゴリズム。OID で指定する。
signaureValue (署名値)	発行者のデジタル署名が格納される。

(2) クオリファイド証明書

欧州における個人認証の必要性から、自然人(個人)を対象にして、法的に認められるための証明書として必要となる各種条件が検討された。その結果、セキュリティポリシーとそれを反映した証明書フォーマット(プロファイル)の制定が必要と判断された。これを満たすため欧州の標準化団体により提唱された標準が IETF で採用され、RFC 3039「クオリファイド証明書」として規定された。その後、このクオリファイド証明書のプロファイルについて改訂され、RFC 3739 が規定された。

クオリファイド証明書には、以下のような特徴がある。

- X.509v3 に準拠している
- 基本領域、拡張領域への記載内容にルールが設けられている。
- クオリファイド証明書に特化した拡張領域を持っている
- 「人」を対象とした証明書であり、そのために必要となるポリシーを規定している

記載内容に関するルールには、欧州電子署名指令案(EU-directive、1999)の指示のもと ETSI による標準化検討がなされた。検討された標準に加えて、実際には欧州における法律制度・社会制度にのっとり、内容についてさらに詳細な規定を加えて利用されようとしている。

クオリファイド証明書を定義した RFC 3739 は、利用される国や団体の幅広い要件に対応できるように汎用的な内容になっている。

RFC 3739 では、X.509 電子証明書プロファイル(RFC 3280, RFC 2459)をベースに、クオリファイド証明書に必要となるプロファイルを定めている。クオリファイド証明書として独自に拡張したフィールドには「生体情報」があり、既存の基本領域・拡張領域はクオリファイド証明書として必要となる注意事項・制約事項等を定義している。表 2-6 に RFC 3739 のプロファイルを示す。

表 2-6 RFC 3739　証明書プロファイル

項目		内容
Basic Certificate Fields (基本領域)		
	Issuer (発行者名)	発行者組織・名称は公的に登録された名称。
	Subject (主体者名)	証明書の発行を受ける人(主体者)のディレクトリネーム、通称、本名等。
Cirtificate Extensions (拡張領域)		
	Subject Alternative Name (主体者別名)	ディレクトリネームなどで表現される主体者の別名。この項目は必須ではない。
	Subject Directory Attributes (主体者ディレクトリ属性)	主体者の生年月日、出生地、性別、国籍、居住地が記載可能。この項目は必須ではない。
	Key Usage (鍵使用目的)	この項目は必ず設定することになっている。設定内容はRFC 3280 に従う。
	Biometric Information (生体情報)	この項目には、オプションとして、生体情報のハッシュ値を格納できる。値の実体を格納したアドレスを記載することも出来る。
	Qualified Certificate Statements (QC 宣言)	法的な説明文(QC 宣言)を登録した OID を記載する。この項目はオプションである。

クオリファイド証明書を利用したバイオメトリック認証の利点としては、既存のバイオメトリック認証を利用したシステムと異なり、利用者の生体情報を保管・保持しなくて良く、また、生体情報をネットワーク経由でサーバに送らなくても良い等プライバシの問題が解決できることがある。

(3) PKI アプリケーション

TLS(Transport Layer Security)は、クライアント／サーバ間における安全な通信環境を提供する通信プロトコルである。TLS は電子証明書を利用することにより通信の守秘性、認証、完全性を確保している。TLS は通信プロトコルのモデルのうち、トランスポート層での暗号化を行なっている。このため、さまざまなアプリケーションプロトコル(HTTP, LDAP, FTP, TELNET)などで利用可能である。

TLS は NetScape 社によって提唱された SSL(Secure Socket Layer)から派生している。SSL は、主に Web において、クレジットカード番号や個人情報のような重要な情報を保護するのに利用されている。

IETF において、SSL のバージョン 3.0 を元に TLS が策定され、RFC2246 として公開されている。

TLS / SSL を利用するにはサーバに電子証明書が必要となる。クライアント認証をする場合には、クライアント側にも電子証明書が必要となる。

TLS / SSL は以下のセキュリティ機能を提供する。

(a) 認証(Authentication)
電子証明書を利用することで、サーバ及びクライアントの認証を行い、第三者によるなりすましを防ぐ。サーバ側の認証のみを行い、クライアント側の認証を行わないことも可能である。

(b) 守秘性(Confidentiality)
サーバとクライアント間の通信を暗号化することで、第三者への情報の漏洩を防ぐ。暗号化には共通鍵暗号方式(RC4、トリプル DES など)が利用される。共通鍵をサーバとクライアント間で交換する時に、X.509 証明書に含まれる公開鍵が使われる。

(c) 完全性(Integrity)
サーバとクライアント間で交換されるデータの完全性の確認には MAC(Message Authentication Code)を用いている。これにより、サーバ、クライアント間の電文の完全性を確認し、情報の改ざんを防ぐことができる。

(4) JIS X 5056-3

JIS X 5056-3:2002「セキュリティ技術―エンティティ認証―第 3 部：デジタル署名技術を用いる機構」は、1998 年に第 2 版として発行された ISO / IEC 9798-3「Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques」を技術的内容及び構成を変更することなく日本工業規格として採用するために規定されている。

本規格では、デジタル署名、タイムスタンプなどを利用して相手方を認証する方式を規定している。認証方式としては、片方向認証、および相互認証の両方が規定されている。片方向認証は 1 パス及び 2 パス認証方式、相互認証では 2 パス、3 パス、及び並列 2 パスの認証方式がそれぞれ規定されている。

2.6 シングルサインオン

前述のようなID／パスワード認証はユーザの記憶による認証であるから、本人がそれらを失念した場合は本人拒絶されてサービスを受けることができない。そこで、ID／パスワードを手帳やPC のどこかにメモしたり、各種サービスのパスワードを全て同一のものにするといった、セキュリティ上好ましくない対応をとるユーザも少なくない。また、ネットワーク上の様々なサービスを複合的に利用する場合に、複数のID／パスワードの入力を求められるのでは不便でもある。

そこで、複数の OS やアプリケーション、サービスへのログインを一度の認証で許可する仕組みが考案されている。そのような仕組みを「シングルサインオン(Single Sign-on)」と呼び、様々な製品がリリースされている。これらは多種多様なプラットフォームを組み合わせて構築したシステム上でも認証情報を引き継ぐように開発されている。

シングルサインオンを実現するための仕組みとしては、「エージェント型」と「リバースプロキシ型」と呼ばれる二つのの方法がある。「リバースプロキシ型」はユーザと各ウェブサーバの間に認証サーバを構築し、全ての通信は認証サーバを経由する方式である。対して「エージェント型」では認証時にウェブサーバにインストールしたプラグインソフトが認証サーバと通信する。両者には開発時の既存システムとの親和性や認証サーバへの負荷による通信のボトルネックなど様々な面でそれぞれ一長一短があり、それぞれの特徴を活かした中間的な製品も開発されている。

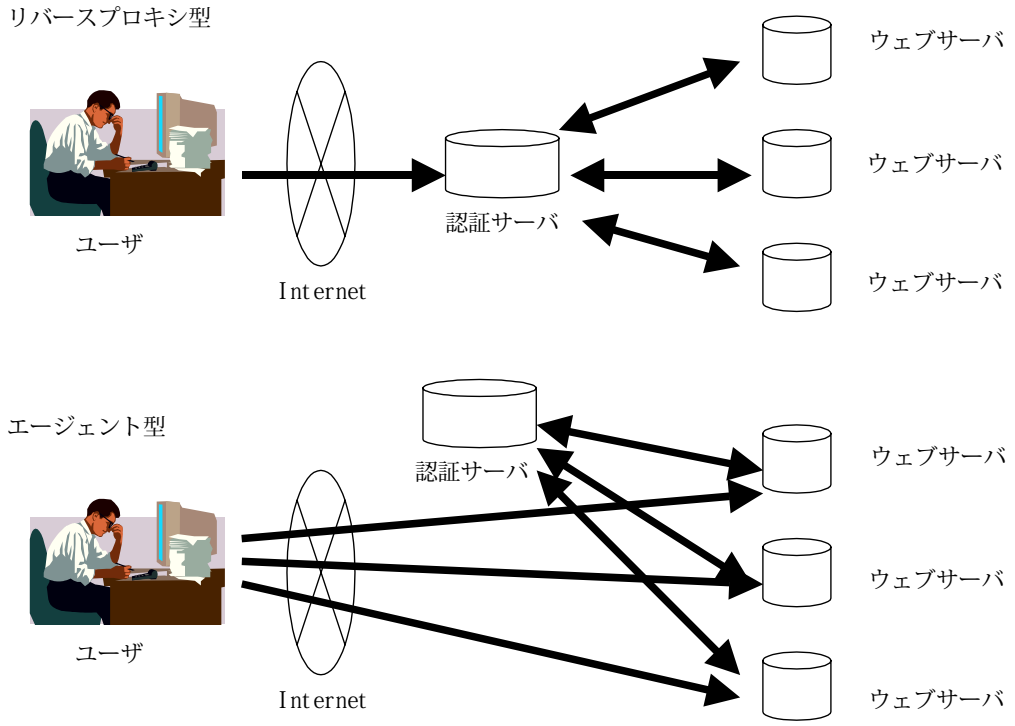


図 2-14 シングルサインオンを実現するための仕組み

2.7　参考文献

[bio01]「本人認証技術の現状に関する調査報告書　2003 年 3 月」(情報処理振興時業会セキュリティセンター)

[bio02] サイバーセキュリティにおける生態認証技術(瀬戸洋一 著、共立出版株式会社)

[bio03] トコトンやさしいバイオメトリクスの本(明石正則監修、神鋼リサーチ 編著、日刊工業新聞社)

[bio04] バイオメトリクスカタログ(http://www.atmarkit.co.jp/、株式会社アットマーク・アイティ)

[bio05] バイオメトリクス認証(http://www.keyman.or.jp/、株式会社リクルート)

[bio06] Standardization(http://www.sdl.hitachi.co.jp/、日立製作所 システム開発研究所)

[bio07] バイオメトリクス認証技術(http://www.secom.co.jp/isl/j/theme/、セコム株式会社 IS 研究所)

[bio08] 利用者の意思を確実に伝える情報セキュリティ基盤技術の研究(研究代表者 松本勉、国立情報学研究所 http://www.nii.ac.jp/research-j.html)

[bio09] バイオメトリクス認証と PKI(セコム株式会社 IS 研究所 松本泰)
http://www.jnsa.org/seminar/API/Matsumoto2.pdf

[2-3-a]「リモートアクセス環境におけるセキュリティ」2002 年 3 月(情報処理振興事業協会セュリティセンター)

[2-3-b]「キーマンズネット」(株式会社リクルート) http://www.keyman.or.jp/

[2-4-a]「IT 用語辞典 e-Words」(株式会社インセプト) http://e-words.jp/

［HOUMU-SHOMEI］電子署名及び認証業務に関する法律(平成 12 年 5 月 31 日法律第 102 号)

[IPA-PKI] PKI 関連技術解説、http://www.ipa.go.jp/security/pki/、2004 年 5 月 21 日：情報処理推進機構

[IEC0798] ISO / IEC 9798-3：1998 Information technology –Security techniques–Entity authentication–Part 3：Mechanisms using digital signature techniques

[JIS5056] JIS X 5056-3：2002 セキュリティ技術　－エンティティ認証－第 3 部：デジタル署名技術を用いる機構

3. 属性認証

アイデンティティ認証によって、システムにアクセスを要求したユーザが何者であるか、その本人性についてシステム側で認証することが可能であるが、それに加えてユーザの持つ権限について把握した上でアクセスの可否を決定したい場合がある。システムによっては、ユーザの住所や年齢によって表示するデータを変えたい場合もあるだろう。このような、ユーザの権限や住所、年齢などのデータをユーザの「属性」と言い、システムの振舞いを変えるためにユーザの属性を確認することを「属性認証」と言う。規模の小さなシステムであれば、ユーザのアイデンティティが確認できればユーザの属性がわかってしまう場合もあるが、複数のサーバを組み合わせたような比較的規模が大きなシステムの場合には、個々のユーザの属性を全てサーバ側で管理把握しておくことは困難で、認証処理の時にユーザ側から自身の属性について申告をさせたい場合もある。

この章では、属性の概念について説明した後、属性認証に使われる技術を 6 つ紹介する(公開鍵証明書、属性証明書、データベース管理、SAML、Liberty、WS-Federation)。さらに、属性が認証された上で、ユーザの役割に基づいて権限を管理する仕組みと、ポリシーに基づいて権限を管理する仕組みについて解説する。

3.1 属性とは

属性(attribute)とは、商取引等に関わる主体の職責／資格／地位などである[1]。これは、商取引や申請手続き等が電子化されているか否かに依存しない概念である。営業部長が「代表取締役から委任を受けて、ある期間、ある業務に関する営業行為を行える」ことは、その営業部長が業務を行う上で重要な属性の一つである。属性を持つ主体はこの例のように典型的には個人であるが、組織や組織内の職位などの場合もある。属性には次のような例がある。

- 商取引における代表取締役の社内代理人に関して代表取締役からの委任を受けている事実(委任状で表現されている)
- 申請等の代理人資格(社内代理人、行政書士、弁理士等)
- 社員の所属、担当業務、職位
- 処方箋発行における医師資格
- インターネットショップにおける会員資格

属性を利用するにあたっては、以下の観点で利用する属性を分類し、適切に取り扱う必要がある。

[1] より正確には、職責／資格／地位などの中でも特にある具体的な応用や業務の観点から関心の対象となるものを属性という。ある営業部長が将棋四段の免状をもっている事実は、電子商取引の観点からは不要な属性である。

(1) 時間の経過に伴い変化する属性情報かどうか

属性情報には、生年月日など時間の経過に伴い変化しない情報と、権限・職責など変化する情報がある。

- 先天的に変わらない情報
  生年月日、バイオメトリクス情報など
- 後天的に付加され変わらない情報
  学歴・職歴などの経歴、賞罰など
- 時間的にあまり変化しない情報
  資格、免許、職業、住所など
- 時間的に変化しやすい情報
  所属、職責、権限、ランク、資産(預金残高等) など

(2) オープンな属性情報かどうか(信頼できる属性情報かどうか)

属性情報には広くオープンな世界で通用する情報と、限られたコミュニティ内で通用する情報がある。

- オープンに通用する情報
  住民基本情報、商業登記情報、資格、免許(許認可) など
- コミュニティ内で通用する情報
  所属部門、職責、ランク、会員資格など

属性情報は使用される世界と密接な関係があり、コミュニティごとに信頼される情報は異なる。

(3) センシティブな属性情報かどうか

属性情報には一般に知られてもよい(あるいは知らせたい) 情報と、知られたくない情報がある。

- 知られてもよい(知らせたい) 情報
  氏名、会社名、資格、免許(許認可) 、賞など
- 知られたくない情報
  戸籍情報、住所・TEL、経歴、罰、診療情報、成績、年収など

属性情報には個人(私的) 情報と組織人(公的) 情報がある。個人(私的) 情報は保護対象として特に厳重な管理が必要である。

(4) 必要な属性情報かどうか

コミュニティやサービス毎に必要とする属性情報は異なるが、おおむね以下の 3 つに分類される。

- コミュニティやサービスに共通な属性情報(一次属性情報とよぶ)
- コミュニティやサービスで必要とする属性情報(二次属性情報とよぶ)
- コミュニティやサービスで必要としない属性情報(不要属性情報とよぶ)

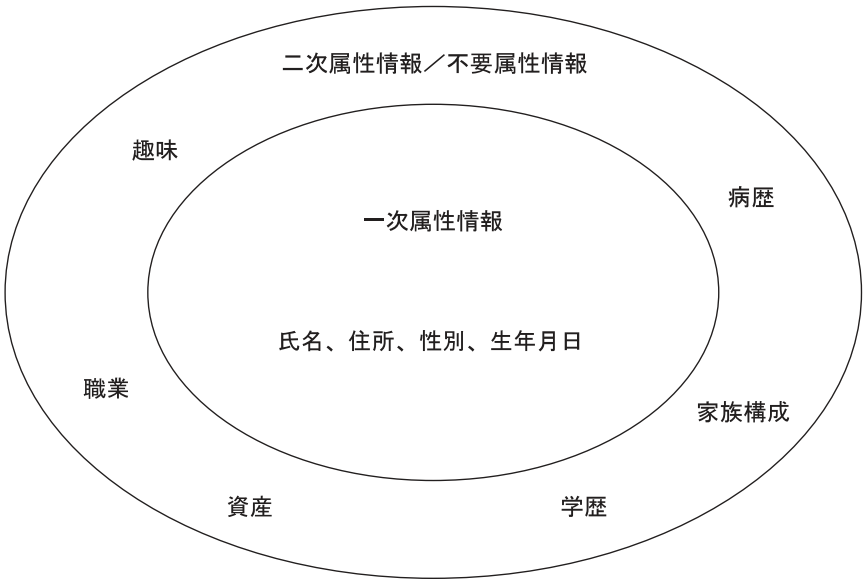


図 3-1 属性の分類

属性情報の分類を図 3-1 に表すと、氏名、住所等はどのコミュニティやサービスでも必要(一次属性情報) だが、病歴は診療サービスでは必要(二次属性情報) 、金融サービスでは不要(不要属性情報) であろう。つまり、二次属性情報と不要属性情報については、コミュニティやサービス毎にどの属性を二次属性情報とみなし、どの属性を不要属性情報とみなすかが変化する。

一般に電子商取引や申請手続きを電子化した情報システムで属性を利用する目的は、主体の属性によって主体がそのシステムで利用できるサービスや機能に差をつけることである。そのためには主体の属性を認証する前に、システムが主体を識別し、主体のアイデンティティを認証しておく必要がある。システムが主体を識別するためには通常一次属性情報が利用され、主体のアイデンティティの認証には二次属性情報が利用される。システムが主体を認証するための段階とその目的、利用される属性をまとめると、表 3-1 のようになる。

表 3-1 認証の段階と利用される属性

認証の段階	目的	利用される属性
識別	ある母集団の中から特定の主体を他の主体と区別すること。	名前、性別、住所、生年月日などの一次属性情報。
アイデンティティ認証	システムにアクセスした主体が提供した情報によって、その主体と主体の識別情報との対応付けを行うこと。	パスワードや社員番号などの二次属性情報。識別時にシステムから主体に秘密裏に提供された情報が利用される場合もある。
属性認証	アイデンティティ認証済みの主体のその時点での属性によって、システムが主体にシステムを操作したり、システム内の情報にアクセスしたりする権限を与えること。	二次属性情報。

本章では属性認証の手段について概説する。

3.2　公開鍵証明書

システムが主体(＝システムの利用者)の属性を認証するための情報を得る手段として、公開鍵証明書(PKC:Public Key Certificate)を利用する方法である。公開鍵証明書のフォーマットはITU-T Recommendation X.509[X509]で規定されている。システムがPKCを使用して属性を得る方法には以下の2つの方法がある。

表 3-2 公開鍵証明書に属性を設定する方法

PKCによって属性を提示する方法	説明
PKC の発行対象者を限定することで利用者がある属性及び属性値を有することをPKCに暗示する方法	PKCを属性とみなす方法である。属性を保証する団体が属性を持つ個人に対してのみPKCを発行する。PKCを用いた利用者認証が成功したら、利用者がその属性及び属性値を有していることを検証できたことになる。 例. 弁護士会等、士業の団体が各加盟者に対してPKCを発行。 PKCの申請時に、各個人が属性情報を添えて申請するか、所属する組織がまとめて発行することが想定される。
利用者がある属性及び属性値を有することをPKCに明示する方法	利用者が保有する属性及び属性値をPKCの中に記載してPKCを発行する。属性及び属性値を記載するPKCのフィールドとしてはsubject やextensions を使用している例がある。PKC の属性及び属性値を確認することによって利用者がある属性及び属性値を有していることを検証する。PKC発行時に属性情報を確定する必要がある。

3.2.1　特徴

PKCを利用した属性認証には以下の特徴がある。

① 継続性の低い属性(一時的な属性)を取り扱いにくい。

② 属性及び属性値の改ざん防止がデジタル署名によって保証される。

③ 利用者がある属性及び属性値を有することをPKCに暗示する方法では、単にPKCの検証ができたことをもって属性及び属性値を有すると判断するため、市販のPKI 関連ソフトウェアを利用できる可能性が高く、システム構築が比較的容易である。

④ 利用者がある属性及び属性値を有することをPKCに明示する方法では、実際に利用する際には PKC に独自の属性も含めて複数の属性が格納されている可能性がある。しかし、そのようなPKCの、属性を検証する汎用的検証ソフトウェアを予め用意するのは容易でない。市販の検証ソフトウェアを利用するにしても、アプリケーションに応じてカスタマイズ開発が必要になる可能性が高い。

⑤ 利用者がある属性及び属性値を有することをPKCに明示する方法では、相互運用性の高いシステムやグローバルなシステムを構築するには、属性及び属性値(ならびにそれらに対応するオブジェクト ID)の業界標準化／国際標準化が必要であるが、現状では不十分である。

⑥ PKC を用いる技法は、利用者同士のデータのやり取りで一方の利用者が他方の利用者の属性を検証するモデル(End-to-End モデル)のアプリケーション構築に向いている

⑦ 属性／属性値が変更になった場合はPKCを発行しなおす必要がある。
⑧ PKCを用いる技法では、複数のPKCの中から使用するPKCを選択する操作が必要になる可能性があり、他の技法と比較してその分操作性が低下する恐れがある。
⑨ 速度性能については、PKC を用いる技法では、PKC の検証のみを行えばよいため、属性証明書(AC: Attribute Certificate)を用いる技法と比較すれば早いと考えられる。

以下の節では、PKCに属性及び属性値を明示する方法の例を説明する。

3.2.2 ISO/TS 17090 hcRole

ISO/TS 17090-2[ISO17090]で定められている hcRole 属性は医療従事者の役割を表すための属性で、PKCに設定する場合はSubject directory attributesエクステンションに設定して使用する。現在のところ、国内においてはhcRole属性には医療従事者の役割を示す以下の文字列を設定する方向で検討が進められている。

表 3-3 hcRole属性の値と医療従事者の役割

hcRoleに設定される文字列	医療従事者の役割
Medical Doctor	医師
Dentist	歯科医師
Pharmacist	薬剤師
Medical Technologist	臨床検査技師
Radiological Technologist	診療放射線技師
General Nurse	看護師
Public Health Nurse	保健師
Midwife	助産師
Physical Therapist	理学療法士
Occupational Therapist	作業療法士
Orthoptist	視能訓練士
Speech Therapist	言語聴覚士
Dental Technician	歯科技工士
National Registered Dietitian	管理栄養士
Certified Social Worker	社会福祉士
Certified Care Worker	介護福祉士
Emergency Medical Technician	救急救命士
Psychiatric Social Worker	精神保健福祉士

hcRole属性は厚生労働省の医療情報ネットワーク基盤検討会でも、国家資格を表現する方法として使用することが推奨されている［HPKIREPORT］。

3.2.3 日本認証サービスの属性型証明書

日本国内で事業として属性付きのPKCを発行するサービスとして、日本認証サービス株式会社の属性型証明書がある［JCSICP1］［JCSICP2］。日本認証サービスの属性型証明書発行サービスはAccreditedSign パブリックサービス 1 と Accredited パブリックサービス 2 の 2 つがあり、AcrreditedSignパブリックサービス1は一般用途用として一般の電子文書へのデジタル署名や電子メールの暗号化などに利用できるPKCを発行するサービスであり、AccreditedSignパブリック

サービス 2 は行政に対する申請・届出等の手続きを電子的方法で行う場合に利用できる PKC を発行するサービスである。どちらも電子署名法の特定認証業務の認定を受けている。この二つのサービスで発行される属性型の加入者証明書には加入者が所属する組織に係る属性をあらわす識別情報が記載される。属性の記載方法はどちらも同じであり、Subject alternative name エクステンションに Name の形式(DN と同じ形式) で格納される。属性値の格納に使用される属性は以下のとおりである。

表 3-4 日本認証サービスの属性型証明書で使用される属性と属性値

属性	設定する属性値	エンコード
C	“JP”固定	PrintableString
O	法人名または屋号	UTF8String
OU	代表者氏名：法人代表者氏名	UTF8String
OU	法人所在地：法人登記所在地	UTF8String
OU	部門名：本支店名、事業所名等の部門名	UTF8String
OU	事業所所在地：本支店、事業所等の所在地	UTF8String
OU	コード：事業所コードまたは企業コード	UTF8String
T	肩書名	UTF8String
CN	加入者氏名	UTF8String

3.2.4　応用例

電子カルテサーバに保管された電子カルテのアクセス制御を行う例を図 3-2 に示す。PKC に hcRole 属性を設定しておけば、電子カルテシステムへの接続時に PKC を使用したクライアント認証を行うことにより、電子カルテシステム側でクライアントの資格を確認することができる。そしてアクセスを要求したクライアントの資格によって、電子カルテ上のアクセス可能な範囲を制御することが可能である。

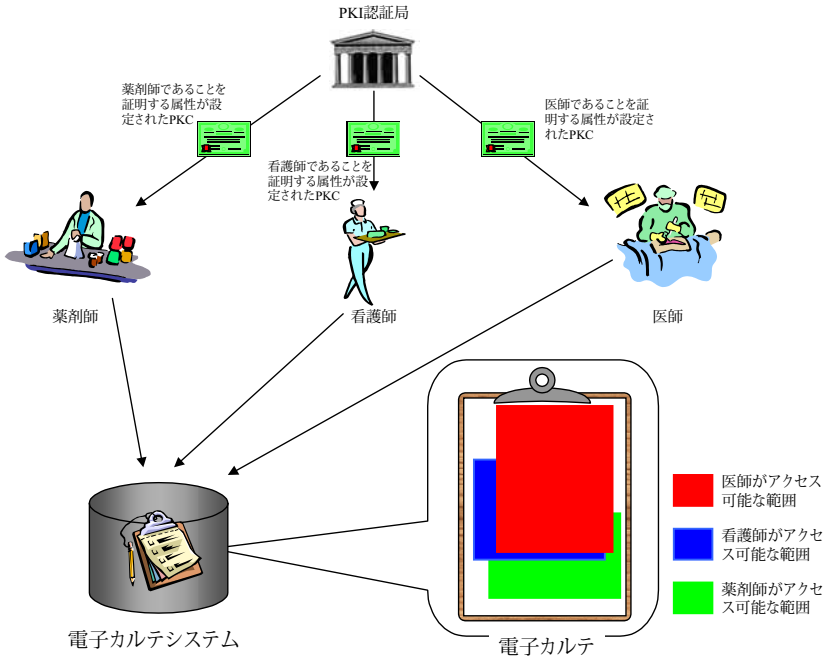


図 3-2 PKC に設定された属性を利用したアクセス制御の例

3.3　属性証明書

システムが主体(＝システムの利用者)の属性を認証するための情報を得る手段として、属性証明書(AC: Attribute Certificate)を利用する方法である［X509］［RFC3281］。属性証明書のフォーマットは ITU-T Recommendation X.509(08/97)で初めて規定され、X.509(03/00)でより詳細に再定義された。AC を利用するシステムの場合、システムは PKC を保有する利用者に対して、利用者ごとに付与する属性及び属性値を記載した AC をあらかじめ発行する。AC は複数の属性及び属性値を取り扱うことができるので､必要に応じて複数の属性及び属性値を一つの AC に記載しても良いし、複数の AC に分けて記載しても良い。PKC によって利用者を認証し、検証したい属性を記載した AC を用いて属性及び属性値を検証する。

3.3.1　特徴

AC を利用した属性認証には以下の特徴がある。

① 属性の有効期間を明示的に設定管理することができる。
② 属性認証局(AA: Attribute Authority)間の属性付与権限の移譲などにより、属性の分散管理を行いやすい。また、属性付与権限の移譲先を明示的に制限できる。
③ 属性及び属性値の改ざん防止がデジタル署名によって保証される。
④ 秘匿する必要がある属性を暗号化して設定する機能が標準的に用意されている。
⑤ AC は PKC を参照しており、PKC が存在することを前提としている。
⑥ 実際に利用する際には AC に独自の属性も含めて複数の属性が格納されている可能性がある。しかし、そのような AC の、属性を検証する汎用的検証ソフトウェアを予め用意するのは容易でない。市販の検証ソフトウェアを利用するにしても、アプリケーションに応じてカスタマイズ開発が必要になる可能性が高い。
⑦ 相互運用性の高いシステムやグローバルなシステムを構築するには、属性及び属性値(ならびにそれらに対応するオブジェクト ID)の業界標準化／国際標準化が必要であるが、現状では不十分である。
⑧ AC を用いる技法は、利用者同士のデータのやり取りで一方の利用者が他方の利用者の属性を検証するモデル(End-to-End モデル)のアプリケーション構築に向いている
⑨ 属性／属性値が変更になった場合は AC を発行しなおす必要がある。
⑩ AC を用いる技法では、複数の AC の中から使用する AC を選択する操作が必要になる可能性があり、他の技法と比較してその分操作性が低下する恐れがある。
⑪ 速度性能については、AC を用いる技法では、PKC の検証と AC の検証の両方が必要になるので、比較的遅いと考えられる。

3.3.2　属性認証局の属性証明書発行モデル

基本的な AA のモデルの考え方として、RFC 3281 をベースとするモデルと、ITU-T 勧告 X.509 をベースとするモデルが存在する。それぞれのモデルに登場する当事者は以下の通りである。

- RFC 3281 ベース
  - 属性認証局(AA)
  - 認証局(CA)
  - 属性証明書保有者(AC holder)
  - 属性証明書検証者(AC verifier)

- ITU-T 勧告 X.509 ベース
  - 属性認証局(AA)
  - ルート属性認証局(SOA：source of authority)
  - 認証局(CA)
  - 属性証明書保有者(AC holder)
  - 属性証明書検証者(AC verifier)

AA のモデル構成は、ベースとなるモデルが RFC 3281 か X.509 かにより異なる。それぞれの概念モデルを図に示す。図中の実線矢印は、PKC または AC の発行を意味する。

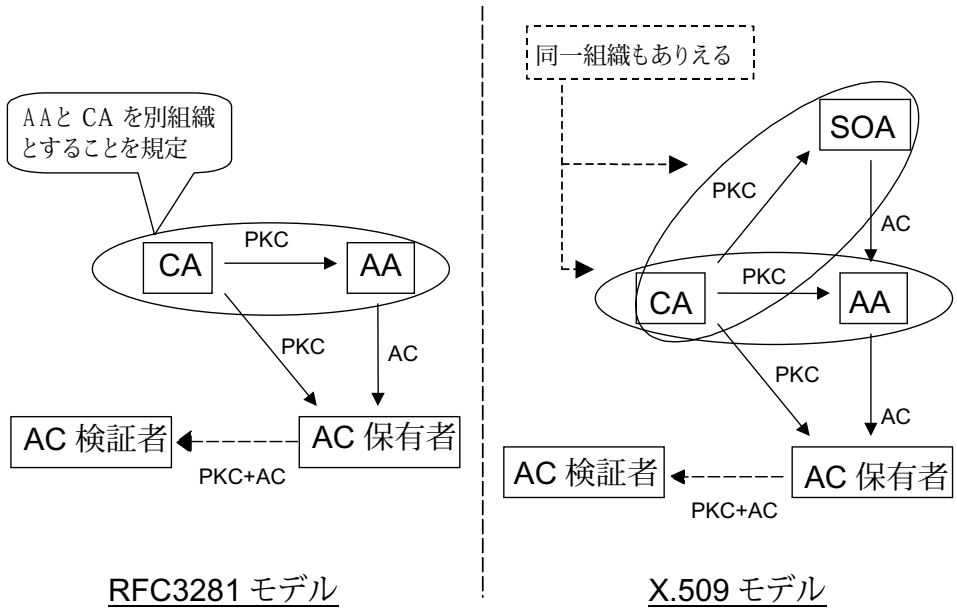


図 3-3 属性証明書発行モデル

RFC 3281 をベースとするモデルでは、PKI におけるルート認証局に相当する SOA(source of authority) と呼ばれるルート属性認証局は存在せず、属性認証局間の権限の委譲(delegation) が存在しないシンプルなモデルを提供している。さらに、CA と AA は異なる機関でなければならないとしている。また、AA と AC 保有者のそれぞれに対する CA が存在するモデルも考えられる。

一方、X.509 をベースとするモデルでは SOA が存在する。SOA は、属性証明書保有者(AC holder)

に付与される属性の最終的な責任者である。さらに、CA と AA は異なる機関であることを推奨しているが、強制はしていない。また、SOA、AA、AC 保有者のそれぞれに対する CA が存在するモデルも考えられる。

両モデルにおける共通事項として､AA と AC 検証者が同じであるモデルも考えることができる。

3.3.3 属性認証局の運用手順概略

属性認証の利用要求が生じた場合、AC 保有者は当該属性を証明する AC の発行依頼を行う。また、企業内の利用における新入社員への属性付与のように、AC 保有者の申請なしに企業側が一律に発行することもある。

(1) AC 保有者の申請に基づいて、AC を作成、及び発行

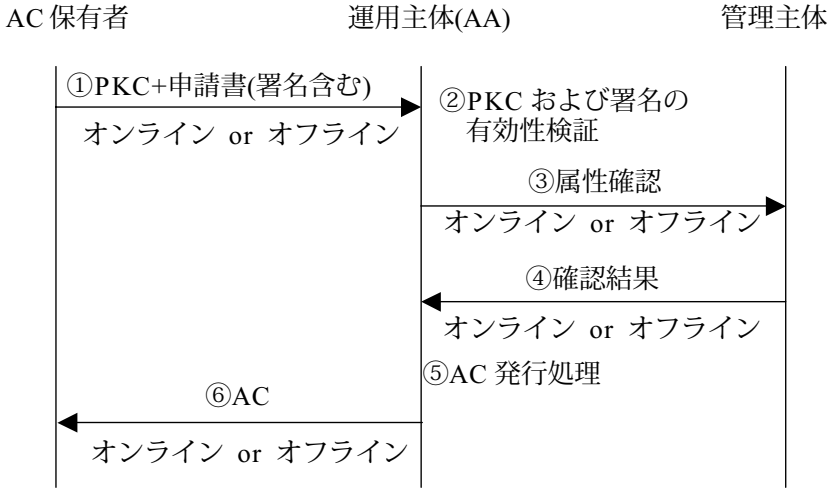


図 3-4 AC 保有者の申請に基づく属性証明書の発行フロー

① AC 保有者は、AA に対して、PKC と、その PKC に対応した秘密鍵による署名が付与された申請書を AA に送付する。
② AA は、PKC および申請書に対する署名の検証を行う。
③ AA は、②が OK の場合、属性の管理主体に申請された属性の確認を行う。
④ 属性の管理主体は、確認結果を AA に返す。
⑤ AA は、④が OK の場合、AC 発行処理を行う。
⑥ AA は、AC 保有者に AC を送付する。

(2) 管理主体の申請に基づいて、一律に AC を作成、及び発行

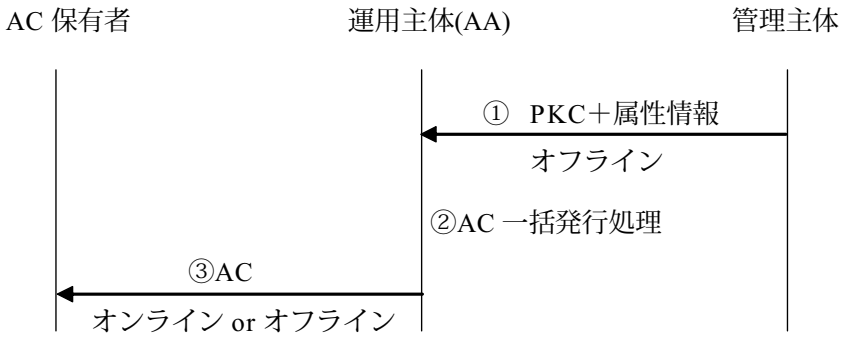


図 3-5 管理主体の申請に基づく属性証明書の発行フロー

① 管理主体は、AA に対し、AC 発行対象者の PKC および属性情報を送付する。
② AA は、AC 一括発行処理を行う。
③ AA は、状況に応じて AC を AC 保有者に提供する。

(3) AC の利用

本節では、AC の発行を受けた AC 保有者が AC を利用して自身の資格や権限を証明するための手順について述べる。AC の利用方法には push 型と pull 型のモデルがあり、一般的には push 型モデルが主流である。例外的な利用方法として、本人の関知しないところで AC がサーバで集中管理されており、AC 検証者はそのサーバにアクセス可能であるという利用モデルも考えられ、そのような場合には pull 型モデルが適する。

push 型とは、AC 保有者が、AC 検証者に対して AC を送り付けることにより、AC 保有者の資格や権限を確認するモデルである。従って、AA によって発行された AC は、AC 保有者自身が保持(管理) していることが、push 型モデルの前提となる。

pull 型とは、AC 検証者が、資格・権限の確認に必要な AC をリポジトリ等から取り寄せることにより、AC 保有者の資格や権限を確認するモデルである。従って、AA によって発行された AC は、AC 検証者がアクセス可能なリポジトリによって保持(管理) されていることが、pull 型モデルの前提となる。

属性証明書利用フローを図に示す。

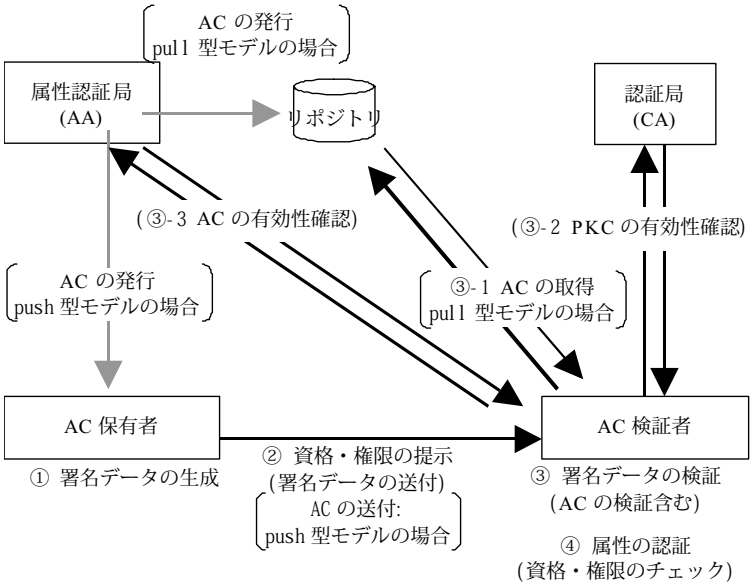


図 3-6 属性証明書利用フロー

① AC 保有者は、資格・権限を提示するために必要な署名データを生成する。すなわち、AC 保有者は、AC 保有者の資格・権限を記載した AC(pull 型の場合 AC 自体は含まず、AC を保持するリポジトリの所在を示す情報)、および、AC 保有者の PKC を含む形で署名データを生成する。当該署名データの生成には、AC 保有者の秘密鍵を使用する。

② AC 保有者は、資格・権限を AC 検証者に提示する。すなわち、①で生成した署名付きデータを AC 検証者に送信する。

③ AC 検証者は、AC 保有者から受信した署名データを検証する。署名データの検証には、AC 保有者の AC の検証も含まれる。pull 型の場合は AC 保有者の AC を取得するために、署名データに含まれる AC の所在を示す情報を元に、AC の取得を行う(③-1)。次に AC 保有者の PKC や、AA の PKC の検証が行うが、必要に応じて CA に対して有効性確認を行う(③-2)。その後 AC の検証を行うが、AC が失効することがある場合には、AA に対して AC の有効性確認を行う。具体的には、ACRL(Attribute Certificate Revocation List)の取得や OCSP(Online Certificate Status Protocol)による有効性確認を行う。

④ AC 検証者は、AC に記載されている属性(資格・権限等)と、AC 検証者で管理しているルールやアクセス制御ポリシーとを比較し、当該 AC を提示した AC 保有者がサービスを受ける資格・権限があるかどうかを判断する。

(4) 属性証明書の失効

AC 保有者の秘密鍵の危殆化、属性の変更などが生じた場合、AC 保有者の安全性確保、サービス品質の維持のため、その AA が発行した AC の失効を要することも想定される。また、AC は、その失効理由、AC 有効期間など個々の要素およびそれらの組み合わせにより失効を要しない場合も存在する。たとえば、AC の有効期間をごく短くし、有効期限を迎えるたびに

自動的に AC の更新を行うことによって、時間的にあまり変化しない属性を AC によって証明するようなシステムの場合、属性の変更などが発生した場合には、単に AC の自動更新を停止すればよいからである。

3.3.4 属性証明書利用上の留意事項

PKC を使用した属性認証の場合とは異なり、AC を使用する場合には、属性の検証者が CA の他に AA も信頼する必要がある。属性の検証者が AA を信頼する根拠として利用できる方法としては、

- 属性証明書ポリシー(ACP)/属性認証局実施規定(ACPS)を確認した上で信頼可能かどうか判断する。
- SOA の PKC に設定された sOAIdentifier エクステンションにより、CA に対する信頼の延長として AA を信頼する。
- AA の PKC に設定された AAControls エクステンションにより、AA が AC に設定可能な属性を確認する。
- acceptablePrivilegePolicies エクステンションや acceptableCertPolicies エクステンションにより、AC に設定されている属性と、AC や AC を発行した AA の PKC に設定されている発行ポリシーとの間に矛盾が無いことを確認する。

などがあり、実行時に機械的に検証可能な部分の検証のほかに、事前の書類審査なども含む包括的な確認が必要である。

3.3.5 応用例

法人による電子申請において、申請業務の代理人に法人代表者が AC を委任状として発行する。代理人は自身の PKC とともに法人代表者から発行を受けた AC を申請先に送付することにより、法人の代理人であることを申請先に示す。AC には法人が委任する申請業務の範囲が属性として設定され、委任期間が AC の有効期間として設定される。

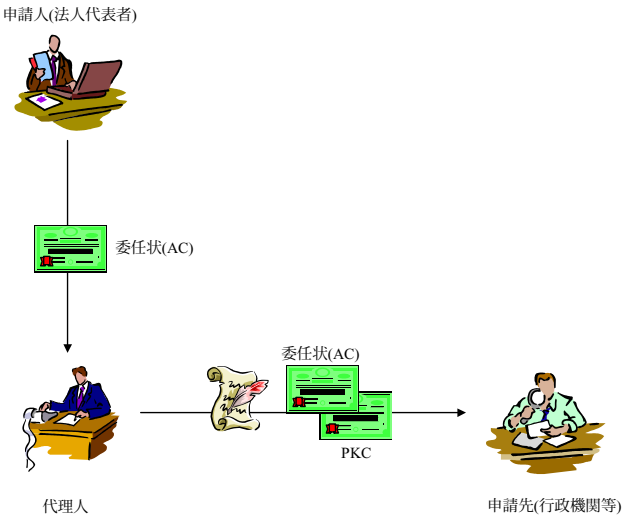


図 3-7 AC を委任状として利用する代理人システムの例

3.4　データベース管理

システムが主体(＝システムの利用者) の属性を認証するための情報を得る手段として、主体の属性値が格納されたデータベースを利用する方法である。システムは最初に主体を識別した際に、主体に関するレコードをデータベースに作製し、そのレコードに主体の属性値を格納する。データベースには複数の属性及び属性値を格納することができる。PKC などで利用者を認証し、データベースからその利用者の検証したい属性を読み出して、属性値を検証する。

3.4.1　特徴

データベースに格納された属性を使用して行う属性認証には以下の特徴がある。

① 属性データはデータベースに格納する方法の他、ディレクトリサーバに格納する方法などがある。
② 複数の属性の追加や属性値の変更などを集中管理しやすい。
③ 属性や属性値のやり取りの方法や、管理方法などが標準化されておらず、相互運用性に欠ける。
④ 一般にアイデンティティ認証の方法に独立している。

3.4.2　データベースによる属性管理のモデル

属性データをデータベースで管理する場合の、属性認証の利用形態を示す概念モデルを図に示す。このモデルではアイデンティティ認証に PKC を利用することとしている。また、サービス提供者と属性データベースは同一組織が運用する場合と、別組織が運用する場合の両方を想定しているが、基本的な動作はどちらであっても変わりは無い。

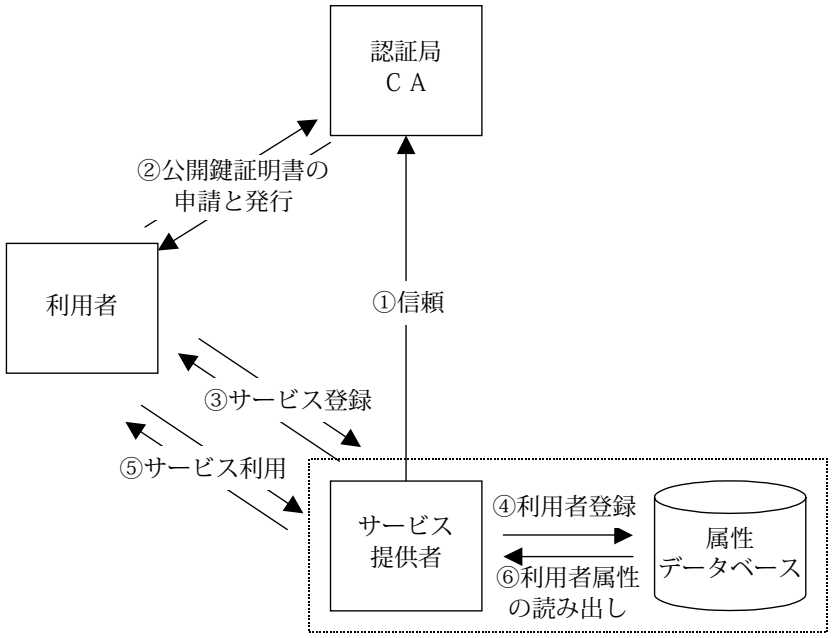


図 3-8 データベースによる属性管理モデル

① サービス提供者はあらかじめ信頼できる CA を見つけ、サービス提供者が利用者認証に使用する PKC を発行する CA として決めておく。
② 利用者はサービス提供者が決めた CA から PKC の発行を受ける。
③ 利用者はサービス提供者に、②で発行を受けた PKC と、利用者の識別を行うために必要な属性を提示して、利用者登録を行うよう求める。
④ サービス提供者は③で利用者から提示を受けた情報で利用者を識別し、利用者から提示された PKC を、信頼する CA の証明書で検証して問題がなければ、利用者の PKC と、その他の属性を属性データベースに登録する。PKC は利用者認証に、その他の属性は利用者の権限を決定するために使用する。
⑤ 利用者が署名と自身の PKC を提示して、サービス提供者のサイトでサービスを要求する。
⑥ サービス提供者は利用者から提示された署名を利用者の PKC で検証し、さらにその PKC を属性データベースで検索することによって、利用者認証を行う。そして属性データベース上の利用者のレコードから、利用者の権限を決定するために必要な属性を取り出して確認し、利用者がサービスを利用することを許可または拒否する。

3.4.3 応用例

セキュア Web アクセスサーバに対するアクセス制御を、SSL クライアント認証によるクライアントのアイデンティティ認証と、属性データベースに登録されたクライアントの属性によって実現する。セキュア Web アクセスサーバが、属性データベースに登録されたクライアントの属性によって、クライアントにアクセスを許すサービスの範囲を制御する。セキュア Web アクセスサーバは図 3-9 のように他のサーバから提供されるサービスへのポータルとして構成することもできるし、セキュア Web アクセスサーバ自身がサービスを提供してもよい。

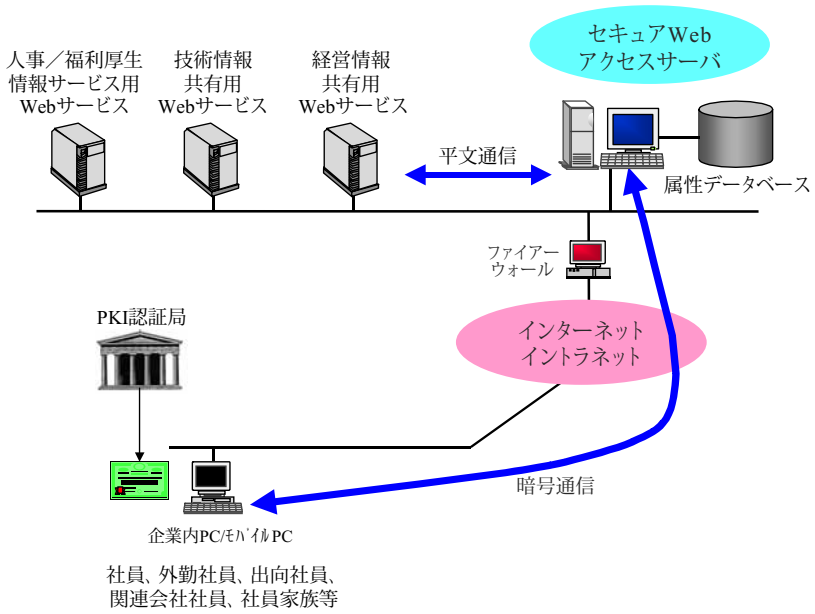


図 3-9 属性データベースを利用した Web サーバアクセス制御の例

3.5 SAML、Liberty、WS-Federation

システムが主体(＝システムの利用者)の属性を認証するための情報を得る手段として、SAML(Security Assertion Markup Language)やSAMLを利用するLiberty、あるいはWS-Federationを利用する方法がある。

SAML は非営利の国際コンソーシアムである OASIS(Organization for the Advancement of Structured Information Standard)が策定する、セキュリティ情報交換のためのXMLベースのフレームワークで、現在のバージョンは1.0であるが、バージョン2.0の策定中である。

また、Libertyは非営利の150以上の企業によるアライアンスであるLiberty Allianceが策定する、連携されたアイデンティティ管理と分散的な認証・認可を実現する技術標準で、仕様の一部にSAMLを拡張して使用している。なおSAMLのバージョン2.0ではLibertyが拡張した仕様の一部が取り込まれている。

WS-Federationは Microsoft、IBM、VeriSign、BEA、RSA Securityの5社が2003年7月8日に発表し、仕様を公開した。WS-Federationは2002年4月にIBMとMicrosoftが示したWebサービスセキュリティのロードマップの中で「WS-Policy」「WS-Trust」「WS-SecureConversation」の仕様に続くものであり、どのようにしてセキュリティ的に異質な環境同士の信頼関係を管理し仲介するか(アイデンティティ連携のサポート／属性の共有／仮名の管理を含む)を定めている。なお、IBM、MicrosoftはWS-Federation仕様を、近い将来標準化団体に提出することを表明している。

3.5.1 SAMLの概要

(1) SAMLの特徴

多くの関連するWebサイトやページにアクセスする時、それぞれから独立した認証を求められるのでは極めて面倒であるばかりではなく、それぞれのIDやパスワードを個別に覚えておくことは難しい。シングルサインオン(SSO)はWebページやサイトの連携をはかるために必須な機能といえよう。

SAMLの第一の目的は柔軟でかつセキュリティの強固なSSOを実現することである。しかし、実際のセキュリティアプリケーションでは、認証の後に利用者の資格などの属性によってアクセスできるページやWebサイトを制限したり、また与えられたアクセス権限によりリソースへのアクセスを制御したりする、いわゆる認可サービスが必要である。SAMLは認証情報伝達サービス(Authentication Assertion)に加えて2つの認可サービスが加えられている。1つは属性情報の伝達(Authorization Assertion)であり、もう1つはアクセス制御情報の伝達(Authorization Decision Assertion)である［SAML］。

SAML仕様はPKIのセキュリティ環境の利用のみを前提にしたものではなく、セキュリティ

を強化したID/パスワードから対称鍵による認証も含む幅の広いものとなっている。しかし、SAMLにPKIを導入することによって強力なセキュリティを持つSSOと柔軟で強力な属性制御を実現できるのである。

SAML標準のもう1つの目的は、異なるベンダのアクセス制御製品間の相互運用性を推進することである。現状では異なるベンダ製品で既に大規模に展開されたサイト間でのSSOを実現することが難しい。しかし、SAML標準を実装したアクセス制御製品間では連携SSOが可能となる。Liberty Alliance仕様もこのような環境の実現を目指している。

(2) SAMLのモデル

SAMLはセキュリティ情報交換のためのXMLベースのフレームワークである。SAMLは要求と応答のプロトコルと応答に含まれるアサーションの構文仕様を定めたものである。このセキュリティ情報は対象とする主体(Subject：人またはコンピュータ)のあるセキュリティドメインにおける認証情報や属性情報や認可情報をSAMLの規格の中で定義しているアサーションと呼ぶデータ形式で表現する。アサーションはSAMLオーソリティが発行し、主体の名前／属性／主体がアクセスを認可されたリソースなどが、SAMLオーソリティのデジタル署名により改ざん不可能な形式で含まれている。

SAMLオーソリティには以下の3つのオーソリティがある。

- 主体のアイデンティティ認証を行い、認証情報の再利用(SSO)を可能にする「認証アサーション」を発行する「認証オーソリティ」
- 認証オーソリティが発行した認証アサーションを入力として、ポリシーに基づいて主体に付与された資格や役職などの属性を記述した「属性アサーション」を発行する「属性オーソリティ」
- 認証アサーションと属性アサーションを入力として、主体が求めるリソースへのアクセスを認めるか否かポリシーに基づいて決定し、「認可決定アサーション」を発行する「認可決定オーソリティ」

図3-10にSAMLのモデルを示す。図に示すようにSystem Entity(主体のクライアントまたはクライアントからアクセス要求を受けたサーバ)は、まずSAML認証オーソリティにSAML認証要求としてクレデンシャル(パスワードや鍵情報など)を示す。認証オーソリティは認証ポリシや外部の認証環境(PKIなど)を検証して、主体の本人性(Identity)を証明する認証アサーション(Authentication Assertion)を発行する。次にこの認証アサーションを属性オーソリティまたは認可決定オーソリティ(PDP: Policy Decision Point)に示して認可権限を問合わせる。属性オーソリティはポリシーに沿った主体の資格などの属性アサーションを発行し、認可決定オーソリティに主体のアクセス権限を問合わせる。認可決定オーソリティは予め登録された認可権限をポリシデータベースから検索し認可決定アサーションを発行し、実際にアクセス制御を行うPEP(Policy Enforcement Point)に渡しアプリケーション要求に沿

ったWebページへのアクセスやリソースへのアクセスを実行させるのである。

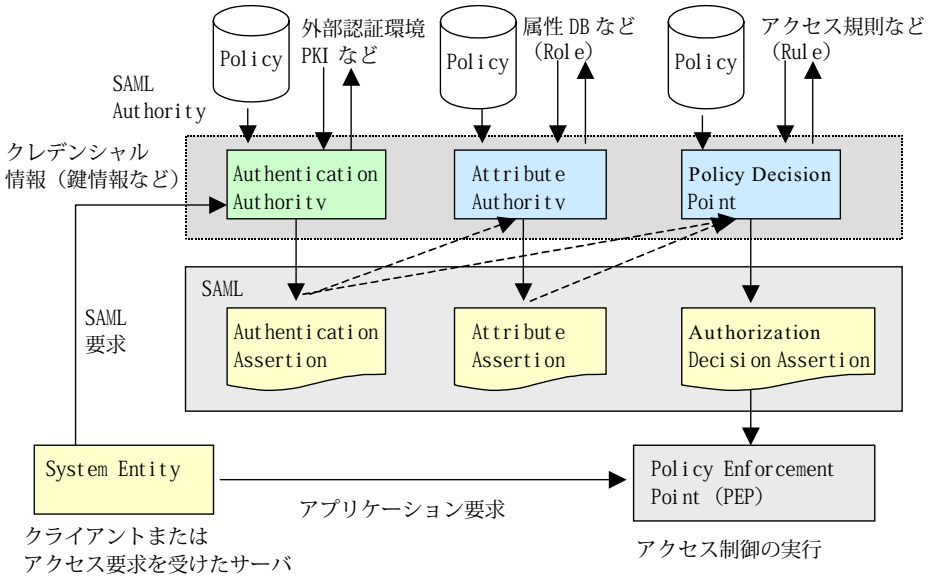


図 3-10 SAMLのモデル([SAML]より引用)

図に示した3つのSAMLオーソリティは必ずしも全て必要ではない。SSOを実現するだけなら、認証オーソリティだけを用いればよい。しかし、属性や認可制御を木目細かに制御するためには、認証オーソリティの認証アサーションを属性オーソリティや認可決定オーソリティに提示し認可決定のアサーションを受ける必要がある。

3.5.2 Liberty Allianceの概要

Liberty Alliance Projectは、インターネット上での新しい水準の信頼、商取引、通信を推進するために結成された幅広い産業が集まった団体である。Libertyはアイデンティティ情報のプライバシやセキュリティを考慮した、仮想的に取引参加可能なネットワーク化された世界を目指し、その実現のために、ネットワークアイデンティティに基づく広範なやりとりをサポートするオープンな技術仕様を策定するとされている。ネットワークアイデンティティとは、現状さまざまなサービス毎に分散して管理されているユーザアカウントに設定された属性のグローバルセットである。

Libertyの目標は、以下の4つ。

- 利用者がネットワークアイデンティティ情報のプライバシとセキュリティを保護できること。
- 企業が第三者の介入を受けることなく、顧客との関係を維持し、管理できること。
- 複数ベンダからの分散型の認証と許可を行えるオープンなシングルサインオンの規格を提供する。
- 現行および新規の全てのネットワークアクセスデバイスをサポートするネットワークIDインフラを構築する。

この目標のために以下の3フェーズで仕様の策定を進めている。

表 3-5 Libertyの仕様策定フェーズ

フェーズ	内容	時期
ID-FF (Identity Federation Framework)	シングルサインオンとアイデンティティ連携	～2003年頭
ID-WSF (Identity Web Services Framework)	アイデンティティサービスのためのWEBサービスフレームワーク	～2003年中頃
ID-SIS (Identity Services Interface Specification)	アイデンティティ連携とWEBサービス関連付加価値サービスの検討	～2004末

［LIBERTYTUTORIAL］［LIBERTYARCHOVERVIEW］

(1) 構成要素

LibertyのID-FFはSAMLを元にシングルサインオンとアイデンティティ連携を実現するための仕様である。ID-FFにおける主要な構成要素は以下のとおりである。

表 3-6 Liberty ID-FFの主要な構成要素

要素	説明
IdP	Identity Providerのこと。ユーザのアイデンティティ認証や属性認証を行い、その結果をSAMLアサーションの形式でSPに提供する。
SP	Service Providerのこと。ユーザのアイデンティティをIdPから提供されたアサーションによって確認し、それに基づいてユーザに対してサービスを提供する。
ユーザ	SPが提供するサービスを利用するユーザ。SPのサービスを利用するために予めIdPにログインする。通常ユーザエージェントを通してIdPやSPと通信を行う。
ユーザエージェント	ユーザがネットワークにアクセスするために使用するS/Wやデバイスなど。WWWブラウザやLiberty対応機能を持つクライアントなどを含む。

(2) トラストサークル

Libertyの動作を説明する上で避けられないものとしてトラストサークルがある。トラストサークルは、複数のIdPとSPが形成する信頼関係のことで、技術的なデータのやり取りでの信頼関係以外に、企業間の契約や運用協定などの法的な側面も含む企業間の連携を指す。ユーザはトラストサークル内のIdPにログインすることによって、そのトラストサークル内のSPから、そのSPのポリシーに基づいてサービスを受けることができる。

(3) シングルサインオンとアイデンティティ連携

Libertyのシングルサインオンとアイデンティティ連携は、シングルサインオンおよび連携プロトコル(Single Sign-On and Federation Protocol)によって行われる。ユーザはIdPに登録されている自身のアイデンティティと、SPに登録されているアイデンティティをあらかじめ連携させておいてシングルサインオンすることもできるし、アイデンティティ連携とシングルサインオンを同時に行うこともできる。また、過去に連携したアイデンティティの

連携を解除することもできる。基本的にどのプロバイダに登録されているアイデンティティ同士を連携させるか決めるのはユーザである。システムはアイデンティティ連携を行える環境を提供する。

基本的なシングルサインオンのシーケンスを以下に示す。

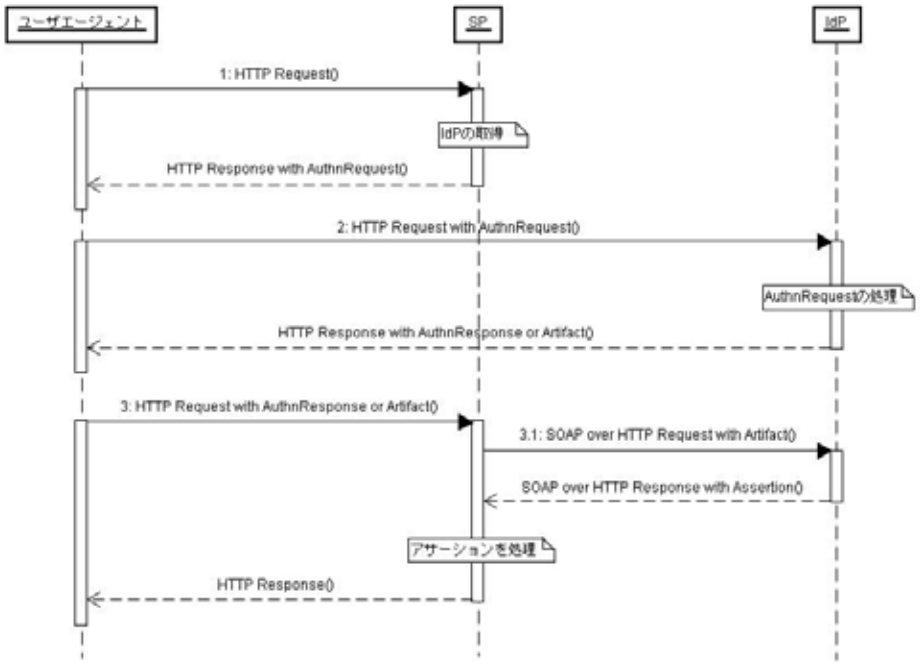


図 3-11 シングルサインオンのシーケンス

ここで SP と IdP はユーザエージェント(ブラウザ)を仲介として通信を行っているが、その為の方法として以下の三つのプロファイルが定められている。

① Liberty ブラウザのアーティファクト・プロファイル

Liberty ブラウザのアーティファクト・プロファイルでは、SAML アーティファクトが埋め込まれた URI(Uniform Resource Identifier)を使用した HTTP のリダイレクションによって、IdP と SP 間でメッセージを交換することが規定されている。SAML アーティファクトとは、SAML で規定されている個々のアサーションを識別するための一種の識別子である。SP が SAML アーティファクトを使用して IdP に SAML アサーションを要求するためには、SP と IdP 間の直接通信が必要となる。この方式の利点は、SAML アーティファクトのサイズが小さいため、SAML アーティファクトを URI エンコードしても URI のサイズ制限を心配することなく埋め込めることである。

② Liberty ブラウザの POST プロファイル

Liberty ブラウザの POST プロファイルでは、JavaScript や ECMAscript をサポートするブラウザに、スクリプトによって自動的にポストされる FORM 要素が存在する HTML ページを送信して、HTML フォームの自動 POST によって IdP と SP 間でメッセージ交換することが規定されている。URI にメッセージを埋め込む場合に比べてサイズの大きなメッセージのやり取りができるため、SAML アーティファクトでなく SAML アサーションそのものを IdP から

SP に送ることができる。そのため SP と IdP 間で直接通信する必要がない。ただし、HTML フォームの自動 POST が可能なスクリプトを実装していないブラウザでは使用できない。

③ Liberty 対応クライアント及びプロキシのプロファイル

Liberty 対応クライアントは、ユーザが SP で使用する IdP を認識しているか、または認識する方法を知っているクライアントである。また、Liberty 対応クライアントは、HTTP リダイレクトに依存したり、URI にプロトコルパラメータをエンコードしたりするのではなく、POST を使用して HTTP リクエストおよび応答の本体で Liberty メッセージを送受信する。そのため、Liberty 対応クライアントには、Liberty プロトコルメッセージのサイズ制限が存在しない。

(4) シングルログアウト

シングルログアウトは SP、IdP のいずれかで開始される。その後、IdP はセッションを確立した SP 全てにログアウトを通知する。

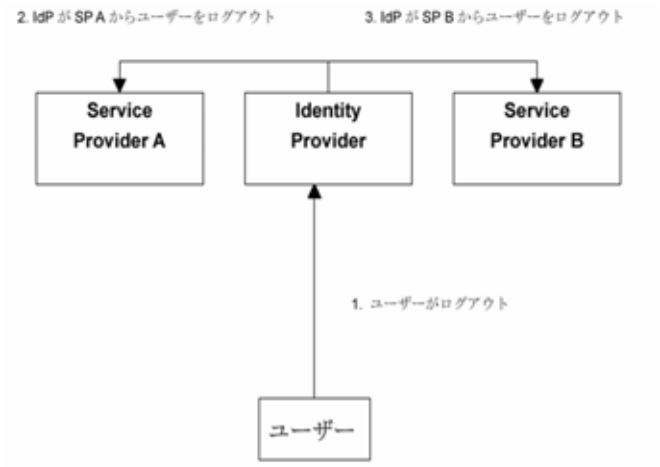


図 3-12 IdP からのシングルログアウト([LIBERTYARCHOVERVIEW]より引用)

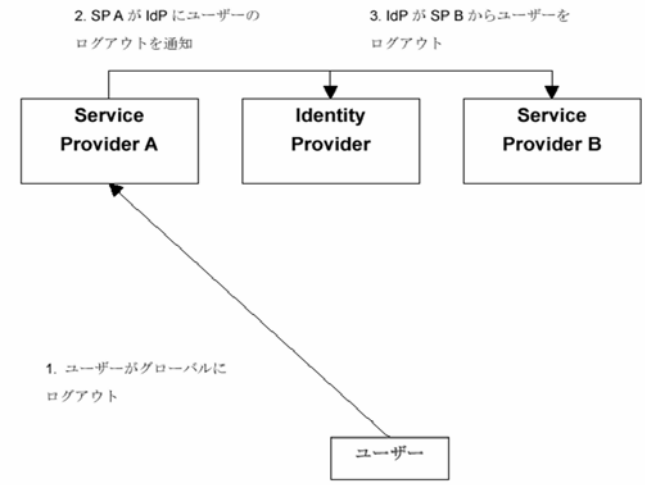


図 3-13 SP からのシングルログアウト([LIBERTYARCHOVERVIEW]より引用)

3.5.3 WS-Federation の概要

IBM と Microsoft が 2002 年 4 月に発表した WS-Security のロードマップの中で、セキュリティ的に異質な連携した環境の信頼関係を管理し仲介する方法について記述したもの、として紹介され、2003 年 7 月に仕様が公開された。WS-Security における、WS-Federation の位置づけは以下のとおりである［WSSROADMAP］。

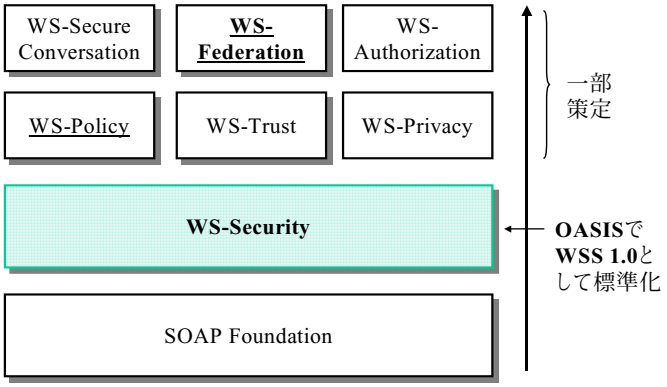


図 3-14 Web Service Security Roadmap([WSSROADMAP] より引用)

図 3-14 にある各仕様の概要は以下の通りである。

表 3-7 Web Service Security の仕様

仕様名	概要
WS-Security	SOAP メッセージのヘッダに、署名や暗号化データを付け加える手段を定める。また、PKC やケルベロス・チケットのようなバイナリ・セキュリティ・トークンを含む、セキュリティ・トークンを付け加える手段についても定める。
WS-Policy	エンドポイントと中継者のセキュリティポリシーの能力と制限(要求するセキュリティ・トークン、サポートされる暗号アルゴリズム、プライバシ・ルールなど) と、どのようにサービスやエンドポイントにポリシーを関連付けるかについて記述した仕様の集まり。
WS-Trust	Web サービスがセキュリティ・トークンを要求し、発行し、交換することによってセキュアに相互作用できるようにするためのフレームワークとトラストモデル。
WS-Privacy	Web サービスとリクエスタに、プライバシに関する選択や組織のプライバシ・プラクティス・ステートメントを適用する方法のモデルを記述する。
WS-SecureConversation	通信路での認証されたメッセージの交換をどのように管理するか(セキュリティ・コンテキストの交換と、セッションキーの確立と配布を含む) を記述する。
WS-Federation	連携アイデンティティや属性の共有、仮名の管理を含め、セキュリティ的に異質な連携した環境の信頼関係を管理し仲介する方法について記述する。
WS-Authorization	認証データと認証ポリシーを管理する方法について記述する。

WS-Security は OASIS で WSS(Web Services Security) 1.0 として 2004 年 4 月に標準化されている。WS-Federation の目標は以下のとおりである［WSFLANG］。

- アイデンティティ、認証、異なる／似通ったメカニズムを使用した認証データなどを適切に共有できるようにすること。
- 信頼とセキュリティ・トークンの交換を仲介すること。
- サービス提供者側でローカルにアイデンティティの管理をする必要をなくすこと。
- 必要に応じてアイデンティティ情報とその他の属性を隠蔽することができるようにすること。

一方以下の項目については WS-Federation の範囲外とされている。

- メッセージレベルのセキュリティの定義や、信頼の確立／検証を行うプロトコル。
- 新しいセキュリティ・トークンのフォーマット仕様。
- 新しい属性ストアのインタフェース仕様。
- セキュリティ・トークン・アサーションのフォーマット仕様。

WS-Federation は WS-Security、WS-Policy、WS-Trust を前提として仕様の策定がされており、WS-Federation を理解するためにはセキュリティ・トークン、IP/STS など WS-Federation の範囲外で規定されている概念についての知識が不可欠である。

(1) 構成要素

WS-Federation における主要な構成要素は以下の通りである。

表 3-8 WS-Federation の主要な構成要素

要素	説明
STS	Security Token Service のこと。STS はセキュリティ・トークンを発行する WEB サービスである。STS は自身が信頼する情報に基づいてセキュリティ・トークンを発行する。セキュリティ・トークンを生成するにあたっては、トークンを発行可能とするような、デジタル署名やセキュリティ・トークンなどの証拠が必要である。STS は自らセキュリティ・トークンを生成する場合もあるが、他の独立した STS が生成したセキュリティ・トークンに頼る場合もある。
IP	Identity Provider のこと。IP は STS の一種で、通信相手の認証を行ってアイデンティティ・クレイムを含むセキュリティ・トークンを発行する。
属性サービス	リクエスタに関する属性をリソースに提供するサービスである。リソースはリクエスタの属性を属性サービスから受け取り、それを用いてリクエスタのアクセス制御を行うことができる。
仮名サービス	属性サービスの一種で、ユーザの別名を管理する属性サービスである。
リソース	リソースはリクエスタにサービスを提供する。
リクエスタ	リクエスタはリソースが提供するサービスを利用する。パッシブリクエスタとアクティブリクエスタがある。パッシブリクエスタは一般的な HTTP ブラウザのことであり、アクティブリクエスタは WS-Security や WS-Trust に記述されているような Web サービスメッセージを発行する能力のあるアプリケーション(Web ブラウザを含む) を指す。

(2) セキュリティ・トークン

WS-Federation では、WS-Security の仕様の一つである SOAP Message Security 1.0 で規定されているセキュリティ・トークンをプロバイダ同士で通信することによって、アイデンティティ連携などの機能を実現する。セキュリティ・トークンはクレイムの集合である。クレイムはリクエスタが名前や権限、属性などについて主張する申告である。クレイムには信頼のおけるオーソリティによって保証されたものと、そうでないものがある。保証されたクレイムの集合には、通常オーソリティによるデジタル署名が施され、署名付きセキュリティ・トークンとして取り扱われる。WS-Security が OASIS で標準化されたため、セキュリティ・トークンについての標準も OASIS 作成されており、現在のところ、以下のセキュリティ・トークンについてのプロファイルが OASIS で標準化済み、または作業中である。[WSSECURITY]

表 3-9 WS-Security のセキュリティ・トークン

Username Token Profile [WSSUSERTOKEN]	ユーザ名とパスワードを使用するセキュリティ・トークンのプロファイル。	標準化済み (2004/03)
X.509 Token Profile [WSSX509TOKEN]	PKC を使用するセキュリティ・トークンのプロファイル。	標準化済み (2004/03)
SAML Token Profile [WSSSAMLTOKEN]	SAML アサーションを使用するセキュリティ・トークンのプロファイル。	標準化済み (2004/12)
REL Token Profile [WSSRELTOKEN]	ISO/IEC 21000-5 に規定されている Rights Expression Language(REL) のライセンスを使用するセキュリティ・トークンのプロファイル。	標準化済み (2004/12)
Kerberos Token Profile [WSSKERBEROSTOKEN]	ケルベロスのチケットを使用するセキュリティ・トークンのプロファイル	作業中

(3) アイデンティティ連携のモデル

WS-Federation では、各リソースにローカルに登録されたリクエスタのアイデンティティを連携させることによってシングルサインオンを実現することもできるし、リソースにリクエスタのアイデンティティが登録されていなくても、リクエスタをリソースにシングルサインオンさせることもできる。WS-Federation のアイデンティティ連携とシングルサインオンの構成にはさまざまなバリエーションが考えられるが、一例として図 3-15 に属性サービスと仮名サービスを使用したアイデンティティ連携の例を示す。図 3-15 ではユーザのドメインの IP/STS とサービス提供者のドメインの IP/STS との間には、あらかじめ信頼関係が結ばれていることを前提とする。

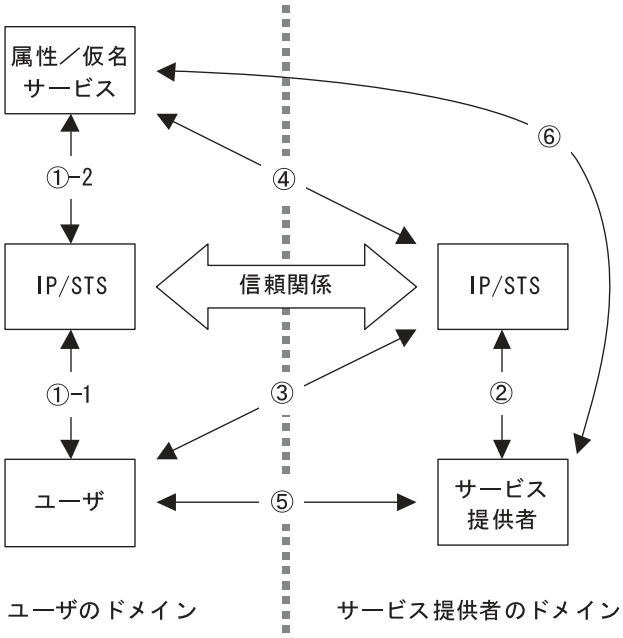


図 3-15 属性サービス・仮名サービス([WSFLANG]より引用)

① リクエスタが自身の IP からセキュリティ・トークンを取得する(①-1)。その際、セキュリティ・トークンにはリクエスタを識別する情報として仮名が設定されている。その仮名はリクエスタの仮名サービスに登録される。仮名サービスに、既にリクエスタのリソース側のアイデンティティとセキュリティ・トークンが登録されている(①-2)場合は、リソース側のセキュリティ・トークンが発行される。(その場合下記③と④は省略される)。

② リソースが自身のIPからセキュリティ・トークンを取得する。

③ リクエスタがリソースのドメインの IP に、①で取得したセキュリティ・トークンを示して、リソースのドメインのセキュリティ・トークンを取得する。この時、リソースのドメインの IP は、必要に応じてリクエスタのリソース側でのアイデンティティを求める場合がある。

④ リクエスタが許可した場合には、リソースのドメインの IP が、リクエスタのドメインの仮名サービスに、リクエスタのリソース側の仮名とセキュリティ・トークンを登録する。

⑤ リクエスタがリソースに③で発行されたセキュリティ・トークンを示してサービスの提供を求める。

⑥ 必要に応じてリソースはリクエスタの仮名サービスと属性サービスに②で取得したセキュリティ・トークンを示してアクセスし、リクエスタの属性を確認してリクエスタにサービスを提供する。

3.5.4 Liberty Alliance と WS-Federation の相違の概要

本節では［WSFEDLIBERTYOV］を参考に、Liberty Alliance と WS-Federation の相違につい

て概要をまとめた。

(1) 仕様項目の相違

仕様項目レベルでは、Liberty の連携フレーム(ID-FF、ID-WSF および ID-SIS)および WS-Federation とでは以下の点は同様のものとなっている

- 基本的なメッセージング・プロトコルのプロファイルを通じて、(これらのクライアントの性質は異なりそうだが)ブラウザおよびスマート・クライアントを識別すること
- セキュリティ・トークンの発行による信頼の仲介業務
- プライバシにコントロールされる属性共有
- 連携されたサインアウトによる基本的なセッション管理の動作

しかしながら、これらの原理を適用する方法は、アプローチおよび基礎技術において大部分が異なる。下記のマトリックスは、これらの違いをより詳細に示す。

表 3-3-10 Liberty/WS-Federation 仕様項目の相違

分類	コンポーネント	Liberty	WS-Federation +
リンク	アカウントのリンク	ID 連携フレームワーク	あり。仮名サービスのセットメッセージによる
シングルサインオン	認証リクエスト	SAML リクエスト	WS-Trust トークン発行リクエスト
	認証レスポンス	SAML レスポンス	WS-Trust トークン発行レスポンス
	アサーション	SAML 認証ステートメント	任意のトークン
	認証の詳細	認証コンテキストはリクエストとレスポンスに規定される場合あり	
	プロファイル	ブラウザアーティファクト、フォーム POST、LEC	バリエーションのある Passive 及び Active リクエスタプロファイル
セッション	IDP によって開始されるシングルログアウト	あり	あり
	SP によって開始されるシングルログアウト	あり	あり
	セッションのクレデンシャル		WS-SecureConversation
プライバシ	不透明識別子	あり	オプション(不透明ではない永続的な識別子が使用される場合あり)
	管理	NameRegistration プロトコル	あり。仮名サービスのセットメッセージによる
	公開される属性に関するポリシー	用途指示子	
	暗号化された識別子及び URI	あり	
認可	認可リクエスト	黙示的	WS-Trust
	認可レスポンス	黙示的	WS-Trust

分類	コンポーネント	Liberty	WS-Federation +
	属性(ロール)		あり
信頼	法律上の合意	認証コンテキストから参照可能	
	ビジネス上の合意	認証コンテキストから参照可能	
	提携	あり	
	イントロダクション	下記を参照	
	トークンの交換/マッピング		WS-Trust
セキュリティ	メッセージのセキュリティ	XML 署名/XML 暗号化 /WS-Security の保護されたメッセージ	XML 署名/XML 暗号化 /WS-Securityの保護されたメッセージ
メタデータ	発行	DNS および既知の場所。UDDI ディレクトリで発行される場合あり	
	取り出し	DNS および既知の場所	
	スキーマ	ID-WSF メタデータ	WS-MetadataExchange
ディスカバリー	主体者のIDP	共通ドメインクッキー	
	発行	ID-WSF DiscoveryLookupUpdate	UDDI
	照会	ID-WSF DiscoveryLookupRequest	UDDI
	セキュリティポリシー		WS-SecurityPolicy
イントロダクション	信頼の仲介	あり	WS-Trust
	主体者の連携の通知	あり	
	信頼の終了の通知	あり	
情報の共有	アクセス	あり。アイデンティティサービス	属性サービス
	保管		UDDI
	プライバシーポリシー	プライバシーポリシー記述言語	WS-Privacy か？
	データ操作	WSFデータサービステンプレート	WS-Federation ではなし。.Net My Services HSDL では可能性あり
	データのインタフェース	ID 個人プロファイル	WS-Federation ではなし。.Net My Services HSDL では可能性あり
	仲介	あり	
ユーザのインタラクション	ユーザの承諾	ID-WSF インタラクションサービス	
	連携の終了	あり	

(2) プロトコル上の差異の例

① Linkage

- Liberty Alliance Project：
  ID-FF に基づく。
  認証方式は限定せず、アカウント連携、仮名、匿名のアイデンティティ連携を行う。
- WS-Federation：

(WSF) Pseudonym Service(仮名サービス)
仮名サービス取得プロトコルメッセージにより、トラストサークルや仲介を経由したアイデンティティ連携

② Request & response & assertion

両者ともよく似た方式であるが、Liberty Alliance では SAML のみをベースにしており、WS-Federation では、WS-Trust(Kerberos, SAML, X509v3, XrML)を利用している。

認証サイトが Liberty Alliance では IdP(LAP)サイトであるのに対して、WS-Federation の場合は SecurityTokenService を動作させている Policy サーバでのサービスレベルである点が異なる。

- Liberty Alliance Project：

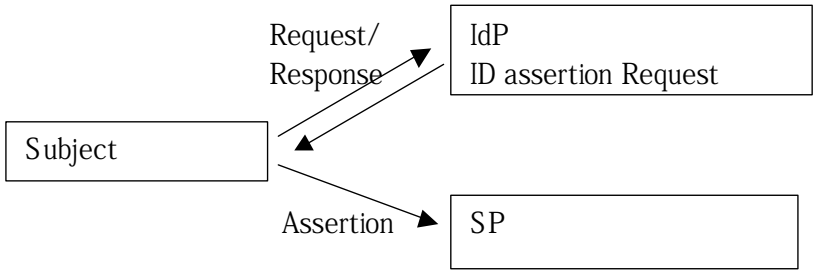


図 3-16 Liberty における Request/Response/Assertion

IdPにAuthenticate対する要求

```
<samlp: Request ...>
  <samlp: AttributeQuery>
    <saml: Subject>
      <saml: NameIdentifier
        SecurityDomain="ecom.jp"
        Name="rimap"/>
    </ saml: Subject>

    <saml: AttributeDesignator
      AttributeName="Employee_ ID"
      AttributeNamespace="ecom.jp">
    </ saml: AttributeDesignator>
  </ samlp: AttributeQuery>
</ samlp: Request>
```

SAMLAuthenticationレスポンス

```
<samlp: Response
  MajorVersion="1" MinorVersion="0"
  RequestID="128.14.234.20.90123456"
  InResponseTo="123.45.678.90.12345678"
  StatusCode="Success">
```

SAMLAuthenticationアサーション

```
<saml: Assertion
    MajorVersion="1" MinorVersion="0"
    AssertionID="123.45.678.90.12345678"
    Issuer="Ecom.jp"
    IssueInstant="2002- 01- 14T10: 00: 23Z">

    <saml: Conditions
      NotBefore="2002- 01- 14T10: 00: 30Z"
      NotAfter="2002- 01- 14T10: 15: 00Z" />

    <saml: AuthenticationStatement
    AuthenticationMethod="Password"
    AuthenticationInstant="2001- 01- 14T10: 00: 20Z">

      <saml: Subject>
        <saml: NameIdentifier
          SecurityDomain="ecom.jp"
          Name="rimap" />
      </ saml: Subject>
    </ saml: AuthenticationStatement>
  </ saml: Assertion>
</ samlp: Response>
```

- WS-Federation(WS-Trust Token Issuance Request)[WSTRUST]

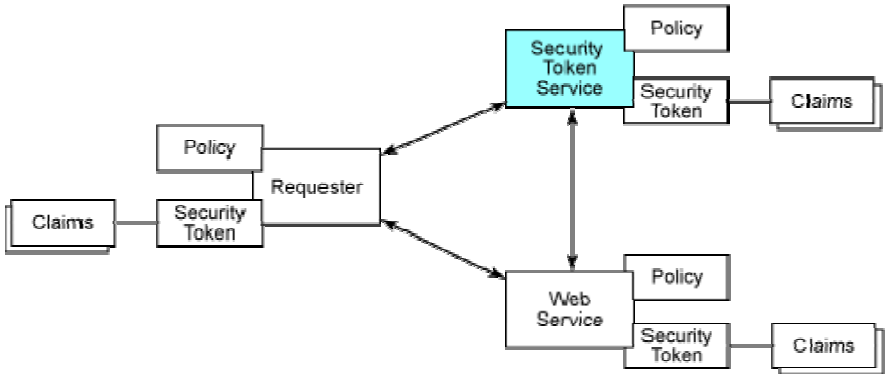


図 3-17 WS-Federationにおける Request/Response/セキュリティ・トークン

Security Tokenサービスに対する要求

```
<S:Envelope xmlns:S="..." xmlns=".../secext" xmlns:wsu=".../utility>
  <S:Header>
    ...
    <Security>
      <UsernameToken wsu:Id="myToken">
        <Username>NNK</Username>
        <Nonce>FKJh...</Non ce>
        <wsu:Created>2001 -10-13T09:00:00Z </wsu:Created>
      </UsernameToken>
      <ds:Signature xmlns:d s="...">
        ...
      </ds:Signature>
    </Security>
    ...
  </S:Header>
  <S:Body wsu:Id="req">
    <RequestSecurityToken>
      <TokenType>wsse:X509v3</TokenType>
      <RequestType>wsse:ReqIssue</RequestType>
      <Base>
        <Reference URI="#myToken"/>
      </Base>
    </RequestSecurityToken>
  </S:Body>
</S: Envelope>
```

セキュリティ・トークンResponseメッセージ

```
<S:Envelope xmlns:S="..." xmlns=".../secext" xmlns:wsu=".../utility">
  <S:Header>
    ...
    <Security>
      <BinarySecurityToken
       wsu:Id="myToken2"
       ValueType="wsse:X509v3"
       EncodingType="wsse:Base64Binary">
       DFJHuedsujfnrnv45JZc0...
      </BinarySecurityToken>
      <ds:Signature xmlns:ds="...">
        ...
      </ds:Signature>
    </Security>
    ...
  </S:Header>
  <S:Body>
  <RequestSecurityTokenResponse>
    <RequestedSecurityToken>
      <BinarySecurityToken
       ValueType="wsse:X509v3"
       EncodingType="wsse:Base64Binary">
       MIIEZzCCA9CgAwIBAgIQEmtJZc0...
      </BinarySecurityToken>
    </RequestedSecurityToken>
    <RequestedProofToken>
      <ds:KeyInfo xmlns:ds="...">
        <ds:KeyValue>
          ...
        </ds:KeyValue>
      </ds:KeyInfo>
    </RequestedProofToken>
  </RequestSecurityTokenResponse>
  </S:Body>
</S:Envelope>
```

SAMLを使ったAssertionセキュリティ・トークンの例

```
<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <saml:Assertion
            MajorVersion="1"
            MinorVersion="0"
            AssertionID="SecurityToken-ef375268"
            Issuer="elliotw1"
            IssueInstant="2002-07-23T11:32:05.6228146-07:00"
           xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
        ...
      </saml:Assertion>
      ...
    </wsse:Security>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

③ Authentication Details

- Liberty Alliance Project：
  Libertyの認証オーソリティは、追加のAuthentication Context 情報を含めることができる。それらには、identification, technical Protection, Operational Protection (security audit, records archival), authentication Method(smartcard ..), Governing Agreement などに分類できる項目が含まれる。
- WS-Federation
  この項目は規定されていない。

④ Profile

- Liberty Alliance Project：
  ブラウザのアーティファクト、フォームベースの POST、WML(ECMAScript)ベースの LECプロファイル。(Bindings Profiles)
- WS-Federation
  バインディングプロファイルとして、パッシブ・リクエスタ・プロファイルと アクティブ・リクエスタ・プロファイルとがある。

⑤ Session

シングルログアウトについてはIDP主導、SP主導のいずれにもLAP、WSFともに存在する。

Liberty Alliance Project ではメッセージでなくHTML/SSLなどによるセッション維持によって安全性を保証している。

WS-Federation ではWS-SecurityConversationの中で規定されたセキュアな通信路を使用する。WS-SecurityConversation で規定されているのは、複数のメッセージのやり取りに使用可能なセッションを確立／維持するために当事者間でセキュリティ・コンテキストやセキュリティ・トークン(秘密鍵など)を交換または作成し、共有する方法である。これにしたがって署名されたメッセージをおのおの検証し、メッセージの確からしさと通信の安全性を保証する仕組みとなる。

3.5.5 応用例

LibertyのIdPが発行する属性アサーションを使用して、属性認証を含むシングルサインオンを行う例を以下に示す。この例では、出張のための宿泊施設や航空券の予約システムで、会社が旅行代理店や航空会社などのSPと提携し、会社のIdPを含むトラストサークルを形成している。社員は旅行代理店のポータルサイトにアクセスし、そこから航空券の予約のために航空会社のサイトにフォワードされる。SPとIdPの間であらかじめアイデンティティ連携がなされていれば、この間社員は会社のIdPに一度ログインするだけで、SP毎にログイン操作を行う必要はない(シングルサインオン)。また航空会社のサイトで航空券を予約する際には、IdPが発行した属性アサーションに設定されている役職情報により、課長以上ならばビジネスクラスが利用可能であるといった、属性認証を使用したアクセス制御をすることができる。

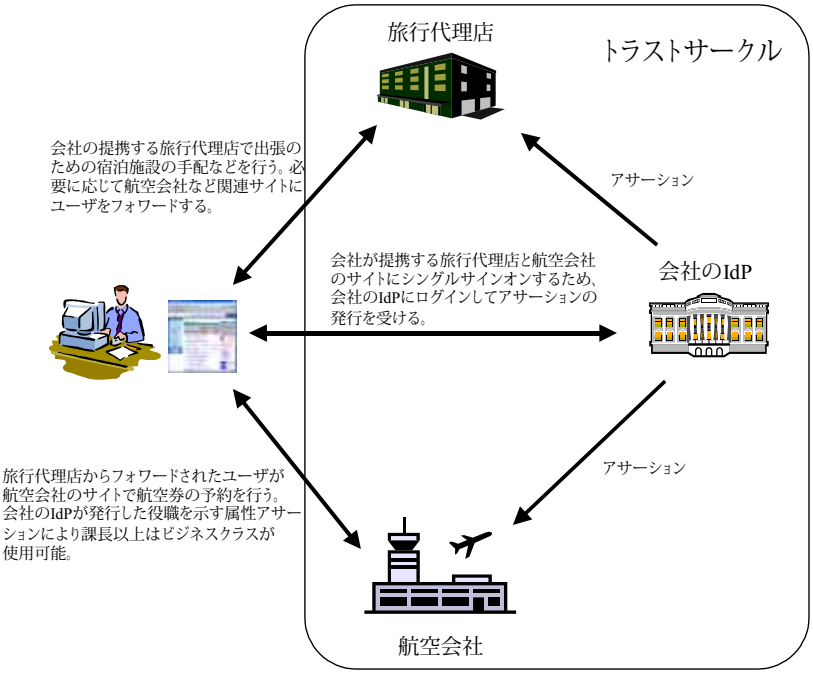


図 3-18 Liberty を使用したシングルサインオンと属性認証の例

3.6 権限付与

3.6.1 RBAC(NIST)

Role-Based Access Control (RBAC) は概念としては古くから存在し、UNIX のグループや、データベースマネージメントシステムの特権グルーピングのように実際に利用されてきた。アイデンティティベースのアクセスコントロールでは、ユーザを認証後そのユーザに与えられた権限を確認して、システムへのアクセスをコントロールする。一方 RBAC では、ユーザを認証後ユーザに割り当てられたロールを確認し、さらにそのロールに割り当てられた権限を確認してシステムへのアクセスをコントロールする。ロールは実世界でのユーザの役職に相当し、個々のロールには実世界での役職にふさわしい権限が割り当てられる。ロールを使用することによって、個々のユーザにそれぞれ権限を割り当てる手間を省いたり、実世界でのユーザの役職が変化した時に、ユーザの権限をひとつひとつ変更するのでなく、ユーザのロールを変更するだけで済ませたりすることができる。しかし、従来は RBAC というものの機能や特徴について、特段の標準化や合意は行われてはいなかった。

近年、米国の National Institute of Standards and Technology(NIST) が資金を提供して調査／研究を進めてきており、2001 年に RBAC のコンセンサスモデルを提案した [RBAC]。その後、2004 年の 2 月に、RBAC は ANSI INCITS 359-2004 として American National Standard Institute(ANSI) の International Committee for Information Technology Standards(INCITS) となった。

(1) 概要

RBAC の基本的なコンセプトには、以下の 6 つの要素が存在する。

表 3-11 RBAC の 6 要素

要素	説明
ユーザ(USERS)	人間。一般的には人間が操作するマシンや S/W、自立的に動作するエージェントなどを含む場合もある。一つのユーザには複数のロールを割り当てることができる。
ロール(ROLES)	ユーザに与えられた権限と責任をあらわす組織内でのユーザの役割。一つのロールを複数のユーザに割り当てることができる。また一つのロールには複数のパーミッションを割り当てることができる。
オブジェクト(OBS)	情報を保持したり、受け取ったりするエンティティ。RBAC を実装したシステムにおいては、ファイル／ディレクトリ／行／列／テーブルなどの情報コンテナであったり、プリンタやディスク容量、CPU 時間などの有限のシステムリソースであったりする。
オペレーション(OPS)	ユーザがオブジェクトに対して行う操作。
パーミッション(PRMS)	オブジェクトに対してオペレーションを行うことの許可。一つのパーミッションを複数のロールに割り当てることができる。
セッション(SESS)	ユーザとユーザに割り付けられたロールとの対応付けを行う。セッションは一つのユーザに割り付けられるが、ユーザは複数のセッションを持つことができる。

上記 6 つの要素の関係を図 3-19 に示す。

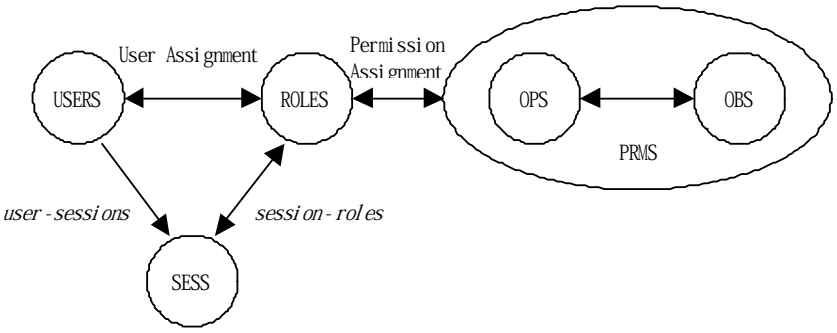


図 3-19 Core RBAC([RBAC]より引用)

ユーザはセッションを介してロールに割り付けられ、それによって、ロールに割り付けられているパーミッションがユーザに与えられることとなる。

(2) 階層化 RBAC

通常の RBAC に対して、ロールの階層化の概念を取り入れたのが階層化 RBAC である。階層化とは、ロール r11 と r1 がある時に、r1 の持つ全てのパーミッションが r11 にも割り当てられており、かつ、r11 に割り当てられている全てのユーザが r1 のユーザでもあることを言う。組織の例にたとえると、部長(r11)の持つ権限は課長(r1)の持つ権限より広いため、全ての部長は自身の部下である課長の代理を務めることができる、といった意味合いになる。

ロールの階層化はロールとロールの間に継承関係を定義する。例えば、r1 に割り当てられている全てのパーミッションが r11 に割り当てられ、かつ r11 の全てのユーザが r1 のユーザである場合、「r11 は r1 を継承している」という。ロール階層化のモデルを図に示す。

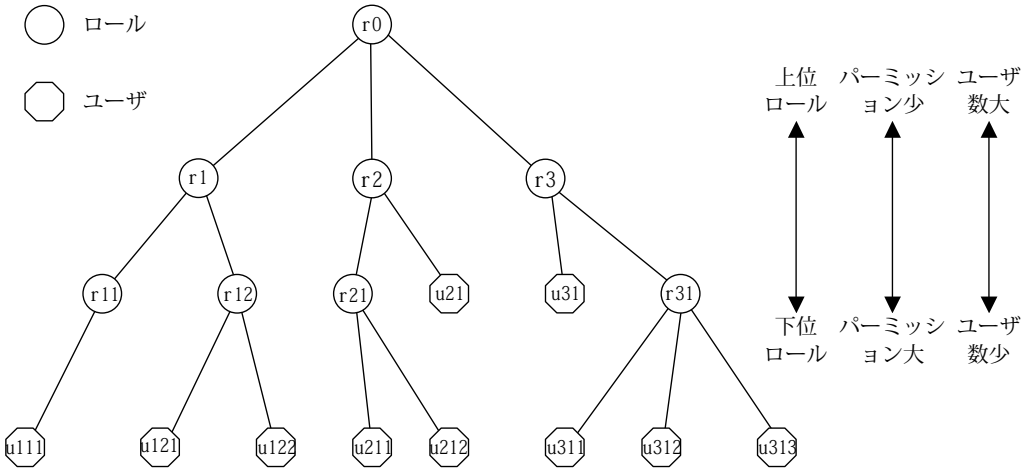


図 3-20 ロール階層化のモデル

この図では図の上に上位ロール、下に下位ロールが書かれている。上位ロールを継承して下位ロールが作られる。ロールの定義では上位ロールになるほどそのロールに割り付けられているパーミッションが少なく、上位ロールになるほどそのロールに割り付けられているユーザが多い。

RBAC では一般的ロール階層化と限定的ロール階層化の二つの概念を取り入れている。

一般的ロール階層化は階層全体の任意の一部分をロールの階層とみなせることを特徴とし、ロールに割り当てられたパーミッションとユーザの多重継承の概念を含む。

限定的ロール階層化とは、一般的ロール階層化に、階層の構造が単純なツリー構造になるような制限を加えたものである。その制限とは、「ロールは複数の下位ロールを持つことができるが、直接の上位ロールは単一でなくてはならない」というもので、多重継承ができないようにする制限である。

(3) 静的職責分離と動的職責分離

職責分離は RBAC の権限ポリシーに発生する矛盾を矯正する。RBAC の権限ポリシーに発生する矛盾は、例えば、ユーザが複数のロールに割り当てられている時に、ユーザの割り当てられたロールの中に、あるオブジェクトに対するオペレーションを許可するパーミッションを持ったロールと、それを許可しないパーミッションを持ったロールが存在するような場合に発生する。RBAC では静的と動的の 2 種類の職責分離方法がある。

表 3-12 職責分離の種類

職責分離の種類	説明
静的職責分離	ユーザをロールに割り付ける時に、割り付けられたロールのパーミッション間の整合性をチェックする。
動的職責分離	ユーザがセッション中に利用しているパーミッション間の整合性を実行時にチェックする。

静的職責分離ではユーザに割り付けられるロールがあらかじめ制限されるため、ユーザが利用できるパーミッションが制限されてしまうが、動的職責分離では実行時に現セッションで利用しているパーミッションがチェックされるため、セッションを張りなおすことによって、ユーザはそれまで使用していたのとは異なるパーミッションで作業を行うことができる。

3.6.2 XACML［XACML］

(1) 概要

ポリシー記述言語 XACML(eXtensible Access Control Markup Language)は、SAML のフレームワークの上で特にポリシー決定点(PDP：Policy Decision Point)に対する認可決定を判断させるためのポリシーおよびルール規定の仕様を定め、ポリシー決定点に対する認可要求やその応答のプロトコルを定めたものである。PDP はこのポリシー規定と認可要求の適合性を判断し、要求者の資源へのアクセスの許可、拒否を決定する。

XACML は XACL(XML Access Control Language)をベースとし、ポリシー記述の柔軟性と拡張性を高めたものとなっている。XACL はファイルの読み書き、生成、削除を制御する仕様を規定したシンプルな構造で、柔軟なポリシー記述やより多様な資源へのアクセスを制御する点では拡張性に欠けるものであったが、XACML は XACL の持っていた制約をできるだけ排除し、多様な資源へのアクセス制御を実現させる拡張性のある柔軟なポリシー仕様を規定できる。

なお、現在の仕様は SAML 環境以外にも適用できるように独立した仕様としてまとめられている。例えば XACML は SAML のほかに J2SE、CORBA などにも用いることができる。以下では XACML に関連する用語について解説する。

- PEP(Policy Enforcement Point)

アクセス要求者からの要求を受けて PDP に資源へのアクセスの可否を問い、PDP の示す許可、不許可の決定に基づき資源へのアクセスの制御を実施するところ。SAML の PEP(ポリシー実行点)の定義と同じもの。

- PDP(Policy Decision Point)

定められたポリシーに従って、PEP が示したアクセス要求が正しい権限を持つものかどうかを判断し、アクセス許可・不許可の決定を行うところ。SAML の PDP(ポリシー決定点)の定義と同じもの。

- PAP(Policy Administration Point)

PDP が参照するアクセス制御のルールを定義し、ポリシーやポリシー集合を生成するところ。

- PIP(Policy Information Point)

PEP の問合わせに対し、主体や資源や環境に関する属性値を提供するところ。SAML の属性オーソリティに相当するところ。

各ポイントの関係は、図のようになる。

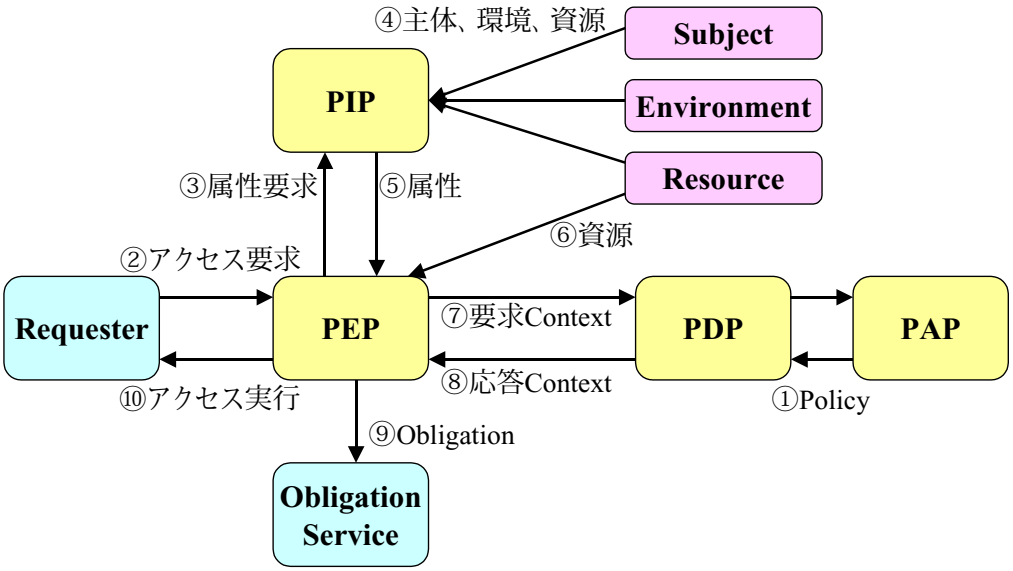


図 3-21 XACML 概要図

XACML では図 3-21 の「①Policy」「⑦要求 Context」「⑧応答 Context」について XML スキーマを定めている。

また、2004 年 12 月には「Core Specification: eXtensible Access Control Markup Language (XACML) Version 2.0」が同じく Committee Draft として公開されている。前バージョンとの違いは主に下記のプロファイルが定義され追加されたことにある。

- SAML 2.0 profile of XACML
- XML Digital Signature profile of XACML
- Privacy policy profile of XACML
- Hierarchical Resource profile of XACML
- Multiple Resource profile of XACML
- Core and Hierarchical Role Based Access Control (RBAC) profile of XACML, Version 2.0

(2) 要求 Context と応答 Context

XACML 要求 Context の<Request>要素は次に示す構文で構成される。

① 要求 Context

```
<Request xmlns="urn:oasis:names:tc:xacml:1.0:context"
  <Subject>
    <Attribute AttributeID= …>
        <AttributeValue> … </AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <ResourceContent> … </ResourceContent>
    <Attribute AttributeID= …>
        <AttributeValue> … </AttributeValue>
    </Attribute>
  </Resource>
  <Acction>
    <Attribute AttributeID= …>
        <AttributeValue>Read</AttributeValue>
    </Attribute>
  </Action>
  <Environment> … </Environment>
</Requst>
```

要求 Context は主体<Subject>がどのような資格や属性を持っているか、アクセスしたい資源<Resource>は何なのか、資源に対してどんな動作<Action>を要求するのかを、それぞれの属性<Attribute>要素に記述する。オプションとして主体、資源、動作以外の要求条件をオプションとしての環境<Environment>に与えることもできる。

上記に示す要求 Context に対して PDP は要求 Context の主体、資源、動作についてのそれぞれの属性をポリシーの対象<Target>のそれぞれ主体、資源、動作の規則と比較評価し、要求とポリシーがマッチすれば、次示すような応答 Context を返す。

② 応答 Context

```
<Response xmlns="urn:oasis:names:tc:xacml:1.0:context"
  <Result ResourceID= >
    <Decision>Permit</Dicision>
    <Obligations> … </Obligations>
  </Result>
</Response>
```

応答 Context は資源に対してのアクセスの認可決定<Decision>要素で Permit または Deny を指定する。PDP が要求に対するポリシーを評価する際、適用できるルールがなかった場合の決定<Decision>は NotApplicable となり、評価でエラーが起こった場合の決定<Decision>は Indeterminate となる。

もし評価するポリシーに責務<Obligations>が記載されていれば、応答 Context に責務<Obligations>をそのまま含める。責務<Obligations>を指定された PEP は責務を実行しなければならない。

(3) ポリシー記述言語

ポリシー記述言語はアクセス規則をいくつかの XML のルール<Rule>要素に記述する。ルール<Rule>には対象<Target>(主体、資源、動作)に対する規則を定める。ルールには条件<Condition>を加えることもでき、ポリシー<Policy>は複数のルール<Rule>を結合したものである。ポリシーにはオプションとして<Obligation>を指定できる。ポリシーが複数あった場合、このポリシーを結合してポリシー集合<PolicySet>を作る。ポリシー集合<PolicySet>は対象<Target>と<Policy>およびオプションとしての責務<Obligation>を含めてよい。

このように複数のルールを結合してポリシーを作り、ポリシーを結合してポリシー集合を作るのは PAP(Policy Administration Poinｔ)が行う。作られたポリシーは PDP が検索できるようにリポジトリなどに置かれる。

- Rule

ルール<Rule>は適用する対象<Target>の主体<Subjects>、資源<Resources>、動作<Actions>に対する規則からなり、オプションとして条件<Conditions>とする規則を設定することができる。ルールにはルール作成者が示す方針に従って、アクセス要求 Context の主体<Subjects>、資源<Resources>、動作<Actions>の属性が、このルールにそれぞれ適合した場合に決定する値(許可：Permit、または不許可：Deny)を、ルールの属性(Effect)として設定しておく。

- Target

Target では、アクセス制御の対象である主体<Subjects>、資源<Resources>、動作<Actions>の 3 つの要素に対する規則を定める。ルールはだれ<Subjects>に対して、どのような資源<Resources>を、どのように動作<Actions>させるかを記述する。PDP はこのルールに沿って要求 Context に示された主体、資源、動作の属性と、以下に示すルールの主体、資源、動作を比較する。評価結果は True または False となる。

➢ 主体<Subjects>

<Subject>は複数指定が可能であり、要求 Context で示される主体の属性が、ルールで規定する主体の条件とマッチするかを評価できるように、それぞれの主体の資格など属性のルールを記述する。要求とルールとの比較を行うために用いる比較関数の指定も行わなければならない。PDP は主体の資格が manager として要求されれば、ルールで決めた対象資格の文字列を比較して一致を検査する。またルールで 20 歳以上など主体を限定する場合、要求 Context の主体の年齢と比較を行うための関数を指定する。どのような主体でも許すのであればすべての主体を示す<AnySubject>要素を指定する。

➢ 資源<Resources>

<Resource>は複数指定が可能であり、要求 Context が示す<Resource>属性に対して、ルールを適用する資源を限定し、アクセスの対象とする資源を指定する。<Subject>と同様に要求とルールを比較するために比較演算の関数が用意されている。どのような資源でも許すのであればすべての資源を示す<AnyResource>要素を指定する。

➢ 動作<Actions>

<Action>は複数指定が可能であり、要求 Context が示す<Action>属性に対して、ルールを適用する資源へのアクセスに対する動作を指定する。SAML(Security Assertion Markup Language) で定めている Read、Write、Delete などの動作以外も定義することができる。また、要求とルールを比較するための比較演算の関数が用意されている。どのような資源でも許すのであればすべての資源を示す<AnyAction>要素を指定する。

● Condition

オプションとして指定できる条件<Condition>は、対象<Target>に対するルール以外に適用する規則である。例えば「月曜～金曜のみアクセスを許可する」というルールが与えられた場合、このルールは<Target>に対するルールではなく、条件<Condition>として指定する。この条件を評価したとき<Condition>の値は True か False を取る。

PDP は要求 Context の主体、資源、動作の属性と、ルールの<Target>の各主体、資源、動作を比較評価し、かつ<Condition>があれば<Condition>も評価して認可決定を下す。この場合、要求 Context とルールの<Target>の主体、資源、動作がすべてマッチし、<Condition>が True の場合のみルールの値は、結果属性 Effect で指定された Permit または Deny となる。<Target>がマッチしても<Condition>が False の場合は<Rule>の値は NotApplicable となる。

● Policy

ポリシー<Policy>は複数のルール<Rule>をまとめたものである。PAP は複数のルールを 1 つのポリシーに結合する。ポリシー<Policy>の子要素には対象<Target>、複数の<Rule>、責務<Obligations>があり、ポリシーID と PDP がこのポリシーを評価するときに用いるルール

結合アルゴリズムID(RuleCombiningAlgID)を<Policy>の属性に指定する。

対象<Target>はこのポリシーの対象で、主体、資源、動作のルールを規定する。ポリシー<Policy>に複数のルール<Rule>がある場合、PDPはルール結合アルゴリズムIDで示された以下に示すルール結合アルゴリズムで評価し、ポリシーの評価(Permit、Deny、NotApplicable、Indeterminate)を決定する。

もしこのポリシーに責務<Obligations>が規定された場合は、PDPはこの債務をPEP(Policy Enforcement Poiint)に示して責務の実行をPEPに課すことになる。PEPはPDPの結果がPermitであっても責務<Obligations>も同時に実施しなければならない。PEPは責務<Obligations>が実施できなければ資源へのアクセスを拒否しなければならない。

・ルール結合アルゴリズム

ポリシー<Policy>にはポリシー属性として、PDPがルールを評価し認可決定するために用いるルール結合アルゴリズムを指定しておく。ルール結合アルゴリズムとしては以下の3つのアルゴリズムが規定されている。

- Deny-overrides

ポリシー内のすべてのルールについて、どれか1つのルールのEffectが“Deny”であれば結果は“Deny”とする。すべてのルールが“Permit”、あるいは幾つかのルールが“Permit”で残りのルールすべてが“NotApplicable”の場合のみ結果を“Permit”とする。

- Permit-overrides

ポリシー内のすべてのルールについて、どれか1つのルールのEffectがが“Permit”であれば結果は“Permit”とする。すべてのルールが“Deny”、あるいは幾つかのルールが“Deny”で残りのルールすべてが“NotApplicable”の場合のみ結果を“Deny”とする。

- First-applicable

ポリシー内のすべてのルールについて順番に評価して、対象<Target>がマッチした場合、以後の<Rule>の評価を中断し<Target>の評価結果をMatchとし、次にConditionを評価し、これがTrueなら結果はEffectで指定された“Permit”または“Deny”とする。

ルール結合アルゴリズムはルールを評価するとき、上記の“Permit”または“Deny”の決定以外に、要求Contextが用意されたルールにすべてマッチしなければ評価結果は“NotApplicable”を返し、評価の過程で構文エラーとなった場合“Indeterminate”を返す。

- PolicySet

ポリシー集合<PolicySet>は複数のポリシー<Policy>をまとめたものである。もし複数の

ポリシーが登録されるなら PAP のポリシーは複数のポリシーを 1 つのポリシー集合に結合する。ポリシー集合<PolicySet>の子要素には対象<Target>、複数の<Policy>、<PolicySet>、責務<Obligation>があり、<PolicySet>の属性にポリシー集合 ID(PolicySetID) とポリシー結合アルゴリズム ID(PolicyCombiningAlgID) を指定する。ポリシー結合アルゴリズム ID は、このポリシー集合を評価する PDP が適用するポリシー結合アルゴリズムで、ルール結合アルゴリズムと同様な 3 つのポリシー結合アルゴリズムが規定されている。

対象はこのポリシー集合の対象で、主体、資源、動作を規定する。もしこのポリシー集合に責務<Obligation>が規定された場合は、PDP がこの責務を PEP に課すことになる。

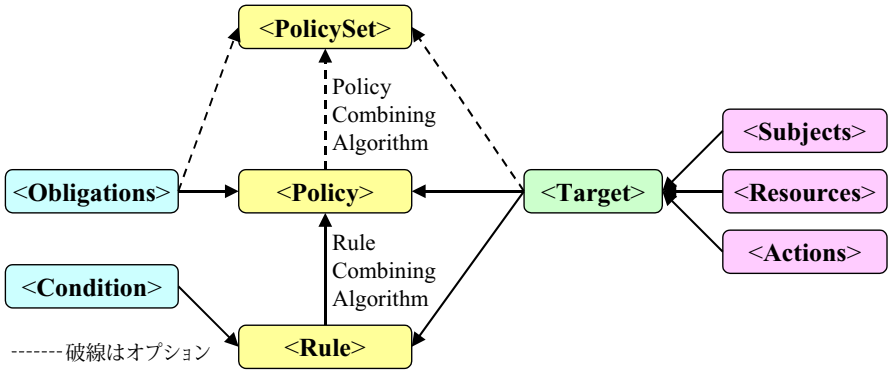


図 3-22 ポリシー記述言語

(4) RBAC プロファイル［RBAC-PROF］

XACML のプロファイルは多数存在するが、ここでは属性という観点から RBAC プロファイルについての例を示す。具体的にはマネージャと従業員という 2 つの Role を想定し、それぞれについて Permission と Role を定義している。

① Permission <PolicySet> for the manager role

以下に、マネージャ Role の Permission を定義したプロファイル例を示す。

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:1.0:policy"
 PolicySetId="PPS:manager:role"
 PolicyCombiningAlgId="&policy-combine;permit-overrides">
  <Target>
    <Subjects><AnySubject/></Subjects>
    <Resources><AnyResource/></Resources>
    <Actions><AnyAction/></Actions>
  </Target>

  <!-- Permissions specifically for the manager role -->
  <Policy PolicyId="Permissions:specifically:for:the:manager:role"
   RuleCombiningAlgId="&rule-combine;permit-overrides">
```

```
  <Target>
    <Subjects><AnySubject/></Subjects>
    <Resources><AnyResource/></Resources>
    <Actions><AnyAction/></Actions>
  </Target>

  <!-- Permission to sign a purchase order -->
  <Rule RuleId="Permission:to:sign:a:purchase:order"
   Effect="Permit">

    <Target>
      <Subjects><AnySubject/></Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="&function;string-match">
            <AttributeValue
             DataType="&xml;string">purchase order</AttributeValue>
              <ResourceAttributeDesignator
               AttributeId="&resource;resource-id"
               DataType="&xml;string"/>
          </ResourceMatch>
        </Resource>
      </Resources>

      <Actions>
        <Action>
          <ActionMatch MatchId="&function;string-match">
            <AttributeValue
             DataType="&xml;string">sign</AttributeValue>
            <ActionAttributeDesignator
             AttributeId="&action;action-id"
             DataType="&xml;string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
</Policy>
```

```
    <!-- Include permissions associated with employee role -->
    <PolicySetIdReference>PPS:employee:role</PolicySetIdReference>
  </PolicySet>
```

② Permission <PolicySet> for employee role

以下に、従業員 Role の Permission を定義したプロファイル例を示す。

```
  <PolicySet xmlns="urn:oasis:names:tc:xacml:1.0:policy"
   PolicySetId="PPS:employee:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
    <Target>
      <Subjects><AnySubject/></Subjects>
      <Resources><AnyResource/></Resources>
      <Actions><AnyAction/></Actions>
    </Target>

  <!-- Permissions specifically for the employee role -->
    <Policy PolicyId="Permissions:specifically:for:the:employee:role"
     RuleCombiningAlgId="&rule-combine;permit-overrides">
      <Target>
        <Subjects><AnySubject/></Subjects>
        <Resources><AnyResource/></Resources>
        <Actions><AnyAction/></Actions>
      </Target>

      <!-- Permission to create a purchase order -->
      <Rule RuleId="Permission:to:create:a:purchase:order"
       Effect="Permit">
        <Target>
          <Subjects><AnySubject/></Subjects>
          <Resources>
            <Resource>
              <ResourceMatch MatchId="&function;string-match">
                <AttributeValue
                 DataType="&xml;string">purchase order</AttributeValue>
                <ResourceAttributeDesignator
                 AttributeId="&resource;resource-id"
                 DataType="&xml;string"/>
```

```
            </ResourceMatch>
          </Resource>
        </Resources>

        <Actions>
          <Action>
            <ActionMatch MatchId="&function;string-match">
              <AttributeValue
               DataType="&xml;string">create</AttributeValue>
              <ActionAttributeDesignator
               AttributeId="&action;action-id"
               DataType="&xml;string"/>
            </ActionMatch>
          </Action>
        </Actions>
      </Target>
    </Rule>
  </Policy>
</PolicySet>
```

③ Role <PolicySet> for the manager role

以下に、マネージャ Role の Role を定義したプロファイル例を示す。

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:1.0:policy"
 PolicySetId="RPS:manager:role"
 PolicyCombiningAlgId="&policy-combine;permit-overrides">
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="&function;string-equal">
          <AttributeValue
           DataType="&xml;string">manager</AttributeValue>
          <SubjectAttributeDesignator
           AttributeId="urn:someapp:attributes:role"
           DataType="&xml;string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources><AnyResource/></Resources>
```

```
    <Actions><AnyAction/></Actions>
  </Target>

  <!-- Use permissions associated with the manager role -->
  <PolicySetIdReference>PPS:manager:role</PolicySetIdReference>
</PolicySet>
```

④ Role <PolicySet> for employee role

以下に、従業員 Role の Role を定義したプロファイル例を示す。

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:1.0:policy"
 PolicySetId="RPS:employee:role"
 PolicyCombiningAlgId="&policy-combine;permit-overrides">
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="&function;string-equal">
          <AttributeValue
           DataType="&xml;string">employee</AttributeValue>
          <SubjectAttributeDesignator
           AttributeId="urn:someapp:attributes:role"
           DataType="&xml;string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources><AnyResource/></Resources>
    <Actions><AnyAction/></Actions>
  </Target>

  <!-- Use permissions associated with the employee role -->
  <PolicySetIdReference>PPS:employee:role</PolicySetIdReference>
</PolicySet>
```

3.7　参考文献

[RBAC-PROF] http://docs.oasis-open.org/xacml/cd-xacml-rbac-profile-01.pdf

[ISO17090] ISO/TS 17090-2 Health informatics – Public key infrastructure – Part 2: Certificate profile

[HPKIREPORT] 医療情報ネットワーク基盤検討会最終報告「今後の医療情報ネットワーク基盤のあり方について」http://www.mhlw.go.jp/shingi/2004/09/s0930-10a.html

[JCSICP1] AccreditedSign パブリックサービス 1 標準規程(V1.22)
http://www2.jcsinc.co.jp/repository2/A_SignCPS1.pdf

[JCSICP2] AccreditedSign パブリックサービス 2 標準規程(V2.22)
http://www2.jcsinc.co.jp/repository2/ASignCPS.pdf

[X509] ITU-T Recommendation X.509 Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks

[RFC3281] RFC 3281 An Internet Attribute Certificate Profile for Authorization
http://www.ietf.org/rfc/rfc3281.txt

[SAML] Assertions and Protocol for the OASIS Security Assertion Markup Language(SAML)V1.1
http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

[LIBERTYTUTORIAL] Liberty 仕様チュートリアル
http://www.projectliberty.org/jp/resources/LAP_DIDW_Oct_15_2003_jp.pdf

[LIBERTYARCHOVERVIEW] Liberty ID-FF アーキテクチャ概要　バージョン:1.2
http://www.projectliberty.org/jp/resources/LAP-ID-FF-archtecture-overview-v1.2-JP.pdf

[WSSROADMAP] Security in a Web Services World : A Proposed Architecture and Roadmap
http://www-128.ibm.com/developerworks/webservices/library/ws-secmap

[WSFLANG] Web Services Federation Language(WS-Federation)Version 1.0
http://www-106.ibm.com/developerworks/library/ws-fed/

[WSSECURITY] Web Services Security:SOAP Message Security 1.0(WS-Security 2004)
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[WSSUSERTOKEN] Web Services Security Username Token Profile 1.0
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

[WSSX509TOKEN] Web Services Security X.509 Certificate Token Profile
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf

[WSSSAMLTOKEN] Web Services Security: SAML Token Profile

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf
[WSSRELTOKEN] Web Services Security Rights Expression Language(REL)Token Profile
http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf
[WSSKERBEROSTOKEN] Web Services Security Kerberos Token Profile 1.0
http://www.oasis-open.org/committees/download.php/8266/oasis-xxxxxx-wss-kerberos-token-profile-1%200.pdf
[WSFEDLIBERTYOV] LIBERTY ALLIANCE PROJECT ホワイトペーパー Liberty Alliance と WS-Federation：比較概要 2003年10月14日
http://www.projectliberty.org/jp/resources/wsfed-liberty-overview-10-13-03-JP.pdf
[WSTRUST] Web Services Trust Language(WS-Trust)Version 1.1
ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf
[RBAC] Proposed NIST Standard for Role-Based Access Control
http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf
[XACML] 第5回PKIとPMIを融合させる次世代言語XACML
http://www.atmarkit.co.jp/fsecurity/rensai/webserv05/webserv01.html
第6回XACMLのアクセス制御ルールとその仕様
http://www.atmarkit.co.jp/fsecurity/rensai/webserv06/webserv01.html

4. 海外動向

4.1 e-Authentication

米国では、電子政府の横断的イニシアチブである e-Authentication イニシアチブと、e-Authentication イニシアチブの成果を参照しながら官民・民民の連携のための電子認証スキームを確立しようとする EAP(Electronic Authentication Partnership) という動きがある。

本節では、米国の電子認証フレームワークの取り組みとして、e-Authentication イニシアチブと EAP の動向を紹介する。

4.1.1 e-Authentication イニシアチブとは

(1) 設立背景と概要

e-Authentication イニシアチブは、2001 年の大統領マネジメントイニシアチブとして発表された電子政府(E-Gov) イニシアチブのひとつである[2]。電子政府の FEA(連邦政府エンタープライズアーキテクチャ) に対応した認証フレームワークであり、政府機関を横断したシングルサインオンの実現を目指している。複数の認証プロバイダが連携するモデルであり、SAML や Liberty Alliance などの技術を統合している。

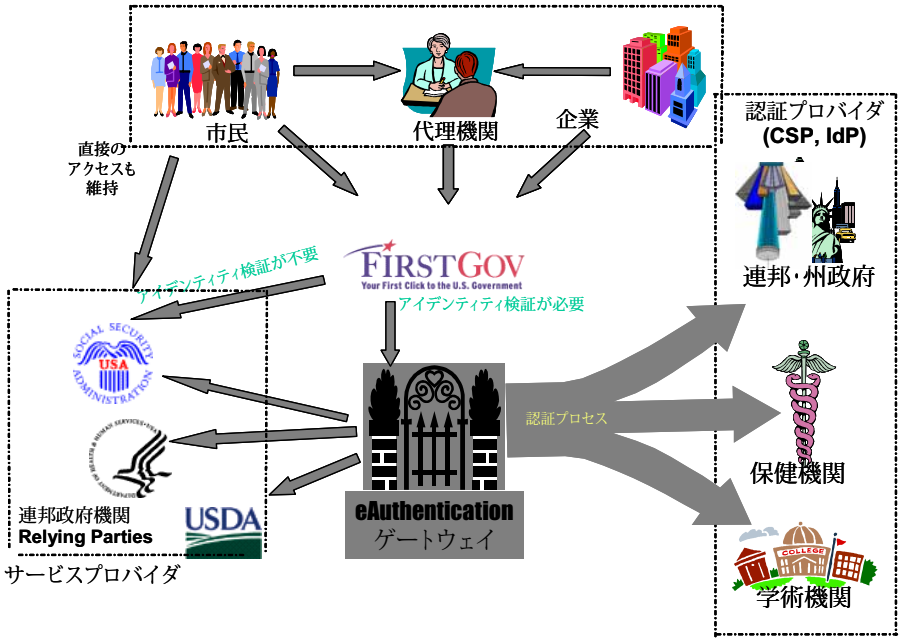


図 4-1 e-Authentication ゲートウェイ[3]

図 4-1 中央の FirstGov［FirstGov］とは連邦政府の行政ポータルサイトであり、

[2] E-Gov http://www.whitehouse.gov/omb/egov/ 横断的イニシアチブである e-Authentication の他、G2C、G2G、G2B、IEE（Internal Efficiency & Effectiveness）のサービスがある。

[3] [出典] “ e-Authentication Initiative: Where we are, How we arrived”（2002/11/21）, http://www.cio.gov/eauthentication/presentations/forum_timchak.ppt ［スライド 9]

e-Authentication イニシアチブでは連邦政府における認証作業をこの行政ポータルサイト経由で一元的に行う、シングルサインオンの環境を実現しようとしている。図の右側に認証プロバイダ、図の左側に連邦政府機関による認証を信頼するサービスプロバイダが存在し、設置されたe-Authenticationゲートウェイを通してPINやPKI などによる認証が行われるとともに、リスクと保証レベルが4段階で定義される。ユーザ(国民、代理機関、企業)は、この4段階の保証レベルに応じてアクセス許可される。また、ユーザは直接サイトへアクセスする手段も維持される、という構想である。

(2) e-Authenticationの現状

e-Authenticationイニシアチブの成果として各種文書やツールが公開されているので、これらの関係を図 4-2 に示し、概要を紹介する。

Policy:
> E-Authentication Guidance for Federal Agencies
> NIST Special Publication 800-63

Tools:
> Trusted Credential Service Provider List
> E-RA Tool (Risk Assessment)
> E-Authentication Technical Architecture
> E-Authentication Credential Assessment Suite
> Approved E-A Technology Provider List

References:
> E-Authentication Cookbook
> E-Authentication Handbook for Federal Agencies
> E-Authentication Handbook for CSPs
> E-Authentication Interoperability Lab Concept of OPs

[出典] http://www.cio.gov/eauthentication/

図 4-2 e-Authenticationイニシアチブの公開文書・ツール

OMB ガイダンス：正式名称は、「連邦政府機関に対する e-Authentication ガイダンス(e-Authentication Guidance for Federal Agencies)」[OMB M-04-04] であり、行政予算管理局(OMB：Office of Management and Budget)の Executive Office of the President の Director である Joshua B. Bolten 氏から全政府部門機関のトップ宛のメモという形式で 2003 年 12 月に発表された。2003 年 7 月に 1 ヶ月間コメントを求めており、寄せられたコメントとその回答のサマリも添付されている。この文書では、4 つの保証レベルを定義し、システムの認証エラーに対するリスク・潜在的影響と要求される保証レベルの関係について記述している。また、認証プロセスの実装についても触れている。

NIST ガイダンス：正式名称は、「電子認証ガイダンス：NIST 勧告(Electronic Authentication

Guideline: Recommendations of the National Institute of Standards and Technology) [NIST sp800-63]であり、2004年7月にVersion1.0が発表され、2004年9月に改定されてVersion1.01となった。ドラフト版は2003年から公開されており、2004年1月に発表されたドラフトに対しては、約2ヶ月間パブリックコメントを募集した。内容は、OMBガイダンスを技術的に補足することであり、4つの保証レベルごとに、アイデンティティ証明や登録、トークン、認証プロトコル、アサーションといった技術的な必要条件を記述している。

OMBガイダンスとNIST技術ガイダンスの関係を図 4-3に示す。

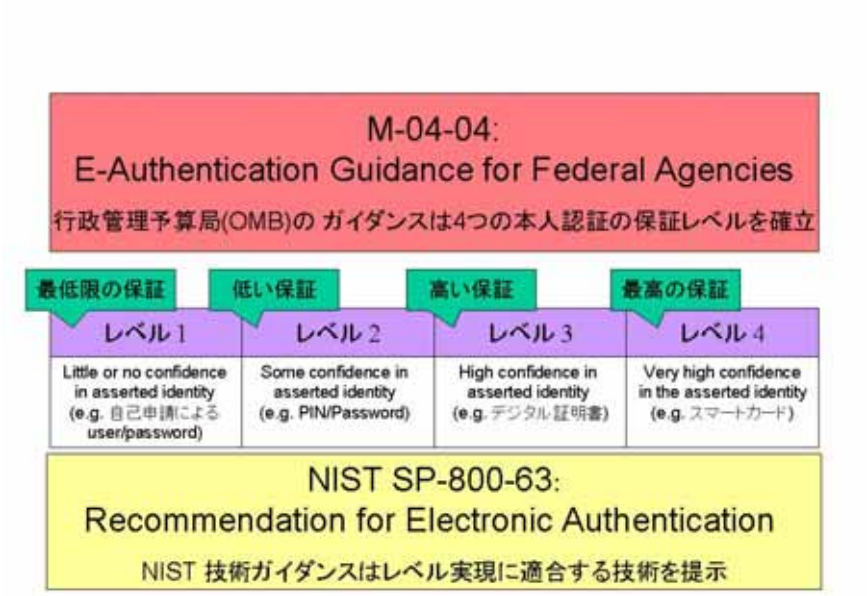


図 4-3 OMBガイダンスとNISTガイダンスの関係

Trusted Credential Service Providers(TCSP)List: e-Authenticationイニシアチブが承認した、クレデンシャルサービスプロバイダ[4]のリストである。4つの保証レベル毎にサービスプロバイダ名・クレデンシャルタイプ・認定日を一覧することが可能である。尚、現在のリストのクレデンシャルタイプはPKIとパスワードとPINである。

[4] e-AuthenticationのCredential Service Provider(CSP)は、Identity Provider(IdP)と同義。

表 4-1 Trusted Credential Service Provider List(2004/12/22 現在)[5]

サービスプロバイダ名	タイプ	認定日
レベル4		
Department of the Treasury (財務省)	PKI	2003/11/03
Department of Defense (国防総省)	PKI	2003/11/03
Department of State (国務省)	PKI	2004/01/21
Federal Common Policy Framework 連邦政府共通ポリシフレームワーク	PKI	FICCのCPLリスト参照 http://www.cio.gov/ficc/cpl.htm
レベル3		
Department of the Treasury (財務省)	PKI	2003/11/03
NASA (航空宇宙局)	PKI	2003/11/03
USDA/National Finance Center (農務省金融センター)	PKI	2003/11/03
Department of Defense (国防総省)	PKI	2003/11/03
Department of Energy (エネルギー省)	PKI	2003/11/03
ACES/DST 電子サービス用アクセス証明書/デジタル・シグネチャー・トラスト	PKI	2003/11/03
State of Illinois イリノイ州	PKI	2003/11/03
ACES/ORC 電子サービス用アクセス証明書/ORC	PKI	2004/08/18
Federal Common Policy Framework 連邦政府共通ポリシフレームワーク	PKI	FICCのCPLリスト参照 http://www.cio.gov/ficc/cpl.htm
レベル2		
USDA(農務省) e-Authenticationサービス	パスワード	2003/12/22 *
ORC, Inc	パスワード	2004/03/10
OPM(人事局) Employee Express	PIN	2004/12/21
レベル1		
ORC, Inc	パスワード	2003/10/27 & 2004/03/10
USDA(農務省) e-Authenticationサービス	パスワード	2003/12/22 *
NSF(全米科学財団) FastLane	パスワード	2004/03/15

* e-Authentication 相互運用性テストはペンディング

FICC［FICC］のCPL リスト(正式タイトル”Shared Service Providers(SSP)Subcommittee Certified Providers List”)

・United States Department of Agriculture/National Finance Center(USDA/NFC)農務省金融センター

・VeriSign, Inc. ベリサイン社

・Betrusted U.S. Inc. 米ビートラステッド社

e-RA ツール：保証レベルを定義したことにより、政府機関はサービスを提供するシステムがどの保証レベルを必要とするのか見極めなければならない。OMB ガイダンスにも記述されてい

[5] http://www.cio.gov/eauthentication/TCSPlist.htm

る、リスクアセスメントの手順は、次の5ステップとなる。

① 電子政府システムのリスクアセスメントを行う
② 特定されるリスクを適切な保証レベルにマッピングする
③ e-authentication技術ガイダンスに基づいて技術を選択する
④ 構築したシステムが求められた保証レベルを達成しているかを認可する
⑤ 定期的に再査定を行う

この手順のステップ1とステップ2において使うツールが本e-RAツールである。e-RAとは、リスクに基づいた認証要件のアプローチElectronic Risk and Requirements Assessmentの略語である。e-RA ツールは、カーネギーメロン大学ソフトウェアエンジニアリング研究所(CMU/SEI)が開発し、e-Authenticationイニシアチブが政府のITシステムに対して認証リスクを評価する際に使用するものである。e-Authenticationイニシアチブを通じて、誰でも利用することができ、現在入手可能なバージョンは 1.4B である。すでに、政府機関の主要なシステムの評価が開始されており、2004年度中に評価を終了する予定になっている。

リスクアセスメントを行った結果、特定された保証レベルと同一基準のクレデンシャルプロバイダを使用するために、前述のTCSPリストを参照することになる。

技術アーキテクチャ：「認証サービスコンポーネントの技術アプローチ」「e-Authenticationに採用されたスキームとしてのSAMLアーティファクト・プロファイル」「SAMLアーティファクト・プロファイル用e-Authentication インタフェース仕様書」の3文書で構成されている。「認証サービスコンポーネントの技術アプローチ」は、複数のプロトコルとスキームをサポートすることを可能にするアーキテクチャ・フレームワークを説明している。SAML は既にe-Authenticationアーキテクチャ・フレームワークに採用されているので、e-Authenticationイニシアチブにおける、SAMLアーティファクト・プロファイル仕様やインタフェース仕様書が用意されている。

クレデンシャル評価フレームワーク：パスワード、PIN、PKI といったクレデンシャルを評価するためのフレームワーク(Credential Assessment Framework, CAF)と評価ガイダンス、評価プロファイルの文書群である。前述の TCSP リストを作成する際の評価プロセスで具体的に利用されるクライテリアは、このCAF中の評価プロファイルに述べられている。

相互運用性承認製品リスト(技術プロバイダ認定リスト)：e-Authenticationイニシアチブが相互運用ラボを作り、連邦政府の環境において基本的な相互運用性を検証し、承認された製品を使用して各連邦政府機関がe-Authenticationを実装してよいとしている。SAML 1.0 アーティファクト・プロファイルを使って検証されており、2004年12月6日付で、下記の9製品がリストアップされている。

表 4-2 相互運用性承認製品リスト

提供元	製品名
Entegrity	AssureAccess v3.0.0.4
Entrust	GetAccess v7.0 SP 2 Patch 3
Hewlett-Packard	Select Access v5.2
IBM	Tivoli Federated Identity Manager v5.1.1
Netegrity	Site Minder 6.0.1.04
Oblix	ShareID 2.0
RSA Security	Federated Identity Manager v2.5LA
Sun Microsystems	JES Identity Server 2Q2004
Trustgenix	IdentityBridge 2.1

e-Authentication クックブック［Cookbook］： e-Authentication イニシアチブに参加したい、連邦政府機関とクレデンシャルサービスプロバイダ(CSP)、ベンダ、その他の利害関係者を手助けするものである。このクックブックは、多くの技術的および非技術的な手続き、および、ソフトウェアとハードウェアの構造の詳細な記述から構成される。このクックブックの文脈では、e-Authentication イニシアチブとの統合に責任を負う人々に、時間節約の手段となるレシピが提供される。これらのレシピは、連合した ID 管理に対して要求される、新しく複雑なアーキテクチャを実装することに起因する、長期的な質問に答えることを目指している。レシピは、4 つのカテゴリ(プロセス、統合、実装、製品とサービス)で構成されている。

政府機関向け e-Authentication ハンドブック［GOVhandbook］：このハンドブックは、e-Authentication への参加を計画している、または既に参加している政府機関に一般的なガイドラインを提示する。このハンドブックは、政府機関に完全な展望やガイダンスを提供するために、e-Authentication 参加の全体的なライフサイクルの見方を提供する。

クレデンシャルサービスプロバイダ向け e-Authentication ハンドブック［CSPhandbook］：このハンドブックは、e-Authentication イニシアチブ向けのクレデンシャルサービスを提供することに興味を持っている民間産業組織や政府機関へ一般的なガイドラインを提示する。このハンドブックは、クレデンシャルサービスプロバイダ(CSP)に完全な展望やガイダンスを提供するために、e-Authentication 参加の全体的なライフサイクルの見方を提供する。

e-Authentication 相互運用性ラボ・運営構想：この文書は、e-Authentication 相互運用性ラボの運営構想を記述している。この相互運用性ラボは、製品とソリューションに対して、業界標準のサブセットである、e-Authentication のインタフェース仕様の適合性に関するテストを行う。さらに、ラボは、製品に対して、同じ適合性を主張する他の製品との相互運用性に関してテストを行う。

このような e-Authentication イニシアチブの成果に対し、2004 年 8 月に発表された Burton

Group の調査報告書［BurtonGroupReport］は、短期的及び長期的な提言を寄せている。PMO の責任者は、e-Authentication に対する内容について、調査結果を喜び「政府規模の ID 認証システム構築するにあたり、イニシアチブのアプローチは正しかった」と述べている［e-auth］。

4.1.2 EAP(Electronic Authentication Partnership) とは

EAP(Electronic Authentication Partnership)は、公的および民間の電子認証システムの相互運用を可能にすることをタスクとしている、多産業パートナーシップである。その背景には、電子認証の相互運用性が、産業界を横断的に、電子処理を行う安全でセキュアなシステムを構築運営していく上で必要不可欠な要素である、という考えがあるためである。

EAP の設立には、ワシントン DC にある戦略国際問題研究所(CSIS)とジョンズ・ホプキンズ大学が公的部門と民間部門での電子認証相互利用の分析のための WG を召集したところから始まる。2003 年春に、その WG はそれぞれ報告書を作成し、その後、CSIS が官民協業の体制を作ろうと動き、EAP となった。EAP の設立発表は、2003 年 12 月、OMB の電子政府 IT 室の Karen Evans 氏によって行われた。当初は政府資金で活動してきたが、2004 年 10 月からは正式に承認された会則[6]に基づき参加メンバーの会費で運営されるようになる。

会員にはビジネスメンバーと非ビジネスメンバーの種別があるが、2004 年 11 月末現在の参加メンバーは、民間企業・政府機関を合わせて、136 団体である。その内 40 団体が、EAP の会員団体として 2004 年 10 月からの会費を支払い、投票権を得ることに同意して LOI(Letter of Intent：同意書)に署名している。

政府機関では、e-Authentication イニシアチブのプログラムマネージャオフィス(PMO)、GSA、国防総省、教育省、保健社会福祉省、環境保護庁環境情報局、社会保障局、米国雇用機会均等委員会、下院議会、郵政省などが参加しているが、LOI に署名したのは GSA のみである。

EAP には 4 つのワーキンググループ(ビジネス要件とプロセス WG、クレデンシャルサービス評価クライテリアと保証レベル WG、評価・診断・コンプライアンス WG、ガバナンス WG)があり、クレデンシャルサービス評価クライテリアと保証レベル WG 内には 3 つのサブワーキンググループ(クレデンシャルサービス評価クライテリア SWG、保証レベル SWG、相互運用性 SWG)が設けられている。WG の活動は非常に活発であり、EAP 全体会合も月 1 回のペースで実施されている。成果物が出来上がりつつあり、EAP のフレームワークのドラフトを 2004 年中に完成させようとしている。

EAP の考える電子認証の相互運用性とは：EAP の最終目標は、様々な e-authentication シス

---

[6] 2004 年 9 月 2 日に採択された BYLAWS of Electronic Authentication Partnership (http://www.eapartnership.org/docs/CompletedEAPBylaws_v1.0.doc)より、正式名称は、Electronic Authentication Partnership, Inc. デラウェア州の法律の下組織された、非株式・非営利法人

テムによって発行されたデジタルなクレデンシャルを信頼する簡単な手段を民間組織に提供することである。下記に基礎を置いている。

- 政府・民間部門・公的利益団体の全レベルから利害関係者間のデジタル認証の統合に対する自発的なパートナーシップを作る。
- 個々の組織が信頼したいと願う認証プロセスを持つ組織とそれぞれ二者合意を結ぶ必要性を排除し、代わりに、EAP のルールに従って運営する団体は、全参加者間の多者間信頼に帰着する規則に承諾する。
- 信頼性・相互運用性と簡単な評価と他の団体によって発行された様々なタイプのクレデンシャルの受理を促進しうるクレデンシャル・クレデンシャルプロバイダ・クレデンシャル処理者に対する共通のポリシーと経験を確立・維持する
- クレデンシャルの評価プロセスを開発し、標準的なアプローチと ID 管理の最低限の要件を確立する。
- 運営ルールと関連するプロセスに対する既存のクレデンシャルメカニズムに基づき、それを補足する。
- 他の国のIDシステムと協調的に動かす。

このようなEAPの考える相互運用性を実現する手段は次のようになる。

- 別の階層的な保証レベルのクレデンシャルと認証システムのためのルールを起草する。これらのルールは、各保証レベル毎にクレデンシャルを評価するクライテリアの標準セットを提供する。
- クライテリアの標準セットに対する、クレデンシャルとシステムを評価し、検証者へその評価を伝達する手段を開発する。
- 第三者機関のクレデンシャルの使用を認める検証者への「契約規則」を起草する。これらの規則は二者間合意に取って代わるものになる。
- クレデンシャルを有効にする運用規則を作成する。どのようにクレデンシャルの有効性を処理するかを定義する。

従って、EAPに参加する組織は、次のようなメリットを得ることができる。

- e-authentication相互運用性において、主要な組織となれる。
- (個々の会社、業界グループあるいは政府系機関としても)自身の意見を貢献し、問題解決の中で影響力をもてる。
- 他の産業における、e-authenticationの動きを学べる。
- e-authenticationに対する二者合意の交渉と実施の必要性を省ける。
- 他の認証システムによって実行された電子認証を受理できることによる、時間と資源を節約する機会を検証できる。

- 技術や経験の相互運用性を促進するために進行中のフォーラムに参加できる

そして、このようなEAPの活動に対して、Burton Groupの調査報告書［BurtonGroupReport］は、e-Authenticationイニシアチブと同様EAPに対しても、短期的及び長期的な提言を寄せている。

最後に、e-AuthenticationイニシアチブとEAPを取り巻く、組織関係を図 4-4に示す。

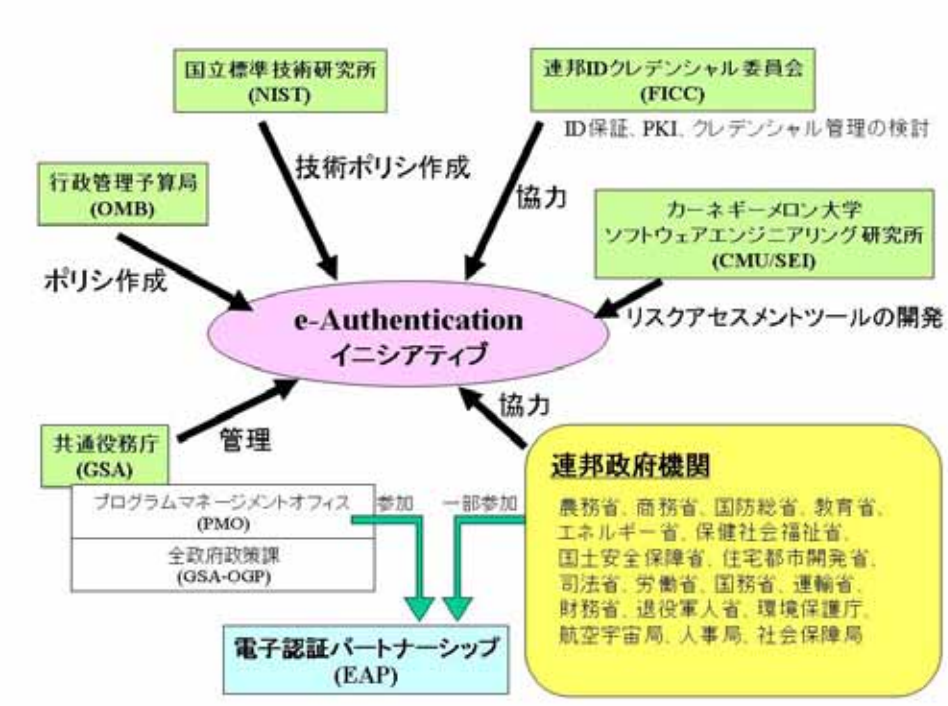


図 4-4 組織関係図[7]

[7] [出展] http://www.cio.gov/eauthentication/

4.2　海外標準化団体とその役割

本節では、属性認証の標準策定に関わる、各種団体やフォーラムを紹介する。

4.2.1　CEN/ISSS(http://www.cenorm.be/isss/)

CEN/ISSS(European Committee for Standardization / Information Society Standardization System)は、欧州における情報化社会の発展を目的とし、市場関係者に包括的で統一的な一連の標準化サービスと製品を提供する。

CEN/ISSS ワークショップは、CWA(CEN Workshop Agreement)として発表される項目に対して、欧州レベルで合意を形成すること目的としており、電子署名については CEN/ISSS WS/E-sign で作業されている。

4.2.2　EESSI(http://www.ict.etsi.fr/EESSI_home.htm)

EESSI(European Electronic Signature Standardization Initiative)は、欧州の電子署名法(EU Directive)を受けて電子署名に関する標準やガイドラインを整備するためのイニシアチブである。EESSI は CEN/ISSS(European Committee for Standardization / Information Society Standardization System)および ETSI(European Telecommunications Standards Institute)を含め多くの欧州標準を策定してきた。これらの標準やガイドラインは EU Directive の 4 つの Annex に沿って、すなわち証明書のプロファイル(QC)、認証局のセキュリティ要件、署名装置(Electronic Signature Devices)、署名検証(長期的な署名検証を含む)の要件に対して標準やガイドラインを策定してきた。これらの標準はまた IETF や W3C にも提案され国際的な標準としての位置を占めるようになってきている。

4.2.3　ETSI(http://www.etsi.org/home.htm)

ETSI(European Telecommunications Standards Institute)は、欧州の通信技術の標準を定める機関である。欧州の電子署名法(EU Directive)を推進するに当たって電子署名技術や運用技術を推進するために作られた EESSI(4.2.2 参照)のもとで電子署名に関連する各種標準を策定している。

作業は、ETSI 内のテクニカルボディである ETSI ESI(Technical Committee-Electronic Signatures and Infrastructures)で行っており、電子署名フォーマットは ETSI 標準(ES)の ES 201733 として 2000 年に発表され、ES に基づきインフォーマルな RFC として、IETF に提案している。

4.2.4 IDA

IDA(Interchange of Data between Administrations)は欧州委員会の戦略的なイニシアチブであり、情報通信技術の発展を通じて加盟国の行政府間の電子情報交換を促進することを目指している。その目的は、欧州連合の意志決定を容易にし、域内市場の取引をスムーズにして、政策の実施を促進することにある。

IDA は、汎欧州通信ネットワークの確立を目指しており、基盤構築・共通のフォーマットの確立・ビジネスプロセスをベースとした新しい ICT(Information and Communications Technologies)の統合に取り組み、最近ではネットワークのサービス、ツール、相互運用性の改良に努力している。

4.2.5 IETF(http://www.ietf.org/)

IETF(Internet Engineering Task Force)はインターネットアーキテクチャとインターネットのスムーズなオペレーションの革新に関心のあるネットワーク設計者、オペレータ、ベンダ、研究者から構成されるオープンな国際的な団体である。いくつかの分野(経路、転送、セキュリティなど)で多くの WG が技術標準の策定作業を行っている。まとめられた標準案は Internet Engineering Steering Group(IESG).に上げられ RFC(Request for Comments)としての標準が作られる。PKI 分野の RFC は、PKIX WG で主要な RFC が数多くまとめられている。関連する WG には暗号メールと暗号フォーマットを扱う S/MIME、トランスポートレイヤのセキュリティを扱う TLS、IP レイヤのセキュリティの IPsec、W3C とジョイントで XML 署名を扱う XMLDSIG などの WG がある。

本ハンドブックで関係する RFC を示す。

RFC	タイトル
RFC 2459 RFC 3280	インターネット X.509 PKI - 証明書と CRL のプロファイル (Internet X.509 Public Key Infrastructure Certificate and CRL Profile) ※RFC 2459 は、RFC 3280 により廃止
RFC 3039	インターネット X.509 公開鍵インフラストラクチャ QC プロファイル (Internet X.509 Public Key Infrastructure Qualified Certificates Profile) ※RFC 3739 により廃止
RFC 3739	インターネット X.509 PKI： 特定証明書プロファイル (Internet X.509 Public Key Infrastructure: Qualified Certificates Profile)
RFC 2246	TLS プロトコル v1.0 (The TLS Protocol Version 1.0)
RFC 3281	認可のためのインターネット属性証明書プロファイル (An Internet Attribute Certificate Profile for Authorization)
RFC 3075 RFC 3275	XML 署名文法スおよび処理 (XML-Signature Syntax and Processing) ※RFC 3075 は、RFC 3275 により廃止
RFC 3076	正則な XML バージョン 1.0 (Canonical XML Version 1.0)
RFC 3741	排他的 XML 正規化 バージョン 1.0 (Exclusive XML Canonicalization, Version 1.0)

4.2.6 ISIS-MTT

ISIS-MTT(Industrial Signature Interoperability Specification -MailTrusT)は、電子署名、暗号、PKI のための TeleTrusT と T7 Group の共通の仕様となっている。TeleTrusT(http://www.teletrust.de)は「情報と通信技術の信頼性向上」を目的とする非営利組織である。

ISIS-MTT 仕様は、相互運用可能な PKI アプリケーションで使用すべきデータフォーマットと通信プロトコルを定義している。既存の国際標準をベースとしていて、次の基本仕様から構成される：「Part 1：証明書と CRL プロファイル」「Part 2：PKI 管理」「Part 3：メッセージのフォーマット」「Part 4：操作プロトコル」「Part 5：証明書パスの検査」「Part 6：暗号アルゴリズム」「Part 7：暗号トークンのインタフェース」。

4.2.7 ISO(http://www.iso.ch/iso/en/ISOOnline.openerpage)

ISO(International Organization for Standardization)は、国際的な広範な技術標準を策定維持する機関で 146 カ国の国の機関や企業がメンバーとなっている。情報技術では IEC とのジョイント技術委員会 ISO/IEC/JTC1 が中心的な役割を果たしている。セキュリティ関係では Information technology -- Security techniques 分野があり、暗号技術、認証、セキュリティ基準を策定している。PKI の中心である公開鍵証明書の標準は ITU-T とのジョイントで OSI 分野の Directory シリーズのひとつとして Information technology -- Open Systems Interconnection -- The Directory: Public-key and attribute privacy frameworks が、いわゆる X.509 を定めている。

本ハンドブックで関係する標準を示す。

標準	タイトル
ISO/IEC 9798-3 (JIS X 5056-3)	セキュリティ技術―エンティティ認証―第3部：デジタル署名技術を用いる機構 Information technology-Security techniques-Entity authentication-Part3: Mechanisms using digital signature techniques
ISO/TS 17090	保健医療情報―公開鍵基盤―第1部: フレームワークと概要 ISO/TS 17090-1:2002 Health informatics -- Public key infrastructure -- Part 1: Framework and overview 保健医療情報―公開鍵基盤―第2部: 証明書プロファイル ISO/TS 17090-2:2002 Health informatics -- Public key infrastructure -- Part 2: Certificate profile 保健医療情報―公開鍵基盤―第3部: 認証極のポリシー管理 ISO/TS 17090-3:2002 Health informatics -- Public key infrastructure -- Part 3: Policy management of certification authority

4.2.8 ITU-T(http://www.itu.int/home/index.html)

ITU-T[8](国際電気通信連合電気通信標準化部門)は元 CCITT[9](国際電信電話諮問委員会)といわれた機関で、国際的な通信技術の標準 Recommendation を定める機関である。情報通信分野ではXシリーズの標準を策定しており、OSI 分野ではISOとのジョイント標準を定めている。X.509公開鍵証明書の標準は X.500 Directory シリーズの一つとして ITU-T が定め、ISO がジョイント標準としている。

本ハンドブックで参照している標準を示す。

標準	タイトル
X.509	ITU-U 勧告「情報技術-オープンシステム相互接続-ディレクトリ: 認証フレームワーク」 Recommendation X.509, "Information Technology--Open Systems Interconnection--The Directory: Authentication Framework" ISO 9594-8 と同じ

4.2.9 OASIS(http://www.oasis-open.org/home/index.php)

OASIS(Organization for the Advancement of Structured Information Standards)は、もと SGML Open としてビジネス分野で SGML 関連の標準化や普及に努めてきたが、1998 年に OASIS として改組され現在 600 以上のメンバーから構成され、XML を中心にしたビジネスフレームワークの協調、Web サービス、セキュリティ、業界アプリケーションの標準などを策定している。2002 年 9 月にはもと PKI Forum を併合し OASIS PKI TC として PKI の普及を目指している。セキュリティ関連では Security Service TC が SAML を策定し、Access Control TC でアクセス制御ポリシー言語 XACML、Web Service Security TC で Web サービスにおけるセキュリティ・トークンの仕様を定めている。また Digital Signature Service TC で署名サーバ、署名検証、XML タイムスタンプなどを検討している。OASIS は W3C(4.2.10 参照)が定めた XML 署名や SOAP や WSDL などの基本仕様の上に、Web サービスのセキュリティ強化を図っている。

4.2.10 W3C(http://www.w3.org/)

W3C(World Wide Web Consortium)は 1994 年に WWW の推進を図るために創設され現在 400 以上のメンバーで構成されている。初期の WWW は URL で指定した HTML のコンテンツを HTTP で転送するものであったが、現在は XML をベースに XHTML やセキュリティメカニズムとして XML 署名、XML 暗号、XKMS を備え、Web サービスやセマンティック Web などのメカニズムを備え豊かなアプリケーションを支える仕組みになってきた。W3C はこれらの標準を策定し Recommendation として仕様公開している。

本ハンドブックで参照している標準を示す。

標準	タイトル
XKMS2.0	XML 鍵管理仕様 バージョン 2.0 XML Key Management Specification(XKMS) version 2.0 X-KISS(鍵情報サービス)と X-KRSS(鍵情報サービス)

[8] International Telecommunications Union Telecommunication Standardization Sector
[9] Consultative Committee for International Telephony and Telegraphy

4.3 参考文献

[e-auth] e-Authentication, http://www.cio.gov/eauthentication/

[EAP] Electronic Authentication Partnership, http://www.eapartnership.org/default.htm

[FirstGov] FirstGov.gov: The U.S. Government's Official Web Portal, http://www.firstgov.gov/

[FICC] Federal Identity Credentialing Committee, http://www.cio.gov/ficc/

[OMB M-04-04] e-Authentication Guidance for Federal Agencies, Joshua B. Bolten(Director, Executive Office of the President, Office of Management and Budget), December 2003, http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf

[NIST sp800-63] NIST Special Publication 800-63 Electronic Authentication Guideline, Recommendations of the National Institute of Standards and Technology Version 1.01, September 2004, William E. Burr, Donna F. Dodson, W. Timothy Polk, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63v6_3_3.pdf

[BurtonGroupReport] Burton Group Report on the Federal e-Authentication Initiative, Daniel Blum, Gerry Gebel, Doug Moench, August 2004, http://www.cio.gov/eauthentication/documents/BurtonGroupEAreport.pdf

[Cookbook] e-Authentication Cookbook, July 2004, http://www.cio.gov/eauthentication/documents/Cookbook.pdf

[GOVhandbook] e-Authentication Federal Agency Handbook, July 2004, http://www.cio.gov/eauthentication/documents/GOVhandbook.pdf

[CSPhandbook] e-Authentication Credential Service Provider Handbook, July 2004, http://www.cio.gov/eauthentication/documents/CSPhandbook.pdf

あとがき

今回、電子商取引推進協議会の認証公証 WG において、ここ数年にわたって多面的に検討されてきた属性認証について、ハンドブックのかたちでまとめることが出来た。

属性認証については、米国においても e-Authentication などの大きなプロジェクトも動き始めているなど、現在まさに注目されているテーマであり日々検討が進んでいる分野である。今回、出来るだけ最新の項目を入れて本ハンドブックを作成したつもりである。

現在、電子認証の利用は広がりつつあるが、本格的な利用の拡大をむかえるには、実社会であたりまえに行われている属性認証をネット上で行う必要がある。実現方式としては、これまでいくつか提案されているが、属性認証が広く使われるには至っていない。また、属性認証に関するまとまった書物、文献もまだ出ていない状況である。そこで、属性認証の導入を考える際に本報告書を活用していただければ幸いである。

今後の課題として、属性認証システムを実際に構築、運営していくための技術的支援が考えられる。すなわち、電子契約等の属性情報が活用されるシステムについての情報収集、分析を行うことにより属性情報が活用されかたについて分類整理し、そのうちのひとつのモデルについて試験システムを構築して、実用に耐えうる属性認証システムのガイドラインの作成を進めたいと考えている。

付録 1　XML 署名

付録 1.1　概要(署名を構成する要素の説明を含む)

近年、アプリケーション間でデータ交換をするためのデータフォーマットとして XML が利用されるようになって来た。企業間の取引に XML 文書を利用するようになると、XML 文書の保護の仕組みが必要になる。XML 文書の保護とは、大雑把に言えば以下の 4 つであろう。

① XML 文書の盗聴を防止すること。
② XML 文書の改ざんを防止すること。
③ XML 文書の作成者が特定できること。
④ XML 文書作成の否認を防止すること。

これら 4 つの要件のうち、②～④はデジタル署名によって実現可能であり、CMS(Cryptographic Message Syntax) などの従来技術を利用しても実装することが可能である。CMS ではデジタル署名の対象となるデータの内容については意識せず、単なるバイナリデータとして取り扱うため、XML の持つ構造化されたデータ形式や、自由度の高い構文などについては意識されることが無かった。この欠点を解消するため、W3C と IETF が 1999 年から合同で作業を進め、2001 年に「RFC 3075 XML-Signature Syntax and Processing」として XML ベースのデジタル署名のメッセージフォーマット(XML 署名) が標準化された。その後 2002 年に RFC 3275 に改定されている[RFC2807][RFC3275]。

XML 署名の設計指針の主なものは以下。

① デジタルデータ特に XML 文書にどのように署名(sign) するかを定め、署名(signature) を表現する XML のシンタックスを XML 署名と呼ぶ。

② XML 署名のシンタックスは任意のアプリケーション／署名の信頼方式をサポートできるよう、拡張できなくてはならない。

③ XML 署名は、署名を交換するために必要なシンタックスや処理ルールを規定するが、署名の信頼方式については規定しない。

④ XML 署名のレベルでは、署名を暗号的に検証するために必要な鍵の情報を取り扱えるようにする。アイデンティティやキーリカバリのための情報を必要とするアプリケーションもありえるが、XML 署名の範囲においてはそのような情報は必ずしも必要としない。

このような設計指針のもとに開発された XML 署名であるが、以下のような特徴がある。

- 署名対象、署名アルゴリズムや署名値および証明書などを XML の文法で統一して表現できる。
- デジタル署名が XML タグ付き言語であり、ASN.1 構文に比べてわかりやすい。
- 任意のデータファイルや XML 文書の全体だけでなく、XML 文書の一部に対しても署名を付けることができ、部分署名や多重署名などの複雑な要件に対応できる。
- XML 署名で参照する暗号アルゴリズムなどのオブジェクト識別子は、ASN.1 が OID(Object Identifier)を指定するのに対して、W3C などで定めている URI を参照する。

XML 署名に登場する要素タグとその構造を以下に示す。この構造の中で、“?”はその直前に記述されたタグまたは属性がオプションであることを示し、“+”はその直前に記述されたタグまたは属性が 1 回以上出現することを示し、“*”はその直前に記述されたタグまたは属性が 0 回以上出現することを示す繰り返し指定子で、構造を示す為に付け加えたものである。実際の XML 文書の中には現れない。繰り返し指定しが指定されていないタグまたは属性は一回だけ(必須である)その位置に出現する。また、繰り返し指定子が影響する範囲を明確にする目的で適宜( )を加えているが、これも実際の XML 文書の中には現れない。

```
<Signature (Id=ID)?>
   <SignedInfo (Id=ID)?>
      <CanonicalizationMethod Algorithm=anyURI/>
      <SignatureMethod Algorithm=anyURI/>
     (<Reference (Id=ID)? (URI=anyURI)? (Type=anyURI)?>
        (<Transforms>)?
         <DigestMethod Algorithm=anyURI>
         <DigestValue>
      </Reference>)+
   </SignedInfo>
   <SignatureValue>
  (<KeyInfo (Id=ID)?>)?
  (<Object (Id=ID)? (MimeType=string)? (Encoding=anyURI)?>)*
</Signature>
```

図 付 1-1 XML 署名の構文

以下に各タグの内容について説明する。

表 付 1-1 XML 署名の主なタグ

タグ	内容
Signature	XML 署名要素の始まりを示す。
SignedInfo	正規化や署名のアルゴリズム、ダイジェストアルゴリズム、ダイジェスト値などを指定する。署名を行う前にデータの変換を行う必要がある場合にはそのアルゴリズムを指定することも可能。
CanonicalizationMethod	正規化のアルゴリズムを指定する。RFC 3275 では以下の 2 種類の正規化アルゴリズムが規定されている。 ・ Required Canonical XML http://www.w3.org/TR/2001/REC-xml-c14n-20010315 ・ Recommended Canonical XML with Comments http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments

SignatureMethod	署名アルゴリズムを指定する。ここで指定するアルゴリズムに、署名に使用する全ての暗号アルゴリズムの指定が含まれている。そのアルゴリズムとは、ダイジェストアルゴリズム、公開鍵暗号アルゴリズム、MAC、パディングなどである。RFC 3275 では以下の 2 種類の署名アルゴリズムが規定されている。 ・ Required DSAwithSHA1 http://www.w3.org/2000/09/xmldsig#dsa-sha1 ・ Recommended RSAwithSHA1 http://www.w3.org/2000/09/xmldsig#rsa-sha1
Reference	ダイジェストアルゴリズム、ダイジェスト値、署名対象オブジェクトの識別子(オプション)、変換アルゴリズムなどを指定する。
Transforms	変換アルゴリズムを指定する。変換アルゴリズムは、ダイジェスト値を計算する前にデータの変換を行う必要がある場合に、その変換アルゴリズムを指定する。変換アルゴリズムは複数指定可能であり、署名対象データに対して前の変換アルゴリズムの出力が次の変換アルゴリズムの入力となるように、指定された順序で適用される。RFC 3275 では以下の 5 種類の変換アルゴリズムを使用可能としている。 ・ 全ての正規化アルゴリズム ・ base64 エンコーディング http://www.w3.org/2000/09/xmldsig#base64 ・ Optional XSLT http://www.w3.org/TR/1999/REC-xslt-19991116 ・ Recommended Xpath http://www.w3.org/TR/1999/REC-xpath-19991116 ・ Required Enveloped Signature http://www.w3.org/2000/09/xmldsig#enveloped-signature
DigestMethod	ダイジェストアルゴリズムを指定する。RFC 3275 では以下のダイジェストアルゴリズムを使用可能としている。 ・ Required SHA1 http://www.w3.org/2000/09/xmldsig#sha1
DigestValue	ダイジェスト値を xs:base64Binary 型で設定する単純内容モデルの要素。
SignatureValue	実際の署名値を xs:base64Binary 型で設定する単純内容モデルの要素。
KeyInfo	受信者が署名の検証を行う時に使用する鍵を取得できるようにする。この要素には、鍵、名前、PKC、その他公開鍵管理に関する情報、などが設定できる。
Object	署名対象データを示す。

付録 1.2　XML 署名の特徴

XML 署名には CMS 等にはない特徴がいくつかある。その一つは、署名対象データと署名データの持ち方の違いで、以下の 3 種類の署名形式があることである。

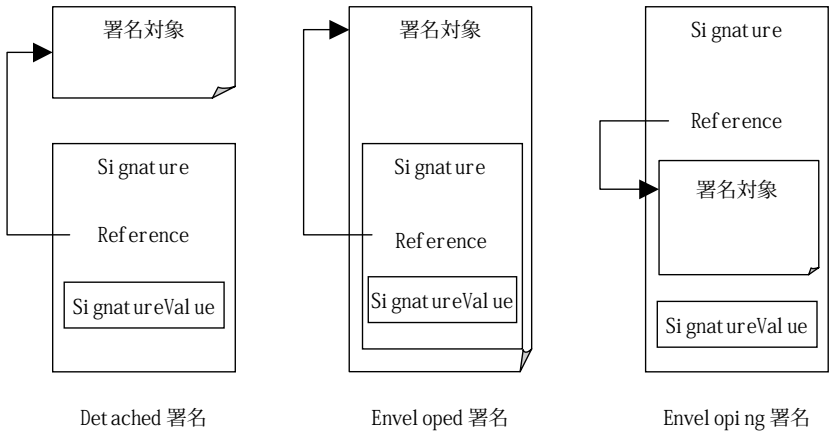


図 付 1-2 XML 署名の署名形式([@ITXMLSIG]より引用)

- Detached 署名
  Signature 要素が署名対象とは独立した要素になっている署名形式である。署名対象と署名が異なる XML 文書になっている場合や、署名対象が XML 文書でなく別ファイルで公開されているような場合もこの形式が使用できる。
- Enveloped 署名
  Signature 要素が署名対象の子要素となっている場合である。Reference 要素には Signature 要素を含む署名対象全体への URL が設定されるが、実際の署名対象は署名対象から Signature 要素を除いた部分が対象となる。同一の XML 文書に複数人が署名をつける場合などにこの形式が使用できる。
- Enveloping 署名
  署名対象が Signature 要素の子要素となっている場合である。署名対象は Object 要素の中に設定される。既存の署名付き文書に署名付きのコメントを加えたりする場合などにこの形式が使用できる。

二つ目は XML 署名の場合、署名対象として XML 文書の一部を指定できることである。Reference 要素の指定の仕方によって、XML 文書全体の中の一部に対して署名を行うことができる。そのため、一つの文書の中に署名がかかって変更不可になっている部分と、自由に追記したり修正したりできる部分の両方を持たせることができる。

三つ目は複数の署名対象に同時に署名をすることができることである。SignedInfo の Reference 要素は複数指定することができるため、それぞれの Reference 要素に異なる署名対象のダイジェスト値を設定することが可能である。その後、SignedInfo 要素に対して署名の処理を行うことによって、複数の署名対象に一回の演算で署名を行うことができる。

付録 1.3　正規化(Canonicalization)

XML 文書は、文字のエンコード方法やデータ表現に柔軟性があり、同一の文書であっても、その文字列表現は無数に存在しうる。デジタル署名は署名対象をバイト列として処理するため、XML 文書にデジタル署名を行うにあたっては、署名対象の XML 文書を一定のルールで正規化する必要がある。RFC 3275 では Canonical XML と Canonical XML with Comments の 2 つの正規化アルゴリズムを規定している［RFC3076］。両者の違いは正規化された XML 文書にコメントを含めるかどうかの違いで、基本的なアルゴリズムには大きな違いはない。Canonical XML によって正規化された XML 文書を比較する場合、正規化後の XML 文書が同一であれば正規化前の XML 文書は論理的に同一であると言えるが、正規化後の XML 文書が異なっている場合に、正規化前の XML 文書が論理的に異なっているとは言えない点には注意が必要である。

Canonical XML では XML 文書に主に以下のような処理を行うことによって XML 文書を正規化する。

- 文書は UTF-8 でエンコードする。
- 改行文字は#xA に正規化する。
- 属性値は XML パーサーが処理する際と同じように正規化する。
- 文字参照と実体参照を置き換える。
- CDATA セクションは文字表現に置き換える。
- XML 宣言とドキュメントタイプ宣言(DTD)は削除する。
- 空要素は開始タグと終了タグのペアに置き換える。
- 文書要素の外と、タグの中のホワイトスペース文字は正規化する。
- 要素内容のテキストに含まれるホワイトスペース文字はそのまま残す(改行の正規化によって取り除かれた文字は除く)。
- 属性値を囲む区切り文字はダブルクォートに置き換える。
- 属性値と要素内容のテキストに含まれる特殊文字は文字参照に置き換える。
- 余分な名前空間宣言は各要素から削除する。
- デフォルト属性は各要素に付け加える。
- 各要素の名前空間宣言と属性は辞書順に並べ替える。

XML 署名では、XML 文書の一部の要素に対して署名を行う場合があるため、Canonical XML でも XML 文書の一部の要素だけを取り出して正規化できる必要があるが、その際文書全体で必要となる名前空間宣言を、取り出した要素に全て引き継ぐ仕様になっているため、取り出した要素の中では不要な名前空間宣言がその要素に加えられてしまう。その為、文書の一部を取り出した場合に不要な名前空間宣言は、正規化した要素に加えない仕様の正規化アルゴリズム(Exclusive XML Canonicalization)が 2004 年 3 月に RFC 3741 として標準化されている［RFC3741］。

付録 1.4　参考文献

[RFC2807] RFC 2807 XML Signature Requirements
http://www.ietf.org/rfc/rfc2807.txt

[RFC3275] RFC 3275 XML-Signature Syntax and Processing
http://www.ietf.org/rfc/rfc3275.txt

[@ITXMLSIG] @IT:XML デジタル署名と XML 暗号
http://www.atmarkit.co.jp/fsecurity/rensai/webserv02/webserv01.html

[RFC3076] RFC 3076 Canonical XML Version 1.0
http://www.ietf.org/rfc/rfc3076.txt

[RFC3741] RFC 3741 Exclusive XML Canonicalization, Version 1.0
http://www.ietf.org/rfc/rfc3741.txt

付録 2　XKMS(XML Key Management Specification)

付録 2.1　概要

XML 署名や XML 暗号において PKC を前提とした仕様の策定が進められるにつれて、PKC の申請／取得や、PKC の失効申請に XML ベースのプロトコルが必要だという機運が高まってきた。また XML の処理系は持つが ASN.1 の処理系を持たず、PKC の裏にあるインフラを意識せずに PKC を利用するシンプルなクライアントのための、洗練された XML ベースの公開鍵管理方式が必要との要求もあって、XKMS が開発された［XKMS］。

XKMS は X-KRSS(XML Key Registration Service Specification) と X-KISS(XML Key Information Service Specification) の二つの仕様からなっている。

表 付 2-1 X-KRSS と X-KISS

X-KRSS	公開鍵の登録／PKC の発行と、その後の管理を行うためのプロトコル。
X-KISS	XML 署名や XML 暗号、SAML で使用される XML 署名の<ds:KeyInfo>の処理の一部または全部を XKMS サーバに委任することができるようにするためのプロトコル。

また、当初は大量発行用の仕様として X-BULK があったが、XKMS2.0 からは一つのリクエストの中に複数の X-KRSS／X-KISS リクエストを入れ子にして含めるコンパウンド・リクエストの方法に統合されている。

付録 2.2　X-KRSS

X-KRSS では公開鍵証明書に関する以下の操作を行うことができる。

表 付 2-2 X-KRSS の操作

Register	公開鍵を登録し、他の情報と結びつけ、PKC を発行する。クライアントが公開鍵ペアを生成する方法と、X-KRSS サーバが公開鍵ペアを生成する方法がある。クライアントが公開鍵ペアを生成する場合には、X-KRSS サーバに秘密鍵を送付する必要が無く、公開鍵だけを送付すればよいが、公開鍵に対応した秘密鍵を持っていることを確認できるよう、送付した公開鍵で検証可能な署名を公開鍵と一緒に送付する必要がある。
Reissue	過去に登録済みの鍵に対して、PKC を再発行する。クライアントが Reissue リクエストを行う主な理由は、X-KRSS サーバに新しい公開鍵証明書を生成させることである。
Revoke	過去の登録済みの PKC を失効させる。
Recover	PKC に対応する秘密鍵を回復させる。秘密鍵を回復させるためには、秘密鍵があらかじめ X-KRSS サーバに預託されていなくてはならない。秘密鍵の預託は、公開鍵の登録時に X-KRSS サーバによって公開鍵ペアを生成させる方法などによって達成される。秘密鍵の回復は鍵の危殆化を招く恐れがあるので、例えば、その秘密鍵を利用していたクライアントではなく管理者による鍵の回復要求が行われた際には、PKC の失効も同時に行うなどポリシー上の配慮が必要である。

付録 2.3　X-KISS

X-KISS は以下の 2 つのサービスをクライアントに提供する。

表 付 2-3 X-KISS のサービス

Locate サービス	XML 署名で規定されている<ds:KeyInfo>要素の解決を行う。例えば、クライアントが受け取った文書に XML 署名がされている場合に、そこに指定された<ds:KeyInfo>要素に<ds:KeyName>要素が指定されていたような場合、その<ds:KeyName>を X-KISS サーバに送付すると、X-KISS サーバが対応する PKC を探してクライアントに返してくれる、といったサービスである。X-KISS サーバは<ds:KeyInfo>要素の解決は行うが、その結果が<ds:KeyInfo>要素に指定されていたデータと正しく対応しているかどうかの検証は行わない。
Validate サービス	クライアントは Locate サービスが行う処理に加えて、公開鍵と名前や属性などその他のデータとの結びつきについての情報を受け取ることができる。クライアントはどのようなデータを Validate サービスに要求するか、そのプロトタイプを指定することができる。さらに Validate サービスは、サービスが提供するデータが有効かどうかを示すステータスと、それらが同一の公開鍵に結びついていることを示す。

付録 2.4　参考文献

[XKMS] XML Key Management Specification(XKMS 2.0)Version 2.0
http://www.w3c.org/TR/xkms2/

用語集

1. CRL(Certificate Revocation List)

証明書失効リスト。認証局が発行するデータで、有効期限内に失効された証明書のシリアル番号の一覧が記載されている。CRL は、認証局の電子署名によって改ざんできない形式となっている。

2. ID(Identification)

ID とは本人を確認(Authentication)するためのユニークな情報である。

指紋・虹彩・DNA などのバイオメトリクス情報は代表的な ID 情報だが、一般的に ID は単一情報で構成されるとは限らない。

3. OCSP(Online Certificate Status Protocol)

検証局等に対して、証明書が失効されているかどうかという確認をオンラインで問い合わせるためのプロトコル。

4. PKI(Public Key Infrastructure)

公開鍵暗号を利用して各種情報通信システムの安全性を確保するための一連の技術およびサービス。

5. PKC 拡張領域

X.509v3 公開鍵証明書において標準領域以外に追加された領域。鍵使用目的のような標準拡張と独自拡張がある。

6. 改ざん(改竄)

データを自分の都合のいいように改変する不正行為。

7. 鍵ペア

公開鍵暗号方式で利用する組となる二つの鍵。公開鍵と秘密鍵とからなる。

8. 危殆化

秘密鍵等の秘密情報が盗難、漏洩、解読などといった様々な原因によって、その機密性を失うこと(失ったものと想定されること)。

9. 検証局

証明書が失効されているかどうかという検証者からの問い合わせを受け付け、応答する機関。VA(Validation Authority)や OCSP Responder とも呼ぶ。

10. 検証者

署名検証を行う人。

11. 公開鍵

公開鍵暗号方式で利用する鍵ペアのうち、広く一般に開示する鍵。検証者が署名検証を行う際に使用する。

12. 公開鍵暗号方式

電子文書を暗号化する際に使用する鍵と、暗号文を復号する際に使用する鍵とが異なる暗号方式。公開鍵暗号方式には、どちらか一方の鍵からもう一方の鍵を算出することが非常に困難であるという性質と、二つの鍵は 1 対 1 対応であって、どちらか一方の鍵で暗号化したデータはもう

一方の鍵でのみ復号可能であるという性質とがある。公開鍵暗号方式は、電子署名を実現する手段として利用される。

13. 証明書

公開鍵とその所有者(署名者、または認証局)とを対応付けるために、認証局が生成する電子データ。認証書、電子証明書、あるいは公開鍵証明書などとも呼ぶ。証明書は、認証局の電子署名によって改ざんできない形式となっており、所有者の名前や公開鍵の値だけでなく、発行した認証局の名前や有効期限や利用目的などといった情報も含まれている。市区町村役場や登記所で発行される印鑑証明書に相当する。

14. 証明書の失効

秘密鍵の危殆化等のため、有効期間内の証明書の効力を失わせる行為。証明書の所有者(署名者、または認証局)の指示に基づいて行われる。

15. 証明書の有効性確認

検証者が、署名検証に使用する証明書が失効されていないかを確認する行為。確認の方法として、CRL に記載されているかどうか調べる方法や、検証局に OCSP でオンライン問い合わせをする方法などがある。

17. 証明書ポリシー

認証局が証明書を発行するにあたって設定するサービスや運用等に関する方針や規定。CP(Certificate Policy)とも呼ぶ。

18. 署名検証

署名付き電子文書を、署名者証明書に含まれる署名者の公開鍵を用いて復号することにより、当該電子署名の正当性(署名者によって生成された電子署名であることや、署名付き電子文書が改ざんされていないこと)を検証する行為。紙の文書に付された印影を印鑑証明書等に記された正しい印影を用いて照合する場合に相当する。

19. 署名者

署名生成を行う人。

20. 署名生成

電子文書に対して、署名者の秘密鍵を用いて暗号化することにより電子署名を施し、署名付き電子文書を生成する行為のこと。紙の文書に捺印する場合に相当する。

21. 信頼点、信頼点情報

利用者が信頼する認証局の証明書。通常は、自分の証明書を取得した認証局のルート認証局であることが多い。信頼点は、検証者が署名者証明書の正当性を確認する際に利用される。

22. 属性(Attribute)

属性とは対象者に与え(Authorization)られた資格や権限、職責、地位等をあらわす情報である。ID が単独で使用されることがあるのに対し、属性は単独で使用されることはない。

23. 属性証明書(Attribute Certificate)

属性認証機関(AA)が発行し、証明書に添付する主体者の属性を指定するもの。公開鍵証明書がパスポートのようなもので、属性証明書は添付する査証(ビザ)のようなものである。属性証明書で定義する属性は、グループ名、組織名、セキュリティ区分などがある。

24. 耐タンパ性

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

25. 電子署名

署名対象となる電子文書、あるいはそのハッシュ値を秘密鍵で暗号化したもの。一般には、タブレット等によって入力された手書きサインも含めて電子署名と呼び、前記秘密鍵で暗号化したものをデジタル署名と呼びわける場合もあるが、本ガイドラインでは、公開鍵暗号方式に基づいて生成されたものだけを電子署名、あるいは単に署名と呼んでいる。

26. 電子署名法

平成13年4月より施行された「電子署名および認証業務に関する法律(平成12年5月31日法律第102号)」の略称。電子署名に対して印鑑と同等の推定効を与えている法律。

27. 電子認証システム

電子署名を用いて、通信相手の確認や通信メッセージの改ざんチェックなどを行うシステム、および証明書の発行など、電子署名を正しく利用するために必要な処理を行うシステム。なりすましや改ざん、否認などといった不正を防ぐ目的で用いられる。

28. なりすまし

他者のふりをする不正行為。

29. 認証局

公開鍵とその所有者とを対応付けるために、署名者または他の認証局に対して証明書を発行する機関。CA(Certification Authority)とも呼ぶ。また、自己の証明書を自分自身で発行する認証局をルート認証局とも呼ぶ。

30. 認証局運用規定

証明書ポリシーに基づいて、認証の実施における手続きや遵守事項等を文書化したもの。CPS(Certification Practice Statement)とも呼ぶ。一般に、利用者等に対して開示される。

31. 認証情報

ある利用者を他の利用者と区別するために用いられる情報。パスワードや生体情報等。

32. ハッシュ関数

電子文書に電子署名を施す際などに、その電子文書をある一定の大きさまで圧縮するための計算手順。ハッシュ関数の計算結果である圧縮データをハッシュ値、あるいはメッセージダイジェストと呼ぶ。ハッシュ関数には、あるハッシュ値が与えられたときに、それと同じハッシュ値となるような電子文書を求めることが困難であるという性質(一方向性)と、同じハッシュ値となる二つの異なる電子文書を探し出すことが困難であるという性質(衝突回避性)がある。

33. 否認

取引などを行った後に、当該取引に関与したことそのものを否定する不正行為。事後否認とも呼ぶ。

34. 秘密鍵

公開鍵暗号方式で利用する鍵ペアのうち、署名者自身が秘密に保持する鍵。署名生成時に使用する。

35. リポジトリ

証明書やCRL等を保管しておき、利用者からの要求に応じてそれら情報を配布する仕組み。

36. 利用者

電子署名技術を利用する人。署名者と検証者に区分される。

メンバーリスト

事務局

前田　陽二	電子商取引推進協議会(ECOM)	主席研究員
川松　和成	電子商取引推進協議会(ECOM)	主席研究員

顧問

大山　永昭	東京工業大学　教授
菅　　知之	関西大学　　教授
平田　健治	大阪大学　大学院　教授

編集メンバー

氏名	会社名
武藤　　裕	NTT コミュニケーションズ株式会社
岩崎　公寛	株式会社 NTT データ
古田　健一	共同印刷株式会社
鈴木　優一	セコム株式会社
漆島　賢二　＊＊	セコム株式会社
佐藤　雅史	セコム株式会社
佐藤　永子	セコム株式会社
海川　正洋	株式会社帝国データバンク
古賀　祐匠	日本電信電話株式会社
千葉　昌幸	株式会社三菱総合研究所
坂上　　勉　＊＊	三菱電機株式会社
本山　信久	三菱電機株式会社
鍛冶　俊彦　＊	株式会社日本電子貿易サービス
有馬　純一郎＊	三菱電機情報ネットワーク(株)
谷口　展郎　＊	NTT 情報流通プラットホーム研究所

(注)　＊　はオブザーバ　　＊＊はリーダ

SWG1．2 メンバー(上記以外)

氏名	会社名
横井　雅彦	NTT コミュニケーションズ株式会社
石津　晴崇	NTT コミュニケーションズ株式会社
関野　公彦	株式会社 NTT ドコモ
黒木　美和	株式会社損害保険ジャパン
菅野　健司	株式会社帝国データバンク
浜田　　誓	電気事業連合会
小林　智恵子	株式会社東芝
島　　成佳	日本電気株式会社
冨田　清次	日本電信電話株式会社
井上　晴司	日本ベリサイン株式会社
下江　達二	富士通株式会社
冨高　政治	富士通株式会社
糸岡　　崇	富士電機ホールディングス株式会社
田中　　稔	三菱電機株式会社
増井　久之　＊	香川大学
東山　栄一　＊	NEC ソフト株式会社
篠崎　政久　＊	株式会社東芝

(注)　＊　はオブザーバ

平成16年度　経済産業省　受託事業
EC技術基盤の相互運用性に関する調査研究
(取引相手先の属性認証技術等の調査)
属性認証ハンドブック
平成17年2月発行

発行所　財団法人　日本情報処理開発協会
電子商取引推進センター
東京都港区芝公園三丁目5番8号
機械振興会館　3階
TEL：03(3436)7500
印刷所　新高速印刷株式会社
東京都港区新橋五丁目8番4号
TEL：03(3437)6365