参考資料３

経済産業省委託調査

平成１４年度ＥＣ技術基盤の相互運用性に関する調査研究事業

（取引相手先の属性認証技術等の調査）

属性認証の適用ガイドライン

平成15年３月

電子商取引推進協議会
財団法人日本情報処理開発協会
電子商取引推進センター

この報告書は、平成１４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１４年度ＥＣ技術基盤の相互運用性に関する調査研究事業（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。

はじめに

インターネットを中心としたITの急速な進展は国際的な電子商取引の期待へと繋がっている。とくに、情報ネットワークを介した電子商取引の信頼性・安全性確保は極めて重要である。電子商取引の普及には取引相手の信頼性保証(本人権限の認証、権限委譲、代理人資格の認証等)が必要あり、技術的には、本人認証技術と属性認証技術を組み合わせることが必要である。属性認証については､必要性が叫ばれながら､いまだ世界中で属性認証局なるものも立ち上がっておらず、本人認証との連携を踏まえた充分な検討が必要である。よって、本書では、その実現モデルと利用形態の両面から検討を行った。

本書をまとめるための検討は、以下のプロセスで推進された。

ⅰ)属性認証モデルの技術動向調査
属性認証モデルの技術動向や、IETF等の属性認証関連標準の調査を行った。

ⅱ)属性認証モデルの想定利用例の検討
属性認証シモデルの利用例をいくつか想定し、それぞれに関して、属性認証モデルに求められる機能要件、セキュリティ要件等を抽出し、議論した。

ⅲ)属性認証モデルの機能要件の検討
属性認証の想定利用例に基づいて、属性認証モデルの利用を、サービス提供者、運用者等の立場から検討し、属性認証モデルのシステム仕様(機能要件)と留意事項等をまとめた。

その結果、本書は以下のような構成を採っている。

「１．属性とその認証」では、検討に向けた共通認識のために、属性（attribute）とその認証について定義を行っている。

「２．属性認証と適用技法」では、属性の認証に関する要件を整理した上で、その適用技法について特徴を述べている。

「３．属性認証局のモデル」では、登場人物と実現モデルを定義した上で、属性証明書の発行、利用、失効の各運用プロセスについて考察する。

「４．属性認証の特徴と利用上の留意事項」では、属性認証の仕様、運用の詳細と、その特徴や利用上の留意事項を記述している。

「５．適用ガイド」では、多忙な読者のために、前章までのキーポイントをガイドとしてまとめている。

「６．XML技術と属性」では、参考情報として、最近話題になっているXMLおよびWebサービスの技術について、その属性認証との関係を解説している。

なお、属性認証を実現する主な技法として、公開鍵証明書を用いる技法、属性認証サーバを用いる技法、属性証明書を用いる技法が挙げられる。本書では、主に属性証明書を用いる技法を考察している。また、基本的な属性認証局のモデルの考え方として、RFC3281 属性証明書をベース

とするモデルと X.509 属性証明書をベースとするモデルが存在する。RFC3281 は X.509 をベースとしながらも、より実装を意識した観点から記述されており、簡潔で曖昧性がない。よって本書では、RFC3281 をベースとするモデルを中心に考察を進めた。

また、本書の姉妹編として、具体的な利用モデルについて考察した「属性認証の利用モデル」が発行されている。本書と併せてお読みいただき属性認証に関する考察を深めていただければ幸いである。

最後に、本書の中で使用される略語について以下に明記しておく。

AA（題名等では属性認証局と記述）：Attribute Authority
AC（題名等では属性証明書と記述）：Attribute Certificate
CA（題名等では認証局と記述）：Certificate Authority
PKC（題名等では証明書と記述）：Public key Certificate
AC 保有者：Attribute Certificate Holder
AC 検証者：Attribute Certificate Verifier
AC 利用者（AC 保有者と AC 検証者を総称して使用）
AC 発行対象者（特定の章で、定義した上で使用）

平成 15 年３月

財団法人日本情報処理開発協会
電子商取引推進センター
電子商取引推進協議会

目　次

図表一覧目次

1. 属性とその認証

本章では、本書の主題である属性や属性の認証とは何か、またその前提となる識別と認証の意味を解説する。

1.1 属性とは

電子商取引や電子申請、あるいは社内業務などのために、情報システムが構築される。この情報システムに着目すると、さまざまな主体が関わりを持ち、それを利用している。電子商取引においては、企業の代表取締役、支店長、営業部長や、営業担当者が情報システムを手段として活用する。

属性(attribute)とは、商取引等に関わる主体の資格、権限などである（注)。これは、商取引等の電子化有無に依存しない概念である。「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。

注) より正確には、属性とは、このような資格、権限の中でも特にある具体的な適用や業務の観点から関心の対象となるものをいう。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属性とはならない。

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

1.2 電子化における属性の利用

電子化以前の手続きでは、委任状や社員証あるいは各種免許の提示により属性が確認される。商取引や申請等の手続きを電子化する際には、属性の提示・確認の処理もあわせて電子化して情報システムで実現することが、効率や安全性の観点から求められる。電子化した属性は、情報システムの利用者に関する資格や権限の確認に利用される。

- 処理の実施やサービスの利用に関する資格、権限： 電子商取引、電子申請、インターネットショップ利用などの資格や権限を確認する。
- 情報の利用に関する資格、権限： 情報システムが持つ情報へのアクセスを管理する。

既存業務の電子化だけでなく、新たなビジネスモデルや手続きも属性を電子的に表現することにより見出せるものと思われる。

1.3 識別と認証

電子化された商取引等において属性を与えられる主体は、あらかじめ識別され、認証されてい

ることが必要である。

(1)　識別

識別(identification)とは、ある母集団の中で、その主体を他の主体から明確に区別することである。識別された主体は、他の主体と異なる情報により区別される。この情報を識別情報と呼ぶ。以下は識別と識別情報の代表的な例である。

・日本において住民登録された人を母集団として、個人は、氏名、性別、生年月日、住所（あわせて住民基本4情報という）により識別される。

・ある企業の全社員を母集団として、個々の社員は社員番号により識別される。

一般に、属性は時間の経過とともに変更されたり、付与、削除がなされたりする。これに対して、識別情報はその目的からも、変化しにくいものが選ばれる。

(2)　認証

認証(authentication)とは、その主体と識別情報の対応づけを行うことである。その方法には

・当該主体のみが知るパスワードを利用する方法

・当該主体のみが保有する物(IDカード等)を利用する方法

・指紋などの生体認証

・PKI 技術に基づく電子認証

などがある。

PKI 技術に基づく電子認証では、第三者である認証局が当該主体と識別情報の対応を確認し、その結果として証明書（公開鍵証明書）を交付するが、これも認証(certification)と呼ぶ。

認証・公証WGにおける本検討では、電子商取引や電子申請などの電子化業務にPKIに基づく電子認証を採用している場合を重要な検討対象とする。この場合、属性は、証明書を持つエンドエンティティに関する属性である。

1.4　属性の認証

情報システムにおいて、属性の認証(authorization 等)とは、識別、認証された主体がある属性を持つことを他者が確認することをいう。また、確認の事実が電子的に表現され、検証できることも必要である。

属性の認証は公開鍵証明書によるエンドエンティティの認証とは別のものである。

1.5　識別情報と属性の表現

電子商取引や電子申請などへの適用に際して、識別情報と属性は様々な方法で表現し、管理することができる。運用、性能、利便性等の面でそれぞれの特徴があり、２章に示すように、適用に応じて適切な方法を選択することになる。

(1)　識別情報の表現の例

・認証局が発行する公開鍵証明書に記載する。
・サーバに保持するデータベースやディレクトリに記録する。
・識別情報を電子的に表現する必要のない場合もある。社内や組織内のシステムであらかじめ利用者が社員に限られていて、識別情報による認証が不要な場合がその例である。

(2)　属性の表現の例

・属性認証局が発行する属性証明書に記載する。
・認証局が発行する公開鍵証明書に記載する。
・サーバに保持するデータベースやディレクトリに記録する。
・インターネットショップにおいて、会員ごとの資格を会員データベースで管理する。
・社内システムの人事データベースに社員の職位、所属、担当業務を記録する。

2. 属性認証と適用技法

属性認証（属性の検証）を実現する技法には、属性証明書を用いる技法の他にも、証明書（PKC）を用いる技法や属性認証サーバを用いる技法もある。それぞれに運用管理面、システム構築・保守面や利用者の使い勝手の面の特徴があるので、業務の要件に適した技法を採用する必要がある。本章では、属性認証（属性の検証）を実現するための主な適用技法について、次の手順で机上評価を試みる。

- 属性認証の要件
  属性の継続性、属性の管理などの観点から、属性認証に対する主な要件を洗い出す。
- 属性認証の主な適用技法
  現技術水準で、属性認証を実現可能と考えられる主な技法を列挙する。
- 適用技法の特徴
  属性認証のための各適用技法を属性認証の要件に照らして机上評価し、各適用技法の特徴と考えられる事項を列挙する。

2.1 属性認証の要件

属性の継続性、属性の管理などの観点から、属性認証に対する主な要件を洗い出す。

2.1.1 属性の継続性に関する要件

属性には、継続性の高い属性がある。目安としては、年単位で有効な属性又は年単位でしか属性値が変化しないものがこれに該当する。何年も有効な資格は、その代表的なものである。

［例］継続性の高い属性

- 電子処方箋発行等における医師資格
- 電子申請等の代理人資格
  電子申請の代理人に相応しい資格。具体的には、申請先に応じて司法書士、行政書士、税理士、弁理士などの資格を意味する。

一方、属性には、継続性の低い属性もある。目安としては、月単位／日単位の有効性しかない属性又は月単位／日単位で属性値が変化するものがこれに該当する。

［例］継続性の低い属性

- 社内システムにおける社員の所属、担当業務、役割
- 電子商取引における代表取締役の社内代理人（営業部長など）

また、属性には、属性の変化に伴って速やかに無効にするための管理（失効管理）が必要なものと不要なものが考えられる。属性の失効管理の要否は属性認証を行うアプリケーションに応じて決定されるが、一般には継続性が高い属性に対しては失効管理の必要性が高く、継続性の低い属性に対しては失効管理の必要性が低いと考えられる。例えば、属性自身がチケットのような 1 回限りの属性認証に用いられるケースや、属性の有効期間が 1 日限りであることを設定・管理で

きるケースでは、属性の失効管理は不要であろう。

属性認証を行うアプリケーションでは、属性の継続性に関してそれぞれのアプリケーションに適した実現方法を採る必要があることから、適用技法に対する「属性の継続性に関する要件」として、次の要件が挙げられる。

- 継続性の高い属性を扱えること
- 継続性の低い属性を扱えること
- 属性に有効期間を設定・管理できること
- 属性の失効管理ができること

2.1.2 属性の管理に関する要件

属性は、属性付与者が責任を持って付与したり必要に応じて速やかに失効させたりするなど、適切に管理する必要がある。ここで、属性によっては、集中管理すべき属性と分散管理可能な属性が考えられる。分散管理可能な属性については、属性付与権限を適切に委譲することによって運用管理効率を向上できると考えられる。

［例］集中管理すべき属性

- 電子処方箋発行等における医師資格
  医師資格は国家資格であることから､厚生労働省が一元的に集中管理すべきであろう。なお、○○病院の外科部長などの職位については、各医療機関が付与すべき属性と考えられる。
- 電子申請等の代理人資格
  電子申請の代理人に相応しい資格であり、具体的には、申請先に応じて司法書士、行政書士、税理士、弁理士などの資格を意味する。これらの資格は、資格を付与する権限を有する各機関が一元的に集中管理すべきであろう。

［例］分散管理が適した属性

- 社内システムにおける社員の所属、担当業務、役割
  社内システムにおけるこれらの属性の管理は､職制の階層構成に従って権限委譲された各職制において、決定権を有する上長が直接的に管理することが実践的であり、運用効率面で有利であろう。

属性認証を行うアプリケーションでは、属性の集中管理／分散管理に関してそれぞれのアプリケーションに適した実現方法を採るべきであろう。属性付与権限の委譲においては、必要に応じて権限委譲先を確実に制限できることが望ましい。また、属性及び属性値の改ざんを防止する対策が必要である。さらに、属性によっては開示したくないものもあるので、必要に応じて属性を秘匿できることが望ましい。

これらのことから、適用技法に対する「属性の管理に関する要件」として、次の要件が挙げら

れる。

- 属性の集中管理ができること
- 属性付与権限を委譲することによって、属性の分散管理ができること
- 属性付与権限の委譲先を確実に制限できることが望ましい
- 属性及び属性値の改ざんを防止できること
- 属性を秘匿できることが望ましい

2.1.3 その他の要件

属性認証を行うアプリケーションシステムを構築・運用・保守する観点や利用する観点から、適用技法に対する「その他の要件」として、主に次の要件が挙げられる。

- システム構築が比較的容易なこと（証明書発行までの手間、属性認証の実現容易性等）
- 利用者側で属性認証のための検証処理を実行するアプリケーションを構築できること
- 複数の属性及び属性値を扱えること
- 利用者の属性及び属性値の変更が比較的容易であること
- 属性認証の速度性能が比較的良いこと
- 複数のアプリケーションに使用できる汎用的な属性認証基盤を実現できること
- 利用者の手間が比較的少ないこと（鍵・証明書類の選択操作が不要又は少ないこと）

2.2 属性認証の主な適用技法

現技術水準で属性認証（属性の検証）を実現可能と考えられる主な技法として、次の技法が挙げられる。

・AC を用いる技法
・PKC を用いる技法
・属性認証サーバを用いる技法

以下に、各技法の概要を示す。

2.2.1 属性証明書を用いる技法

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 によって利用者を識別・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。

2.2.2 証明書を用いる技法

利用者がある属性及び属性値を有することを PKC によって暗示又は明示し、PKC を用いた利用者認証によって属性認証する技法である。

(1) 利用者がある属性及び属性値を有することを PKC に暗示する技法

ある属性及び属性値を有する利用者だけに 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。

［備考］

他の属性及び属性値と対応付けた PKC や他の目的・用途の PKC と区別するためには、PKC の extension の certificatePolicies が保持するポリシーID で区別がつくようにするなどの対策を講じる必要がある。

(2) 利用者がある属性及び属性値を有することを PKC に明示する技法

利用者が保有する属性及び属性値を、PKC の extension（subjectDirectoryAttribute 又は private extension）に記載して PKC を発行する。PKC の extension を検証することによって、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。

［備考］

この技法によれば、PKC によって次のことが可能になる。

- 1 枚の PKC で、複数の属性及び属性値を取り扱える。
- 1 枚の PKC を、異なる属性及び属性値を検証する複数のアプリケーションに対して共通に使用できる。
- 共通の属性及び属性値について、複数の CA ドメインに亙って検証するアプリケーションを構築することもできる。

2.2.3 属性認証サーバを用いる技法

属性認証サーバが、各利用者の属性及び属性値を格納したサーバ上の属性データベースを用いて属性認証する技法である。ここで、属性データベースは、属性認証サーバの利用者情報テーブルに含める方法、ディレクトリサーバに含める方法などがある。

属性認証サーバは、PKC 等を用いて利用者を識別・認証できたら、属性データベースを用いて、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。

この技法は、典型的にはアプリケーションサーバに対するアクセス制御等で用いられている技法である。

2.3 各適用技法の特徴

属性認証のための各適用技法については、運用管理面、システム構築・保守面、利用者の使い勝手の面（操作性、性能等）においてそれぞれ特徴（一長一短）がある。それぞれの特徴を踏まえて、業務要件に適した技法を採用することが望まれる。

本節では、2.2 節に示した属性認証のための各適用技法について、2.1 節に示した属性認証の要件に照らしながら机上評価を試みた結果、主な特徴と考えられる事項を列挙してみる。

2.3.1 運用管理面

(1) 要件

- 継続性の高い属性を扱えること
- 継続性の低い属性を扱えること
- 属性に有効期間を設定・管理できること
- 属性の失効管理ができること
- 属性の集中管理ができること
- 属性付与権限を委譲することによって、属性の分散管理ができること
- 属性付与権限の委譲先を確実に制限できることが望ましい
- 属性及び属性値の改ざんを防止できること
- 属性を秘匿できることが望ましい

(2) 各適用技法の特徴

① AC を用いる技法では、属性の有効期間を明示的に設定・管理できる。
② PKC を用いる技法では、継続性の低い属性（一時的な属性）を取り扱いにくい。
③ 属性認証サーバを用いる技法では、複数の属性を集中管理しやすい。
④ ITU-T 勧告 X.509 仕様に基づく場合、AC を用いる技法では、属性付与権限を委譲することによる属性の分散管理を行いやすい。また、属性付与権限の委譲先を明示的に制限できる。
⑤ PKC を用いる技法及び AC を用いる技法では、属性及び属性値の改ざん防止が証明書（のディジタル署名）によって保証される。
⑥ 属性によっては秘匿することが必要になる。AC を用いる技法では、属性を暗号化する機能が標準的に用意されているので、属性の秘匿を実現しやすい。

2.3.2 システム構築・保守面

(1) 要件

- システム構築が比較的容易なこと（証明書発行までの手間、属性認証の実現容易性等）
- 利用者側で属性認証のための検証処理を実行するアプリケーションを構築できること
- 複数の属性及び属性値を扱えること
- 利用者の属性及び属性値の変更が比較的容易であること

- 複数のアプリケーションに使用できる汎用的な属性認証基盤を実現できること

(2) 各適用技法の特徴

① 利用者がある属性及び属性値を有することを PKC に暗示する技法では、市販の PKI 関連ソフトウェアを利用できる可能性も高く、システム構築が比較的容易である。一方、利用者がある属性及び属性値を有することを PKC に明示する技法や AC を用いる技法では、現状においては次のことからシステム構築が容易とは言えない。

- AC を用いる技法の場合は前提となる PKC が必要である。
- PKC／AC には複数の属性及び属性値を格納できるため、実際に利用する際には PKC／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 PKC／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と考えられる。
- 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。

② PKC を用いる技法及び 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が、属性認証サーバを用いる技法は、End-to-End モデルのアプリケーションよりも、クライアント－サーバ型のアプリケーションで、サーバがクライアントの属性を検証するモデルのアプリケーション構築に向いていると考えられる。

③ 属性及び属性値の追加／変更が必要になったら、属性認証サーバを用いる技法では、属性データベースの追加／変更によって容易に対応できる。AC を用いる技法では、利用者に対して AC を発行し直す（追加／変更する）ことによって比較的容易に対応できる。

④ 種々の属性を種々のアプリケーションで取り扱う必要があるような一般的状況に対応してシステム構築する場合や、属性及びアプリケーションを追加／変更して行ける保守性／スケーラビリティが求められる場合には、汎用的な属性認証基盤に基づいてシステム構築・保守することが望まれる。AC を用いる技法は、このようなニーズに最も適していると考えられる。

2.3.3 利用者の使い勝手の面（操作性、性能等）

(1) 要件

- 利用者の手間が比較的少ないこと（鍵・証明書類の選択操作が不要又は少ないこと）
- 属性認証の速度性能が比較的良いこと

(2) 各適用技法の特徴

① 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。

② 速度性能については、PKC を用いる技法が比較的速いと考えられる。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. 属性認証局のモデル

本章では、属性認証局(AA: attribute authority)の発行する属性証明書(AC: attribute certificate)を利用した属性の認証モデルについて検討する。

認証局(CA: certification authority)の発行する公開鍵証明書(PKC: public key certificate)に基づく個人の認証基盤が公開鍵インフラストラクチャ(PKI: public key infrastructure)と呼ばれるのに対し、属性証明書に基づく属性の認証基盤は権限管理インフラストラクチャ(PMI: privilege management infrastructure)と呼ばれる。公開鍵証明書は「この公開鍵の保有者は○○です」ということを証明するのに対し、属性証明書は「この保有者は○○の属性（資格・権利等）を持っています」ということを証明する。

PKC に対してその所有者の属性を記述すると、以下のような問題が生じる。

- 属性が変更されると PKC を破棄しなくてはならない
- PKC の有効期間と属性の有効期間は通常同じではない
  （一般に PKC の有効期間は長く、属性の有効期間は短い）
- CA が属性を付与することは、CA の本来の役割からすると適切でない場合がある
- 属性情報を公開したくない場合もある

以上のような問題を解決することを目的として、AA による AC の発行に基づく属性の認証の仕組みが考えられた。AC は、権利に応じたアクセス制御、および代理人への委任状などの広範な用途および環境に適用することができる。

本章の構成は以下の通りである。3.1 節で、当事者とモデルを整理し、本報告書で主に検討の対象とするモデルが RFC3281 をベースとするモデルであることを示す。次に、3.2 節で、そのモデルに基づいて AA を運用する際の、AC の発行、利用、失効の手順概要を述べる。

3.1 当事者とモデル

基本的な AA のモデルの考え方として、RFC3281 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。

- RFC3281 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 RFC3281 か X.509 かにより異なる。それぞれの概念モデルを図 3-1 に示す。図中の実線矢印は、PKC または AC の発行を意味する。

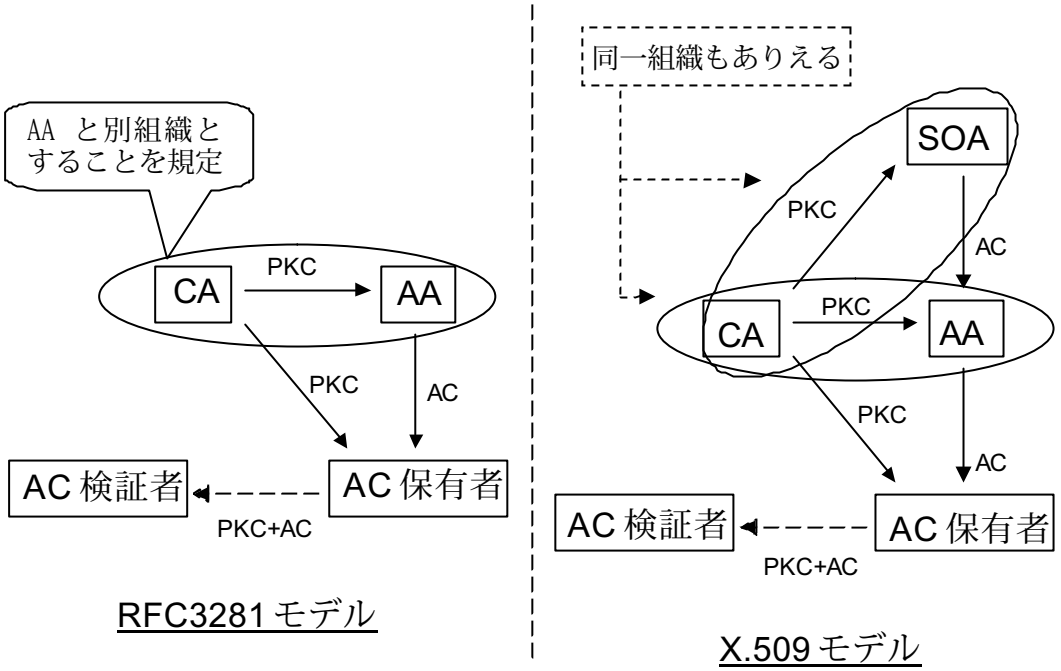


図 3-1 属性認証局のモデル構成

RFC3281 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。また、AC の配布方法を push 型と pull 型に分類することができる。

- push 型
  AC 保有者が AC 検証者に AC を送信することにより、AC 検証者が AC を入手する方式
- pull 型
  AC 検証者が AA またはリポジトリに AC 保有者の AC を要求することにより、AC 検証者が AC を入手する方式

ところで、コンピュータシステムを用いて実際にシステムを構築する観点においては、X.509 は多少複雑である。一方、RFC3281 は X.509 をベースとしながらも、より実装を意識した観点から記述されており、簡潔で曖昧性がない。よって、以降の章では RFC3281 をベースとするモデルに焦点を合わせ、考察していく。

図 3-2 に RFC3281 をベースとするモデルにおける AC の発行と利用のモデル分類を示す(一般的

に、利用時の AC の配布方法については push 型がシンプルであるため、pull 型は省略する)。

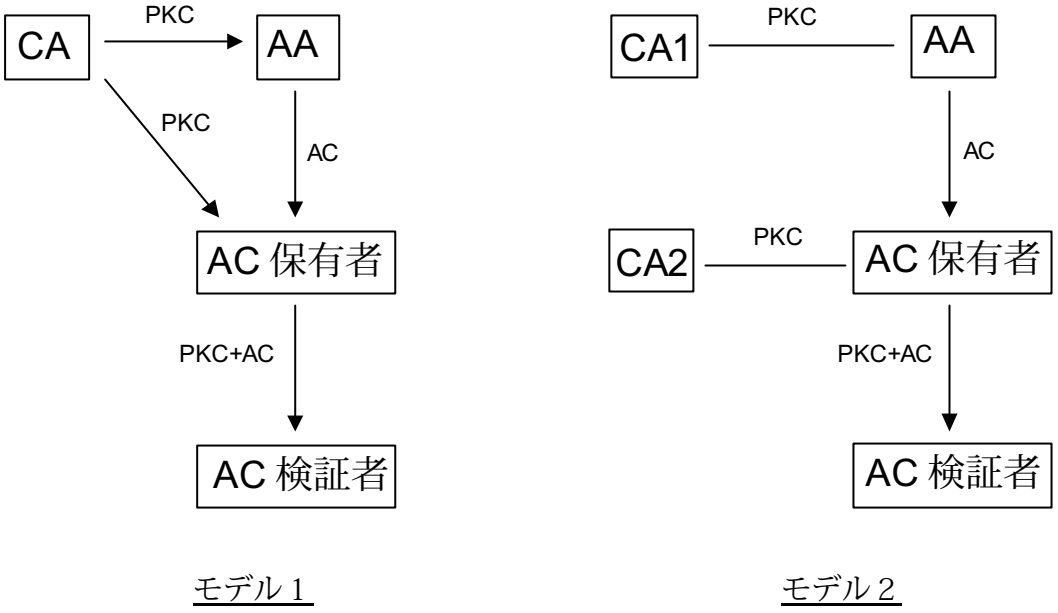


図 3-2 RFC3281 をベースとするモデルにおける AC の発行と利用（push 型）

[モデル 1]

1. AA は、CA から PKC の発行を受ける。
2. AC 保有者は、CA から PKC の発行を受ける。（この CA は 1. の CA と同一組織）
3. AC 保有者は、自身の PKC を AA に提出し、AC の発行を受ける。

[モデル 2]

1. AA は、CA1 から PKC の発行を受ける。
2. AC 保有者は、CA2 から PKC の発行を受ける。（CA1 と CA2 は別組織）
3. AC 保有者は、自身の PKC を AA に提出し、AC の発行を受ける。

3.2 運用手順

本節では、RFC3281 をベースとするモデルに基づいて属性認証局を運用する際に発生するいくつかの処理（属性証明書の発行、利用、失効）の手順を示す。

3.2.1 属性証明書の発行手順

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属性認証局に対し、属性証明書の発行依頼を行う。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。

以上を整理すると、AC の発行手順として、少なくとも以下の 2 つの方式が考慮されるべきと考えられる。ただし、あらかじめ AC 保有者は自身の PKC を保有しているとし、AC 保有者と PKC との対応関係が正当であることを前提とする。

[属性証明書の発行モデル]

(a) AC 保有者のオンライン申請に基づいて、属性証明書を作成、及び発行。

(b) 管理主体の申請に基づいて、属性証明書を作成、及び発行。

ここでは、AC 保有者のオフライン申請（属性認証局またはその登録局の窓口で対面により発行申請する場合や、郵送など紙媒体をベースにして行う申請）については、窓口担当者がオンライン申請における申請者の役割を代行するとみなし、(a) に含めるものとする。(a) の発行申請においては、AC 保有者の PKC に対応する秘密鍵による署名が必要であり、この署名検証が AC 保有者と PKC とのリンクを確認するための本質的な役割を果たす。

また、(b) について、属性情報を管理する「管理主体」と、属性認証局を運用する「運用主体」を別組織として考えているが、場合によっては両者が同一の組織である可能性もある。

上記に示した各方式における属性証明書の発行手順を以下に示す。

(a) AC 保有者のオンライン申請に基づいて、属性証明書を作成、及び発行

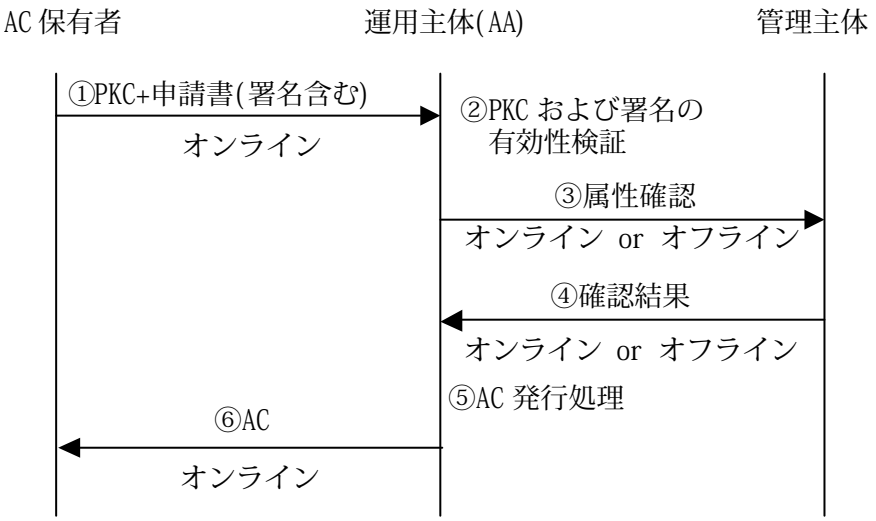


図 3-3 AC 保有者のオンライン申請に基づく発行

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 に送信する。

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 を送信する。

(b) 管理主体の申請に基づいて、属性証明書を作成、及び発行

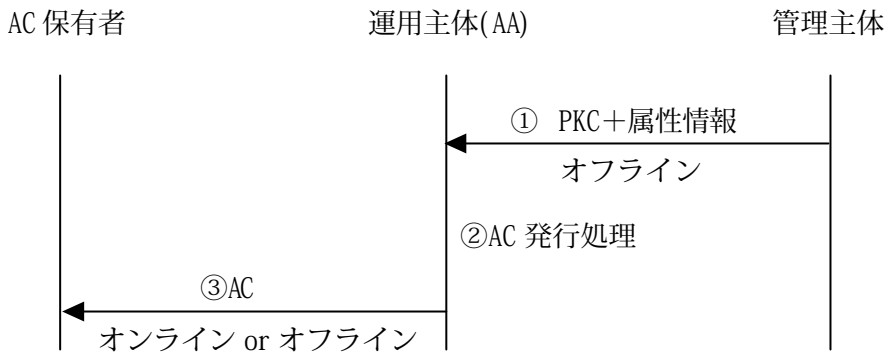


図 3-4 管理主体の申請に基づく発行

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信する。
② AA は、AC 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。

3.2.2 属性証明書の利用手順

本節では､属性証明書の発行を受けた AC 保有者が属性証明書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｡属性証明書の利用方法には 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属性証明書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。

以降で、これら 2 つのモデルについて説明し、各モデルにおける留意事項を述べる。

(1) push 型モデルによる属性証明書の利用手順

ここでは、push 型による属性証明書の利用手順について説明する。

push 型とは、AC 保有者が、AC 検証者に対して属性証明書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、属性認証局(AA)によって発行された属性証明書(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。

push 型モデルにおける属性証明書利用フローを図 3-5 に示す。

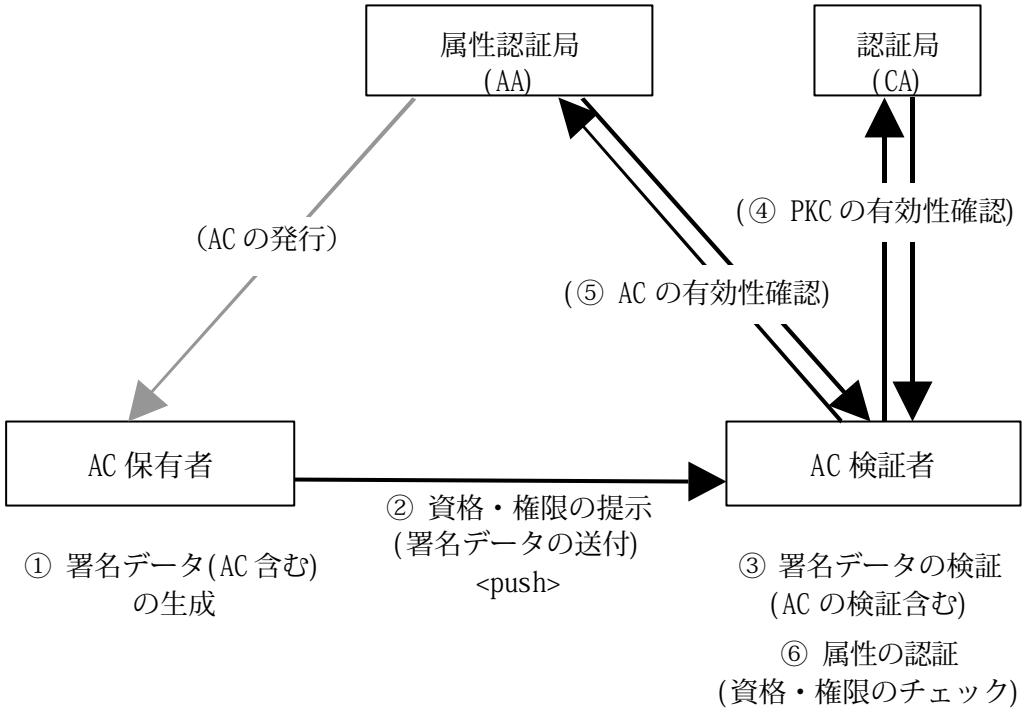


図 3-5 push 型モデルにおける属性証明書利用フロー

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・権限を記載した属性証明書、および、AC 保有者の公開鍵証明書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 保有者の属性証明書の検証も含まれる。属性証明書の検証を行うにあたり、属性証明書が失効するケースがある場合には、属性認証局に対して属性証明書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。また、AC 保有者の公開鍵証明書や属性認証局の公開鍵証明書の検証を行うにあたり、認証局に対して有効性確認を行う。

④ AC 検証者は、属性証明書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、当該属性証明書を提示した 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。

以上のことから、push 型モデルでは、AC 保有者側に、属性証明書の管理を行う機能と属性証明書を伴う署名データを生成する機能が必要となる。

(2) pull 型モデルによる属性証明書の利用手順

ここでは、pull 型による属性証明書の利用手順について説明する。

pull 型とは、AC 検証者が、資格・権限の確認に必要な属性証明書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、属性認証局(AA)によって発行された属性証明書(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。

pull 型モデルにおける属性証明書利用フローを図 3-6 に示す。

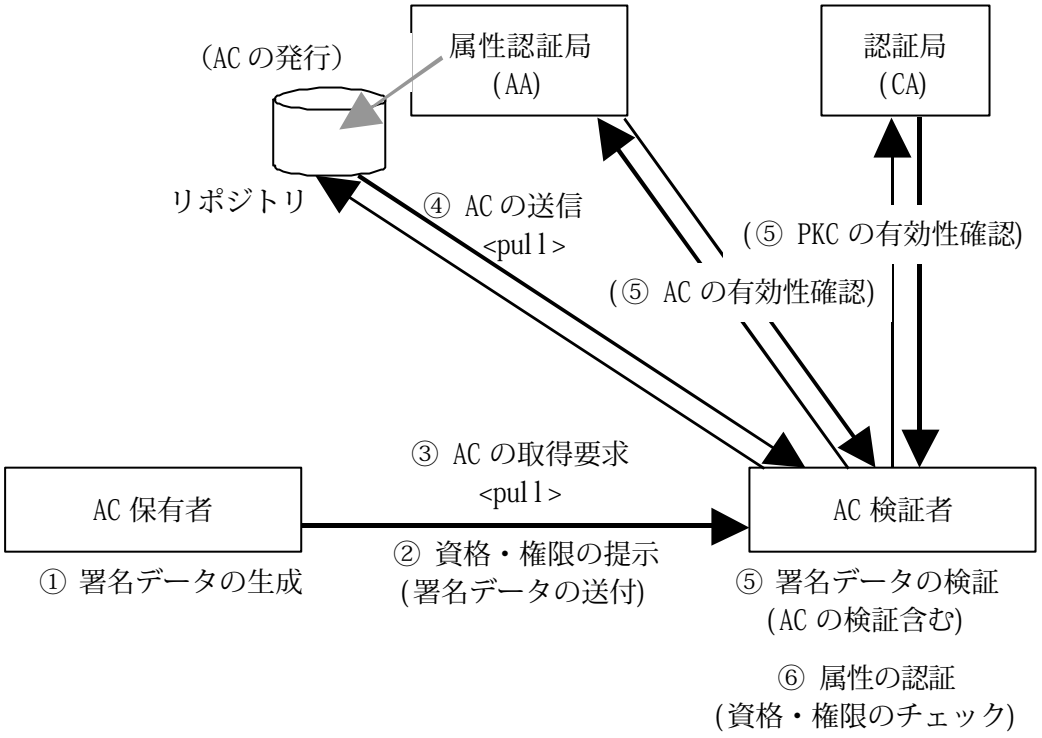


図 3-6 pull 型モデルにおける属性証明書利用フロー

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・権限を記載した属性証明書を保持するリポジトリの所在を示す情報（属性証明書自体は含まない）、および、AC 保有者の公開鍵証明書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に記載されている属性証明書を保持するリポジトリの所在を示す情報を基に、属性証明書の取得要求をリポジトリに対して送信する。

④ リポジトリは、③によって AC 検証者から受信した属性証明書取得要求を基に、対応する属性

証明書を AC 検証者に送信する。

⑤ AC 検証者は、AC 保有者から受信した署名データと、④で取得した属性証明書を検証する。属性証明書の検証を行うにあたり、属性証明書が失効するケースがある場合には、属性認証局に対して属性証明書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。また、AC 保有者の公開鍵証明書や属性認証局の公開鍵証明書の検証を行うにあたり、認証局に対して有効性確認を行う。

⑥ AC 検証者は、属性証明書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、当該属性証明書を提示した 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。

以上のことから、pull 型モデルでは、AC 検証者側に、属性証明書の取得を行う機能が必要となるが、AC 保有者側では、特に属性証明書を扱う必要はないので、既存の PKI 製品のみで対応することも可能となる。

(3) push 型および pull 型利用における留意事項

前記で述べた push 型と pull 型について、それぞれの留意事項を表 3-1 に列挙する。

表 3-1 属性証明書の push 型および pull 型利用における留意事項

	push 型	pull 型
属性情報の保護	AC 保有者の意志に基づいて利用可能 （提供後は二次利用される可能性あり）	AC 保有者の意思に関係なく利用可能 （LDAP サーバからの取得等）
属性証明書検証処理	AC 保有者から提供される	AC 開示場所から取得を要する
属性証明書管理	AC 保有者が媒体（FD、IC カード、etc）を利用して管理する	AA が LDAP サーバ等によって管理する
処理負荷	AC が付与されるため、通信量が大きい	必要に応じて AC を取得するため、AC 検証者の手順が増える

3.2.3 属性証明書の失効手順

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、その属性認証局が発行した属性証明書の失効を要することも想定される。また、属性証明書は、その失効理由、属性証明書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し、有効期間の短いサービスチケットなどはその一例である。

属性証明書の失効可否等詳細に関しては 4.4 節で述べるとし、本節では、属性証明書の利用不

可が生じる場合の整理と共に、それぞれの失効手順について検討を示す。

AC 保有者の属性証明書の利用不可が生じる状況としては、以下の場合が考えられる。

(a) AC 保有者が、自身の判断に基づいて属性証明書を失効させる場合
（要因：AC 保有者の秘密鍵の危殆化、属性の変更、利用停止など）

(b) 属性の管理主体（≠属性認証局）が、その権限に基づいて属性証明書を失効させる場合
（要因：AC 保有者の属性の変更、管理主体の属性管理停止など）

(c) AA が、AA の公開鍵証明書の失効に連動して、属性証明書を失効させる場合
（要因：AA の秘密鍵の危殆化、属性の変更、運用停止など）

(d) CA が、CA、AA、AC 保有者の公開鍵証明書の失効に連動して、属性証明書を失効させる場合
（要因：CA の秘密鍵の危殆化など）

このような属性証明書が利用できない状況に対し、必要に応じて属性証明書を失効させるべきである。上記のそれぞれの状況に応じて、失効の手順を以下に示す。

(a) AC 保有者が、自身の判断に基づいて属性証明書を失効させる場合

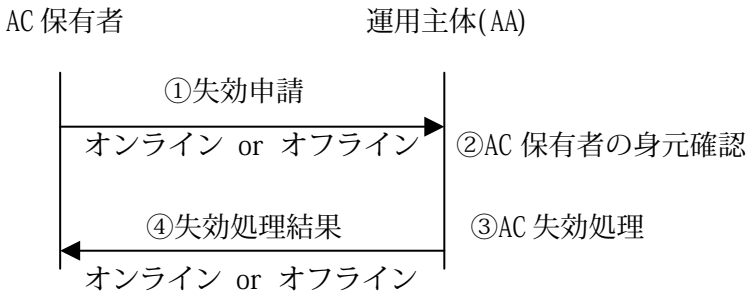


図 3-7 AC 保有者の判断に基づく失効

① AC 保有者は、AA に対して、自身の AC に関する失効申請を行う
② AC 保有者から AC 失効申請を受け取った AA は、AC 保有者の本人確認を行う
③ AA は、②が OK の場合、申請された AC の失効処理を行う
④ AA は、③が OK の場合、結果を AC 保有者に返却する

(注) AC 保有者の本人確認手段としては、オンラインは PKC にて、オフラインは対面確認にて実施することになると考慮される。

(b) 属性の管理主体が、その権限に基づいて AC 保有者の属性証明書を失効させる場合

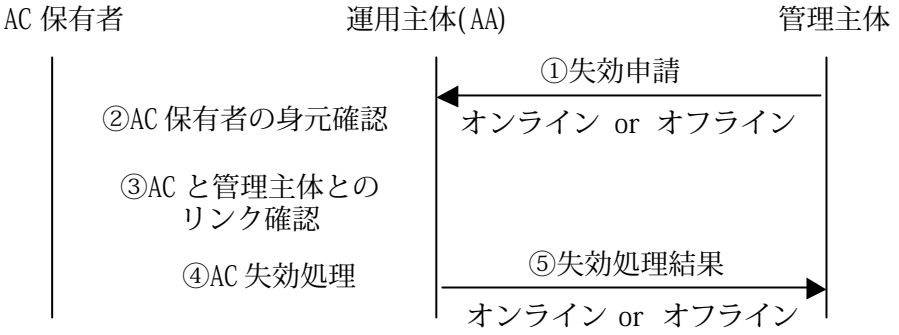


図 3-8 属性の管理主体に基づく失効

① 管理主体は、AA に対して失効対象の AC に関する失効申請を行う
② AA は、管理主体の本人確認を行い、失効対象の AC に関する失効申請を受け取る
③ AA は、②が OK の場合、AC と管理主体との結びつき（リンク）を確認する
④ AA は、③が OK の場合、申請された AC の失効処理を行う
⑤ AA は、④が OK の場合、結果を管理主体に返却する

(注) AC と属性情報の管理主体とのリンクは、属性認証局か管理主体のどちらかで、属性情報と AC の関連付け情報を管理していると考慮される。
(注) 管理主体と属性認証局との間は、専用線、VPN 等セキュアな通信路を介したデータの連携（PKI 以外の運用）も考慮される。

(c) AA が、AA の公開鍵証明書の失効に連動して、属性証明書を失効させる場合

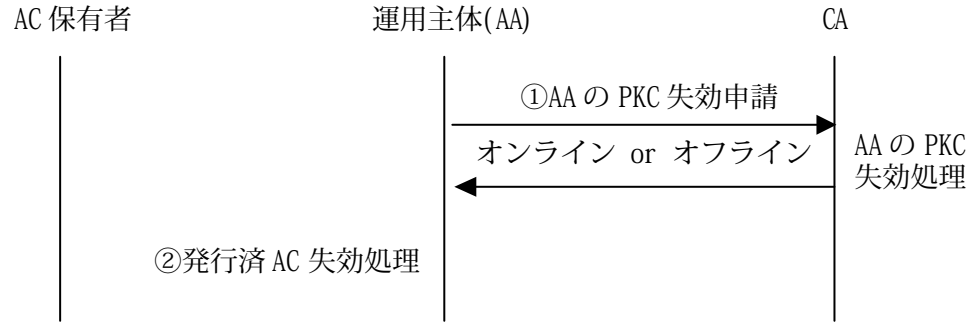


図 3-9 属性認証局に基づく失効

① AA は、CA に対して危殆化等に伴い自身の PKC の失効申請を行う
② AA は、①が OK の場合、自身が発行した AC の失効処理を行う

(d) CA が、CA、AA、AC 保有者の公開鍵証明書の失効に連動して、属性証明書を失効させる場合

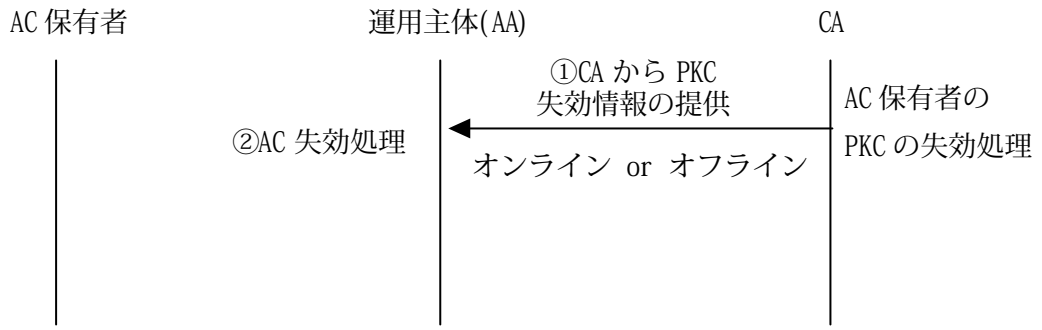


図 3-10 認証局に基づく失効（AC 保有者の PKC 失効に連動する場合）

（例：AC 保有者の PKC 失効に連動する場合）

① AC 保有者の PKC 失効に伴い、CA から AC 保有者の PKC 失効情報が提供される

② AA は、①が OK の場合、自身が発行した AC の失効処理を行う

このように失効の手順としてモデルを提示しているが、(c)や(d)に示されるように、正当性の根幹となる PKC が失効された際には、AC の正当性を確認することができないため、失効の必要性の有無も要検討である。

そのため、この議論を踏まえて、失効の必要性については、4.5 節で検討を行うこととする。

4. 属性認証の特徴と利用上の留意事項

前章では、属性の認証に属性証明書を利用する技法について、その当事者と運用手順概略を示した。本章では、運用手順をより詳細に記述し、ソフトウェアで実装すべき仕様も整理している。また、当事者の利用上の留意事項、システム構築における留意事項、およびソフトウェア実装上の留意事項もその中で提示する。

4.1 属性認証局の信頼の根拠とポリシー

電子商取引や電子申請、あるいは社内業務の電子化等において属性証明書(AC)を利用する場合、その AC が信頼できることが必要である。本節では、AC の信頼性と、その裏づけとなる属性認証局(AA)の信頼性について整理する。

4.1.1 信頼される属性認証局とは

AC を発行する主体として AA は一定の信頼性が求められる。では、AA の信頼性とは何だろうか。まず、運用を含めた「情報システムとしての信頼性」が考えられる。そこではシステムの可用性や安全性といった事項が想起される。異なる観点から「社会的信用」もある。組織としての信用力や、社会認知度等である。このように AA の信頼性には異なるいくつかの観点がある。また、AC を利用する業務により信頼性要求の程度も異なることに留意する必要がある。属性の例を表 4-1 に示す。

表 4-1 属性の例

システム例	属性内容 例
社内システム	社長、部長、課長、等
医療システム	医師、薬剤師、看護士、等
代理人システム	弁護士、税理士、行政書士、等
Web アクセス制御システム	アクセス範囲、等

代理人システムでは、適用業務や属性自体を考えると、社会インフラとして高度な信頼性が AA に求められる。これに対して、Web アクセス制御システムでは、そのシステムで扱う情報の重要性に応じて、経済性とのトレード・オフにも考慮して、AA の実現においてさまざまな信頼性があってよい。さらに、これらの例から、許認可あるいは免許発行等の権限を持つ属性管理主体と、AA の運用主体は、組織として一般には同一ではないことも明らかである。信頼できる AA を実現するには、属性管理主体と AA の運用主体の関係を明確にし、その原則に基づき役割分担と処理手順を定める必要がある。この観点で整理した処理手順例を 3 章で示した。また、属性情報にはプライバシー情報や機密情報も含まれるので、管理主体と AA をとおして属性情報が適切に管理されることも信頼の要件である。

以上は一般的な概説であったが、次に、属性認証に関する標準化で特に意識される「信頼」について少し考察してみる。

「一般的に言えば、あるエンティティ(A)が別のエンティティ(B)を『信頼する』とは、『Bは(Aの)期待通りに行動する』とAが想定できることである。」

ITU-T勧告X.509[X.509, Section 3.3.54]で『信頼(する)』という用語は、上記のように定義されている。この定義は、X.509の属性証明書フレームワークにも適用され、RFC3281もこの枠組みに基づいている。X.509の「信頼」の定義では、上記に続けてさらに次のように述べられている。

「この信頼は、それぞれの目的ごとに適用されるものである。この(X.509の)枠組みでは、信頼の主な役割は､認証されたエンティティと認証局の間の関係を表現することである｡すなわち、エンティティが「その認証局は、有効で信用のできる証明書のみを発行すると信じられる」ことを確信している関係を表現する｡」

このような認証局が発行した証明書が「信頼できる証明書」であり、これは公開鍵認証局(CA)だけでなく、属性認証局(AA)の場合も全く同様である｡「有効で信用のできる属性証明書のみを発行する」と利用者が想定できる属性認証局が、すなわち「信頼される認証局」となる。

本報告書では、前章でもAC保有者とAC検証者を区別して記述している。この点を加えると、X.509におけるこれらの記述の意味がより明確になる。AAは、AC保有者となる申請者にACを発行する。このACの信頼性は、AC検証者に必要なものである。そして、ACの信頼性の根拠は、AAの情報システムとしての信頼性や、申請審査プロセスの信頼性に求められる。AAの信頼性は、標準化が進んでいるCAの信頼性と同様に考えられる部分が多いので、次項ではCAの信頼の根拠となる標準も援用しながら、AAの信頼の根拠について解説する。

4.1.2 信頼の根拠

属性認証局の運用は、基本的には、それが利用する公開鍵基盤とは独立に行うことが可能である。しかし、証明書の検証において、その識別認証部分を公開鍵基盤に依存するため、公開鍵に対する信頼を属性証明書の利用の場合も前提とする必要がある。CAの信頼の根拠となるものとして、証明書ポリシー(CP)や認証局実施規定(CPS)がある。それらの信頼の上で、AAも属性証明書ポリシー(ACP)や属性認証局実施規定(ACPS)を定め、自身の発行するACの属性の記述ならびにその認証に関する義務・責任やその適用範囲等を利用者に対して明らかにして、利用者との間の信頼関係の根拠としなければならない。

属性証明書ポリシーや属性認証局実施規定で述べるべき項目は、CAの場合とほぼ同様である。よって、RFC2527のCP/CPSフレームワークが有効だろう。ただし、AAは、利用者識別に利用する名前に関する規定や鍵のセキュリティに関する要件などは自身の利用する認証基盤のポリシーに従うため、RFC2527で定めるCPSの章立てには検討不要な項も含まれる。しかしCAのCPSとの比較検討の要も考慮すれば、RFC2527の指針にできるかぎり沿った形でACPやACPSを記述することがやはり望ましい。ACP/ACPSで述べられる項としては、例えば、次のようなものがあげられる。

a) 属性の説明、法的根拠

b) 属性の識別・表現方法
c) 属性の適用範囲
d) その属性を公開するか制限するかの判断
e) 属性の提供方法 (PKC or AC)
f) AC の有効期間
g) AC 失効のサポートの有無。失効を行うのであれば、その条件とルール。
h) 属性を委譲できるかどうか。

AC 検証者は AC/ACPS のこれらの項の記述を参考に信頼を判断することになる。また、属性の種類や性質によっては、将来的に特定認証局の認定制度と同様、AC/ACPS に基づき正しくシステム運用がなされているか信頼できる審査が行われた上で公的な認定が行われることになるかもしれない。

4.1.3 属性ポリシーとその制御手段

認証基盤としての認証局を前提として属性証明書を発行する属性認証局は、識別認証に関しては、その認証局のポリシーに従うことになる。よって、属性認証局のポリシーは、属性に関わる部分に限定される。

4.1.3.1 属性のトラストアンカー

なんらかの属性を特定のユーザーに割り当てることに関して、SOA(Source of Authority)と呼ばれる機関が、究極の責任を持つエンティティとして、リソースを持つ権限検証者(privilege verifier)の信頼の対象となる。つまり SOA はその属性に関するトラストアンカーにあたる存在となる。SOA は識別情報すなわち「自分が何者であるか」ということに関しては自身で証明する必要はないが、「特定の属性を付与する権限をもつ」ということに関しては、あらかじめ他のエンティティから信頼してもらう必要がある。X.509 では、次節で述べるような CA が SOA に発行する PKC に SOA であることを示すオプションの拡張も用意している。全ての AC はその信頼のつながりを SOA まで辿ることができなければならない。RFC3281 ではそのつながりの検証の複雑さを避けるため、ある特定の属性に関してはただ 1 つの属性認証局のみを信頼される SOA として、その属性に関する全ての AC 発行を行うことで複雑さを増すことなく十分な目的を達成できるとしている。すなわち、一つの AA はある属性の SOA であるということである。これは言い換えると、属性ごとに複数の AA(すなわち SOA)が存在して構わないという考えである。このような複数の SOA の PKC を制御するための拡張として次のようなオプションが存在する。

4.1.3.2 PKC による AA の制御

CA が属性認証局の公開鍵証明書に AAControls エクステンションを含めることによってその AA の SOA としての信頼の指標を示す方法が RFC3281 では用意されている。この AAControls では、AA が AC に含めることのできる(あるいは、逆に含めることのできない)属性を指定することができるようになっている。

一方、X.509 でも類似のものとして、sOAIdetifier エクステンションにより SOA の PKC としての判断材料を提供できる。しかし、細かな属性の記述は SOA に任されており、CA は行わない。

CA が PKC により AA を強力に制御する必要がある場合には、上記のようなエクステンションと

それを理解するアプリケーションを利用することができるだろう。

4.1.3.3　証明書ポリシー

証明書自体にそのポリシー(すなわち証明書ポリシー)を記述するやり方もある。X.509 における証明書ポリシーの定義は以下のようなものである。

「共通のセキュリティ要求をもった特定のコミュニティやアプリケーションのクラスに対して、証明書の適用可能性を指定する規則の名前付けされた集合」

CertificatePolicies エクステンションは､PKC の証明書ポリシーを規定するために用意されたものである。しかし、AC に関しては、現状では上記のような規定がない。そこで、Attribute Certficate Policies エクステンションを提案するインターネットドラフトが IETF で提出され、議論されている。

4.1.3.4　権限ポリシー

この他に、これは属性に関連づけられるポリシー(権限ポリシー)がある。X.509 勧告における権限ポリシーの定義は以下のようなものである。

「権限検証者が、権限主張者に対して機密のサービスを提供したり、あるいはそれを利用する権限を与えたりする場合の条件のあらましを述べたもの。権限ポリシーは、サービスに関連する属性や権限主張者に関連する属性と関係づけられる。」

しかし、これに関する構文定義は X.509 では標準化はされずに、付録にいくつか例が挙げられるに止まっている。

また､X.509 勧告では AttributeDescriptor エクステンションを用いて､SOA が権限属性の定義とそれに関する規則を権限検証者に伝えるための仕組みの 1 つとして用意されている。SOA はこの拡張を含む属性記述証明書(attribute descriptor certificate)を自己署名証明書(発行者と holder が一致している証明書)として発行する。

X.509 勧告の AC のエクステンションには、他にもポリシーに関わるものとして以下が用意されている。

AcceptablePrivilegePolicies エクステンション

AcceptableCertificatePolicies エクステンション

このように X.509 勧告では SOA が属性の記述・表現方法やその適用条件・権限ポリシーを定める形になっている。RFC3281 でも、属性として X.509 勧告で規定されている属性に加えて、いくつかの IETF 独自の属性を用意している。そこでは､IETF 属性構文(IetfAttrSyntax)を定義して、ポリシー管理局(Policy Authority)を属性認証局とは別に指定することができる。これは、SOA とポリシー認証局を分けることができるようにという理由からである。これにより、一つのポリシー認証局(例えばある企業)が複数の(例えば社内)AA の発行する AC の属性を一元的に制御することが可能となる。

これらのエクステンションは、属性認証局の信頼を支えるポリシーを利用者に伝え、それを細かく制御するための手段として提供されるものである。しかしながら、これらはまだ一般的とは言い難く、また属性証明書の証明書ポリシー エクステンションのように標準化に向けて現在､議論中のものもある。ただし、これらはあくまでも属性認証局を運用する側におけるポリシー制御

の手段であり、属性証明書の利用者の側から見たときには、そのような信頼の管理面の詳細よりも、先に述べたような、その裏付けとなる「システムとしての信頼性」および「社会的位置付けとしての信頼性」の確保と維持を正しく図ることが重要であることは明らかである。

4.2 属性証明書の発行

本節では、属性証明書の発行時において検討が必要と思われる事項として、

- AC の発行手順
- AC と PKC とのリンク方法
- 属性情報の保護方法

について述べる。

4.2.1 属性証明書の発行手順

4.2.1.1 審査

属性認証局は、AC を発行する前に、発行対象者が当該 AC を発行されるべき正当な人物であるかどうかを十分に確認する必要がある。その正当性を判断するために属性認証局が確認すべき事項として、以下が考えられる。

(a) 発行対象者の本人特定
AC を発行される対象者の PKC の正当性と、その PKC に対応する秘密鍵による署名の正当性を検証する。

(b) 発行対象者の属性保持
AC を発行される対象者は、本当に付与される属性（資格）を保持しているかどうかを確認する。

ここで、発行申請者は、発行対象者とは異なる代理人である場合もある。さらに、属性認証局が一方的に AC を発行する場合においては、属性認証局自身が発行申請者となる。

審査をどの程度厳密に行うべきかという問題は、AC を利用するアプリケーションに依存すると思われる。AC を利用するアプリケーションは多岐にわたると考えられるため、目的や用途によって適切な審査基準を設けることが望ましいと考えられる。

4.2.1.2 オンライン発行の場合

4.2.1.1 節で示した審査に基づき、インターネットや属性認証局の窓口端末を介して AC のオンライン発行を行う例として、社内イントラネットの Web アクセス制御を属性証明書により実現するシステム（社内イントラネットシステム）を考える。このシステムでは、社員の所属組織や役職などの属性の管理主体は人事部であり、運用主体は各事業組織である。社員の PKC を発行する CA は人事部に一つ設置され、AC を発行する AA は各事業組織に設置される。

本システムが提供する業務サービスを考慮すると、AC の発行形態には少なくとも以下の 2 つの場合が考えられる。

(a) 社員の申請に基づき AC を発行する場合
(b) 人事部の申請に基づき AC を発行する場合

以下に、各方式における AC 発行手順の一例を示す。

(a) 社員の申請に基づき属性証明書を発行する場合

【発行手順例】

① 社員は、自分の所属する事業組織内の AA に対し、自分の PKC と、その PKC に対応する秘密鍵による署名が付与された申請書を送信する。

② AA は、PKC および申請に対する署名の有効性を検証する。

③ AA は、②が OK の場合、人事部に社員の属性を問い合わせ、申請された属性と同じであるかどうかを確認する。

④ 人事部は、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、社員に AC を送信する。

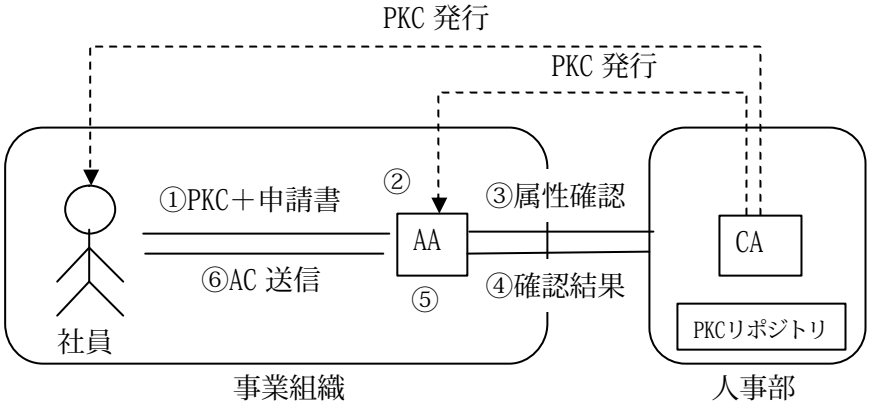


図 4-1 社員の申請に基づき AC を発行する場合

(b) 人事部の申請に基づき属性証明書を発行する場合

【発行手順例】

① 人事部は、事業組織内の AA に対し、AC 発行対象者の PKC および属性情報を送信する。

② AA は、社員の PKC とリンクした AC を発行する。

③ AA は、状況に応じて、AC を社員に送信する。

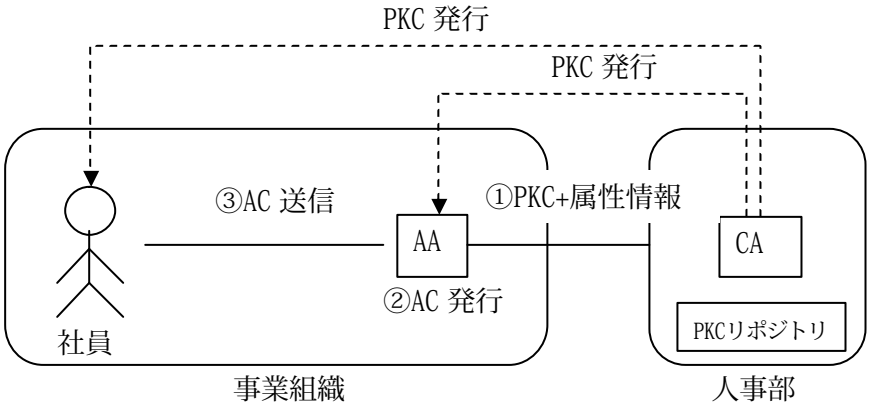


図 4-2 人事部の申請に基づき AC を発行する場合

4.2.2 AC と PKC とのリンク方法

RFC3281 に基づくモデルでは、AC は、AC 保有者の保有する PKC とリンクされるべきであるとされている。そのための手段として、AC プロファイルにおける holder 領域には、以下の 3 つのうち少なくとも 1 つが記述されるべきとされ、baseCertificateID の使用が推奨(SHOULD)されている。

- baseCertificateID (PKC の発行者とシリアルナンバー)
- entityName (AC 保有者の名称)
- objectDigestInfo (PKC、または公開鍵、または実行オブジェクトのダイジェスト値)

基本的には、RFC3281 の推奨に従うことが推奨されるが、ある特殊な用途においてはその限りではない。例えば、医師など永続性のある資格に対しては AC 保有者の PKC の有効期限を超える有効期限を持つ AC が発行される可能性があるが、AC の holder に entityName を採用していれば、PKC の有効期限が近付いてきたとき、AC のリンクを更新された PKC へ付け替えるという措置により AC の再発行を不要にできることが考えられる。以上をまとめて表 4-2 に示す。

表 4-2 AC の holder 領域で利用できるオプション

	RFC3281 推奨	PKC との リンク結合度	異なる PKC への 付け替え
baseCertificateID	○	○	×
entityName		○	○ (注 2)
objectDigestInfo		△ (注 1)	△ (注 3)

(注1) 公開鍵のハッシュ値であれば○、その他であれば×

(注2) 同じ所有者(subject)名を持つ PKC 間で付け替え可

(注3) 公開鍵のハッシュ値であれば、同じ所有者公開鍵情報(subjectPublicKeyInfo)を持つ PKC 間で付け替え可

4.2.3 属性情報の保護方法

AC を利用するサービスによっては、AC の属性情報に個人のプライバシに関する情報が含まれることがあるため、AC の取り扱いに注意が必要となる場合がある。例えば、以下の脅威が考えられる。

[AC 検証者の不正]

- AC 検証者が AC 保有者から提示された属性情報を二次利用することにより、属性情報が漏洩する。

[第三者の不正]

- 不正な第三者が通信路を盗聴し、属性情報が漏洩する。
- 不正な第三者が AC のリポジトリを攻撃し、属性情報が漏洩する。

AC 検証者の不正を防止するためには、まず AC 保有者自身が、信頼できない AC 検証者に対して不用意に AC を配布すべきではないことを、属性認証局の CPS 等により規定することが考えられる。その上で、アプリケーションの運営規定により AC 検証者の不正に対して罰則を設けるなどの運用的な対処が考えられる。

本節では、主に第三者の不正から属性情報を保護（秘匿）するという観点で、その技術的な対策について述べる。

4.2.3.1　通信路上の暗号化

属性証明書の発行時に限らず、AC のデータはネットワーク上を介して流通されるため、不正者による通信路の盗聴を考慮する必要がある。対策として、最も一般的な方法は SSL/TLS による通信路上の暗号化である。しかし、これは暗号化される範囲が通信路に限定されるため、盗聴により不正にデータを入手しようとする者に対しては効果があるが、例えば pull 型モデルにおける AC のリポジトリを攻撃される場合を考慮すると不十分である。この対策として、次節以降に示す属性情報の暗号化という対処法が考えられる。

4.2.3.2　RFC3281 に基づく属性情報の暗号化

通信路上での盗聴対策のみであれば SSL などによる方法で実現できるが、暗号化される範囲が通信路上に限定されるという欠点がある。これに対し、AC の属性情報そのものを暗号化することによりプライバシを保護する方式が RFC3281 で規定されている。

RFC3281 では､AC の属性の一つである encAttrs 属性に EnvelopedData 型の暗号化された属性を複数格納する方法が用意されている。各値は、あらかじめ設定された複数の AC 検証者に対する暗号化された属性であり、当該 AC 検証者のみが正しく復号できる。

以下に、属性を暗号化する際に属性認証局が行う手順を示す。

【属性情報を暗号化した属性証明書の発行手順 1】

① AC 検証者グループごとに、暗号化する属性のセットを確認する

② 暗号化する各属性セットにつき、以下の処理を行う：

a) この AC 検証者グループに対する EnvelopedData 構造を作成する

b) encAttrs 属性の値として EnvelopedData が格納されている ContentInfo をエンコードする

c) 署名する AC に cleartext 属性がないことを確認する

③ AC に encAttrs を（その複数値とともに）追加する

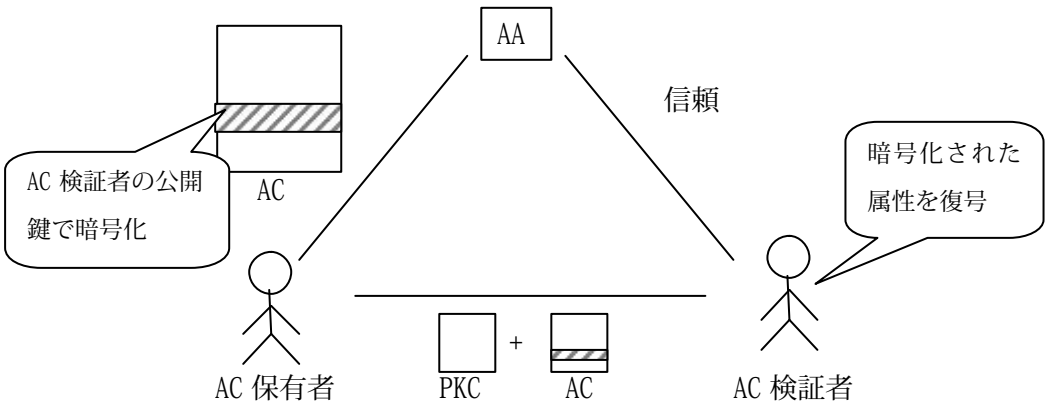


図 4-3 RFC3281 に基づく属性情報の暗号化

4.2.3.3　属性のハッシュ値による方式

RFC3281 に基づく方式では、AC 検証者が暗号化された属性情報を復号できなければならないことから、AC 検証者の公開鍵で属性を暗号化する必要があるため、特定の AC 検証者にしか利用できないことが前提となっている。

この問題を解決し、任意の検証者に対して属性が暗号化された AC を利用できる方式として、属性のハッシュ値を利用する方式が考えられる。以下に、その手順を示す。

【属性情報を暗号化した属性証明書の発行手順 2】

① AC 発行対象者は、属性 m に対する AC の発行を AA に申請する。
② AA は、AC 発行対象者に、属性領域が属性 m のハッシュ値 H(m) である AC を発行する。
③ AC 発行対象者は、乱数 k を生成し、それを共通鍵とする共通鍵暗号で属性を暗号化した SE(m) と、AC 検証者の公開鍵で共通鍵を暗号化した E(k) と、PKC と、AC を AC 検証者に送信する。
④ AC 検証者は、自身の秘密鍵で E(k) を復号し k を得て、k で SE(m) を復号し属性 m を得て、そのハッシュ値 H(m) を求め、AC の属性領域の値と比較する。両者が同じであればその属性を認め、異なれば申請を棄却する。

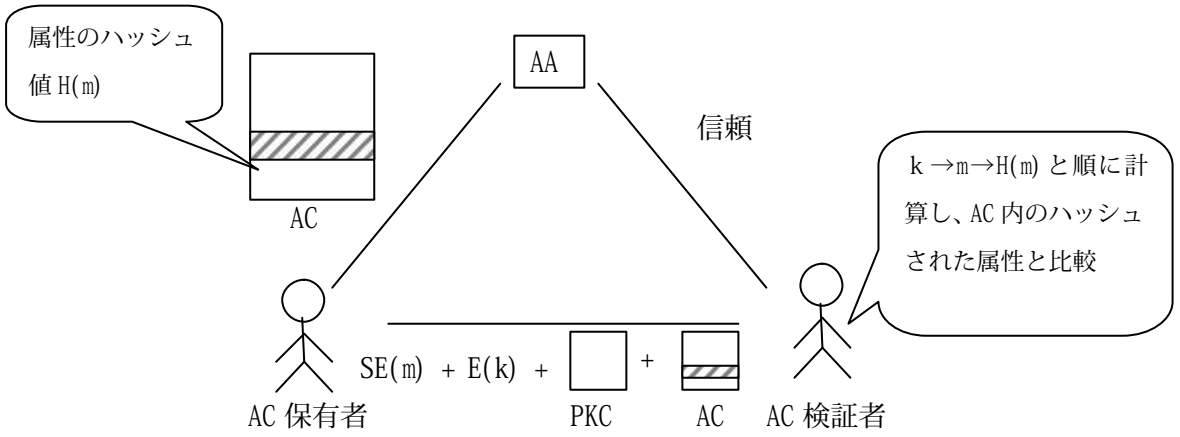


図 4-4 属性のハッシュ値を利用する方式

4.3 属性証明書の利用

本節では、既存の PKI 製品をベースにして AC を利用する場合に、必要となる機能について検討を行う。

4.3.1 想定環境

図 4-5 は、AC 保有者システムの概観の一例を示す図である。図 4-5 において、アプリケーションとは、後述の属性認証プログラムを利用して、あるシステムに対して資格や権限を提示するためのアプリケーションプログラム（例えば、アクセス制限された Web サービスクライアントプログラム、電子申請クライアントプログラム等）である。また、AC 利用プログラムとは、AA から発行された AC を管理し、AC 検証者に対して属性情報の提示を行うために必要な各種処理を実現するプログラムである。PKI プログラムは、署名データの生成・検証や PKC の解析・検証を行うためにプログラムであり、既存の PKI 製品や暗号ライブラリで実装されているものである。

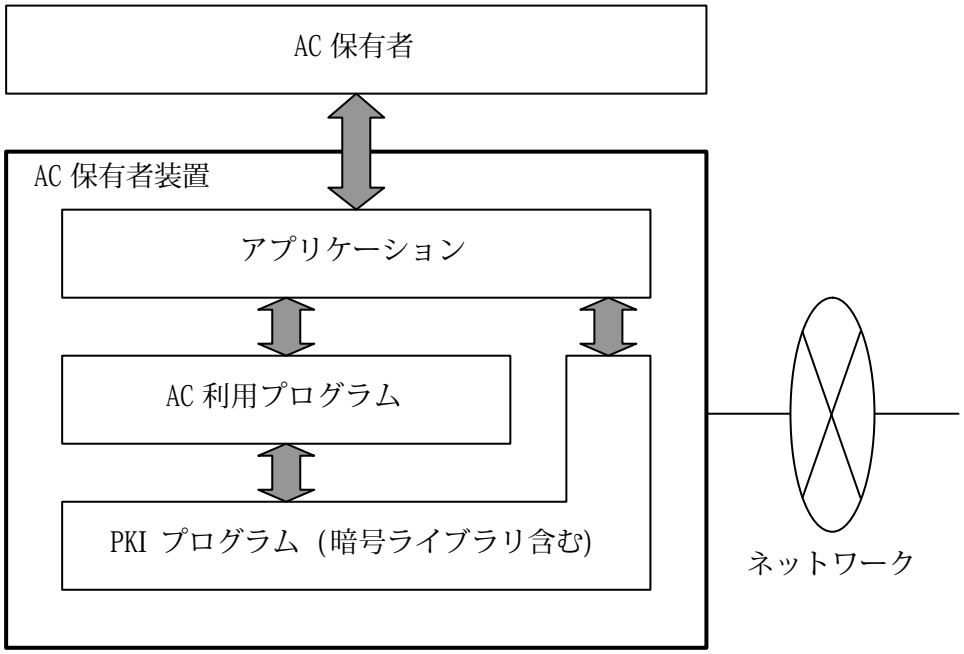


図 4-5 AC 保有者システムのモデルの概観

既存の PKI プログラムでは、具体的に、以下に列挙するような機能を実装しているものと想定する。ここで列挙した機能が、既存の PKI プログラムで実装されていない場合には、属性証明書利用プログラムで実装することとなる。

- 署名データ(RFC3369 CMS v4 SignedData 等)の生成

4.3.2 各プログラムにおける機能要件

AC 利用プログラムに必要とされる機能について、push 型と pull 型に分けて検討を行った。各

プログラムにおいて必要と想定される機能は表 4-3 および表 4-4 のようになる。但し、本検討結果は、あくまでも実装の一例を示したものであり、各プログラムの仕様を限定するものではない。

「手順」の欄は、AC の利用の際の順序を示している。「機能」および「機能概要」の欄は、AC を利用するために必要な機能とその内容を順序ごとに説明している。「利用者による操作」、「アプリケーション」、「AC 利用プログラム」、「ＰＫＩプログラム」の欄は、それぞれの機能がどのプログラムで実装されることになるかを示すものである。その機能が実装されると想定した欄には○を、場合によって実装される欄には△を記入している。

表 4-3 AC 保有者側に必要な機能一覧(push 型の場合)

手順	機能	機能概要	利用者による操作	アプリケーション	AC利用プログラム	PKIプログラム
①	署名対象データ生成	署名対象となるデータを生成する。	○	○		
②	秘密鍵（PKC）の特定	署名対象データに署名するための秘密鍵(署名検証に使用する PKC）を特定する。 特に、AC 保有者が秘密鍵を複数所有している場合には、どの秘密鍵を使用するかを AC 保有者に選択させる。AC 保有者が、秘密鍵の管理にＩＣカード等の秘密鍵格納媒体を利用している場合には、秘密鍵格納媒体を装置にセットし、PIN 入力等によりその中に格納されている秘密鍵を使用可能な状態にする。 また、AC 保有者に対して、署名行為に関する意思確認を行う。	○	○		○
③	AC の特定	AC 検証者の要求する AC を特定する。特に、AC 保有者が AC を複数所有している場合には、どの AC を AC 検証者に提示するかを AC 保有者に選択させることも想定される。	△	△	○	
④	署名データ生成	本表①にて生成した署名対象データと、②にて特定した秘密鍵および PKC と、③にて特定した AC をもとに、署名データを生成する。 既存の PKI 製品において、RFC3369 CMS v4 Si gnedDat a のような AC を含む署名データを取り扱うことができない場合には、AC 利用プログラムで本機能を実装する必要がある。			△	○

表 4-4 AC 保有者側に必要な機能一覧(pull 型の場合)

手順	機能	機能概要	利用者による操作	アプリケーション	AC利用プログラム	PKIプログラム
①	署名対象データ生成	署名対象となるデータを生成する。 AC 検証者が適切な AC の取得先を知らない場合には、AC のポインタを署名対象データ内に含める必要がある。	○	○		
②	秘密鍵（PKC）の特定	署名対象データに署名するための秘密鍵(署名検証に使用する PKC）を特定する。 特に、AC 保有者が秘密鍵を複数所有している場合には、どの秘密鍵を使用するかを AC 保有者に選択させる。AC 保有者が、秘密鍵の管理にＩＣカード等の秘密鍵格納媒体を利用している場合には、秘密鍵格納媒体を装置にセットし、PIN 入力等によりその中に格納されている秘密鍵を使用可能な状態にする。 また、AC 保有者に対して、署名行為に関する意思確認を行う。	○	○		○
③	署名データ生成	本表①にて生成した署名対象データと、②にて特定した秘密鍵および PKC をもとに、署名データを生成する。				○

4.4 属性の検証

本節では、属性の検証に関して、次の観点から検討を行う。

- 検証手順
  AC、および、属性情報の検証手順について説明する。
- 汎用製品での実装
  既存の PKI 製品をベースにして AC の検証を実現しようとした場合に必要な機能について検討を行う。
- 検証手順の性能および運用性
  検証手順を性能面・運用面から評価する。

4.4.1 検証手順

本項では、AC、および、属性情報の検証手順について説明する。

4.4.1.1 RFC3281 における検証手順

本項では、IETF で提案されている RFC3281(“ An Internet Attribute Certificate Profile for Authorization”)に基づいて、AC の検証手順（必須項目のみ）を説明する。

(1) AC 保有者の正当性確認

AC 保有者によって提示された属性の検証を行うために、AC 検証者は、AC の指示する PKC を用いて、AC 保有者によって施された署名を検証する。続いて、前記で用いた PKC の認証パスを、RFC3280 の証明書検証手順に基づいて検証する。

この処理は、AC を提示した者が、本当に AC 保有者であるかということを確認するためのもの、すなわち、AC 中の holder フィールドに示された PKC の所有者であるかどうかということを確認するためのものである。AC は電子データなので、簡単に複製できてしまう。そのため、AC を提示しただけで確認を完了してしまうと、他人の AC を使用して不正に権限や資格を取得できてしまう。このようななりすましを防止するために、AC 保有者が施した署名の正当性を確認することが必要となる。

(2) AC の正当性確認

AC 検証者は、AA の PKC を使用して、AC の署名を正しく検証できることを確認する。続いて、AA の PKC に関する認証パスを、RFC3280 の証明書検証手順に基づいて検証する。

この処理は、AC の発行者の本人性を確認するとともに、AC に記述された内容が AC の発行者によって生成されたものであることを確認するものである。すなわち、AA のなりすましと AC の改ざんを防止するものである。

(3)　AA の適合性確認

AC 検証者は、AA の PKC が RFC3280 のプロファイルに準拠していることを確認する。また、AA の PKC の keyUsage 拡張において、digitalSignature のビットが 0 になっていないことを確認する。さらに、AA の PKC の basicConstraint 拡張において、cA フィールドが TRUE に設定されていないことを確認する。

これは、存在を証明する役割である CA と権限を付与する役割である AA は、権限を分離すべきという考え方に基づいている。また、仮に、PKC を発行する CA と AC を発行する AA が同一のエンティティ名であり、同一のシリアル番号を持つ PKC と AC を発行してしまった場合、失効情報中に記載されたシリアル番号は PKC と AC のどちらを指し示しているのか、あるいは、IssuerAndSerialNumber で指定した証明書は PKC と AC のどちらを指し示しているのかといったような混乱も引き起こす。

(4)　AA の信頼性確認

AC 検証者は、AA が、AC で取り扱っている属性を付与する者（組織）として、信頼できることを確認する。

手順２においては、AC を含む認証パスを検証することで、AA の本人性の確認と AC の内容に改ざんがないことの確認できるが、AA が、本当にその属性を割り当てるのに適切な存在であるのかどうかといった確認はできていない。本処理は、AA が資格や権限といった属性を割り当てるのにふさわしいエンティティであることを確認するためのものである。従って、PKC（存在証明）の信頼点とは別に、AC（資格証明）の信頼点というものも設定することになる。

(5)　AC の有効期間チェック

AC 検証者は、属性の検証日時が AC の有効期間の範囲内であることを確認する。

本処理は、AC の利用が、AA が意図した AC の有効期間の範囲内であるかどうかを判断するためのものである。

(6)　AC 受入資格の確認

AC に targetingInformation 拡張が含まれている場合、AC 検証者は、自身がターゲットとして指定されているかどうかを確認する。

これは、特定のサーバ／サービスでのみで AC を使用可能とさせるための拡張であり、指定されたサーバ／サービスの中に当該 AC 検証者が含まれていない場合は、AC の受入れを拒否しなければならない。

(7)　クリティカルな拡張に関するチェック

AC の中に、AC を検証するアプリケーションで未サポートのクリティカルな拡張を含んでいた場合、AC 検証者は、AC の受け入れを拒否しなければならない。

4.4.1.2　モデルを用いた場合の検証例（その１）

前項で述べた検証手順を基に、特定の属性認証モデルを用いて、具体的な検証方法を例示する。本項では、3.1 節で記述されている「モデル１」、すなわち、AA と AC 保有者が同一の CA から PKC の発行を受けているというモデルを例に挙げて説明する。

図 4-6 に示すモデルでは、PKC を発行する CA、AC を発行する AA、AC の発行を受けそれを所有する AC 保有者、および、AC 保有者から提示された属性を検証する AC 検証者という４つのエンティティが登場する。

CA は、自身に対して PKC①（自己署名証明書）を発行している。また、AA および AC 保有者に対して PKC②・④を発行している。AA は、AC 保有者に対して AC③を発行している。AC 検証者は、PKC を発行する組織として PKC①を信頼し、AC を発行する組織として AA を信頼しているものとする。

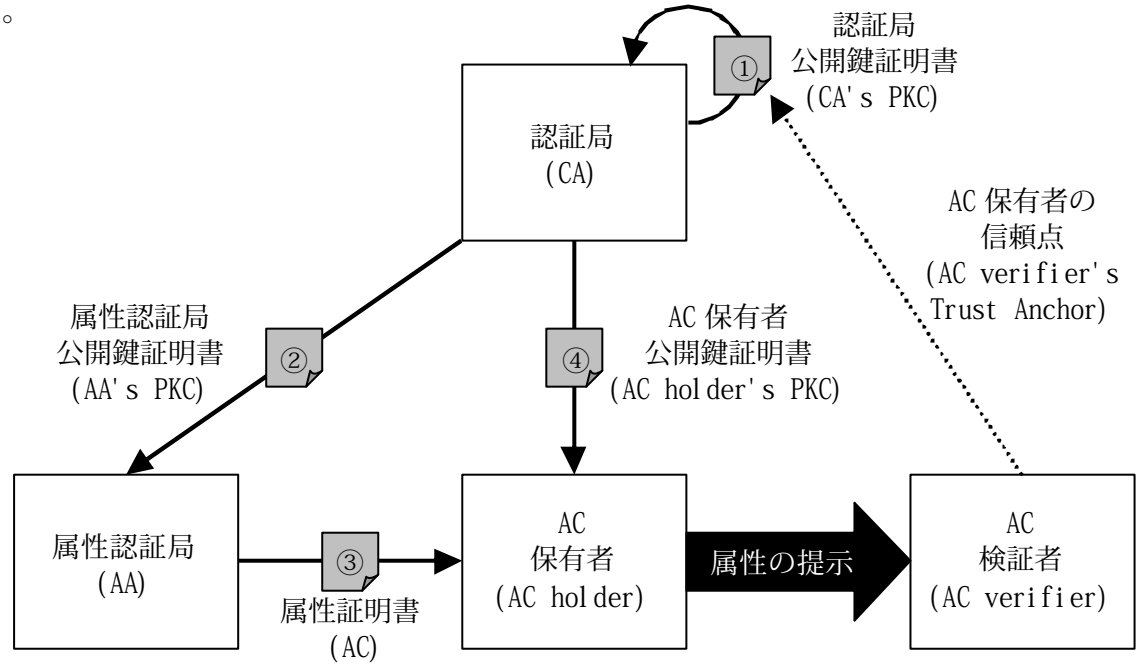


図 4-6 AA と AC 保有者が同一の CA から PKC の発行を受けている場合の属性認証モデル

このような属性認証モデルにおいて、AC 保有者が、AC 検証者に対して属性情報の提示を行った場合（すなわち、図 4-7 の AC③、当該 AC に関わる PKC 群①・②・④等、および、要求メッセージ⑤を AC 保有者から受信した場合）に、AC 検証者が行う AC の検証方法について説明する。

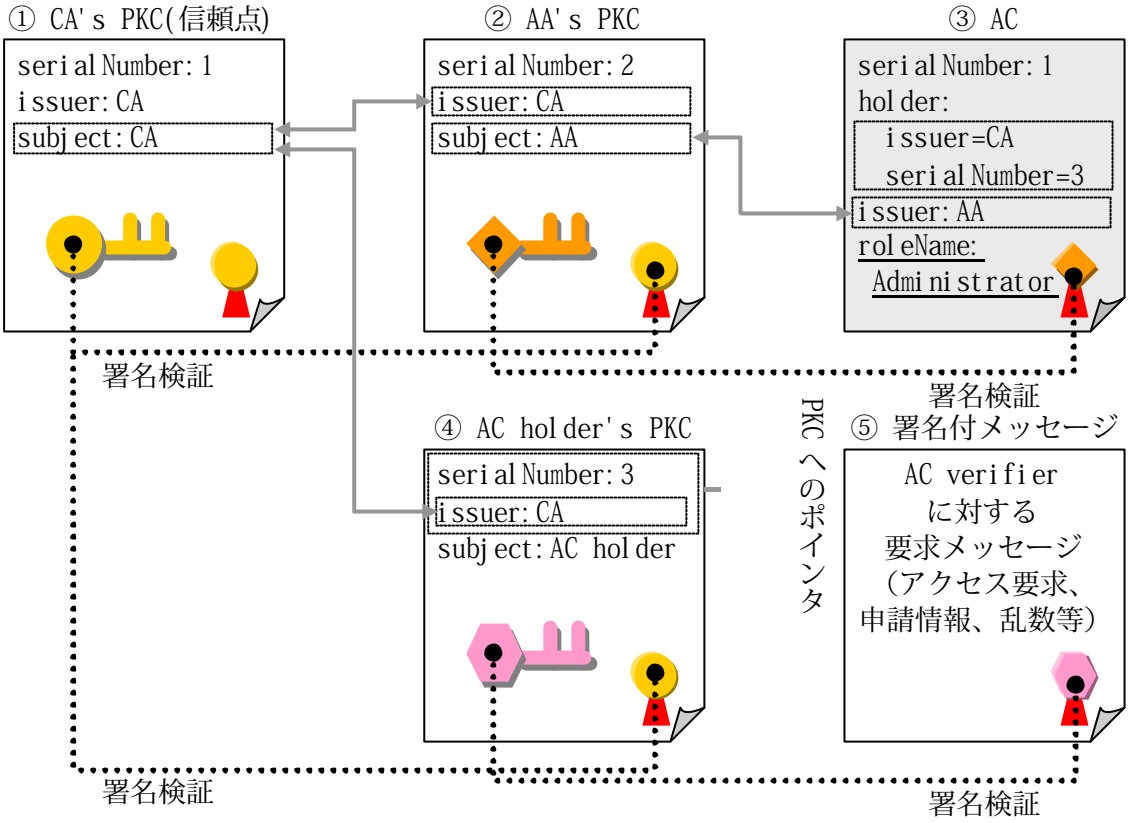


図 4-7 AA と AC 保有者が同一の CA から PKC の発行を受けている場合の認証パス

(1) AC 保有者の正当性確認

4.4.1.1(1)で記述したように、AC 検証者は、AC 保有者から受信した要求メッセージ⑤に付与されている電子署名を、AC 保有者の PKC を用いて検証する。AC 保有者の PKC は、AC③の holder フィールドから特定する。図 4-7 では、holder フィールドに baseCertificateID 型を使用した場合の例を記載しているが、本フィールドが示す issuer=CA、serialNumber=3 である PKC④を AC 保有者の PKC とする。

AC 保有者の PKC④の正当性を検証するため、RFC3280 の検証手順に則り、AC 検証者の信頼点である CA の PKC①から、AC 保有者の PKC④までの認証パスを構築し、署名チェーンの検証と有効性確認を行う。本モデルでは、PKC①の直接配下に PKC④が存在するため、PKC④に付与されている電子署名を、PKC①を用いて検証する。PKC④の有効性確認については、CA が発行した失効情報(CRL、OCSPResponse 等)を取得して検証を行う。

これらの処理を行うことによって、AC 保有者の正当性を確認することになる。

(2) AC の正当性確認

4.4.1.1(2)で記述したように、AC 検証者は、AC③に付与されている電子署名を、AA の PKC を用いて検証する。AA の PKC は、AC③の issuer フィールドから特定する。図 4-7 では、issuer フィールドに v2Form の issuerName 型を使用した場合の例を記述しているが、本フィールドによって示されている subject フィールドが CA である PKC②を AA の PKC として特定する。

AA の PKC②の正当性を検証するため、RFC3280 の検証手順に則り、AC 検証者の信頼点である CA の PKC①から、AA の PKC②までの認証パスを構築し、署名チェーンの検証と有効性確認を行う。本モデルでは、PKC①の直接配下に PKC②が存在するため、PKC②に付与されている電子署名を、PKC①を用いて検証する。PKC②の有効性確認については、CA が発行した失効情報(CRL、OCSPResponse 等)を取得して検証を行う。これらの処理を行うことによって、AC の内容の正当性を確認することになる。

(3) AA の適合性確認

4.4.1.1(3)で記述したように、上記(2)で特定した AA の PKC②が RFC3280 に準拠していることを確認する。それに加えて、②の PKC の拡張項目として keyUsage 拡張が含まれている場合は、digitalSignature のビットが 0 となっていないことを確認する。また、basicConstraints 拡張が含まれている場合は、cA フィールドが TRUE となっていないことを確認する。

(4) AA の信頼性確認

4.4.1.1(4)で記述したように、上記(2)で検証を行った AC③の発行者が、当該 AC に記載されている属性を付与する組織・人として信頼できるかどうかを確認する。図 4-7 では、AC に role 属性を割り当て、その roleName フィールドに Administorator という属性値を設定している。本項の前提として、AC 検証者は、role 属性を含んだ AC の発行組織として AA を信頼しているので、当該 AC を受け入れることができる。この処理によって、AA が属性の割り当て機関として信頼できるかどうかを確認することになる。

(5) AC の有効期間チェック

4.4.1.1(5)で記述したように、AC の検証日時が、上記(2)で特定した AA の PKC②の attrCertValidityPeriod フィールドに記載されている有効期間の範囲内であることを確認する。

(6) AC 受入資格の確認

4.4.1.1(6)で記述したように、AC③に targetInformation 拡張が含まれている場合は、当該拡張の値に記載されている名前もしくは組織が、AC 検証者と一致していることを確認する。

この処理を行うことにより、当該 AC は、意図した検証者にのみ提示されるようになる。

(7) クリティカルな拡張に関するチェック

4.4.1.1(7)で記述したように、AC 検証者の使用しているアプリケーションでサポートしていない拡張が AC③の中に含まれており、かつ、当該拡張が critical である場合は、その AC の受け入れを拒否する。

4.4.1.3　モデルを用いた場合の検証例（その２）

本項では、3.1 節で記述されている「モデル２」、すなわち、AA と AC 保有者が、異なる認証局 CA1 および CA2 から PKC の発行を受けているというモデルを例に挙げて説明する。

図 4-8 に示すモデルでは、PKC を発行する CA1 および CA2、AC を発行する AA、AC の発行を受けそれを所有する AC 保有者、および、AC 保有者から提示された属性を検証する AC 検証者という５つのエンティティが登場する。

CA1 は、自身に対して PKC①（自己署名証明書）を発行している。また、AA に対して PKC②を発行している。同様にして、CA2 は、自身に対して PKC④（自己署名証明書）を発行し、AC 保有者に対しても PKC⑤を発行している。AA は、AC 保有者に対して AC③を発行している。AC 検証者は、PKC を発行する組織として CA1 の PKC①および CA2 の PKC④の２つを信頼し、AC を発行する組織として AA を信頼しているものとする。

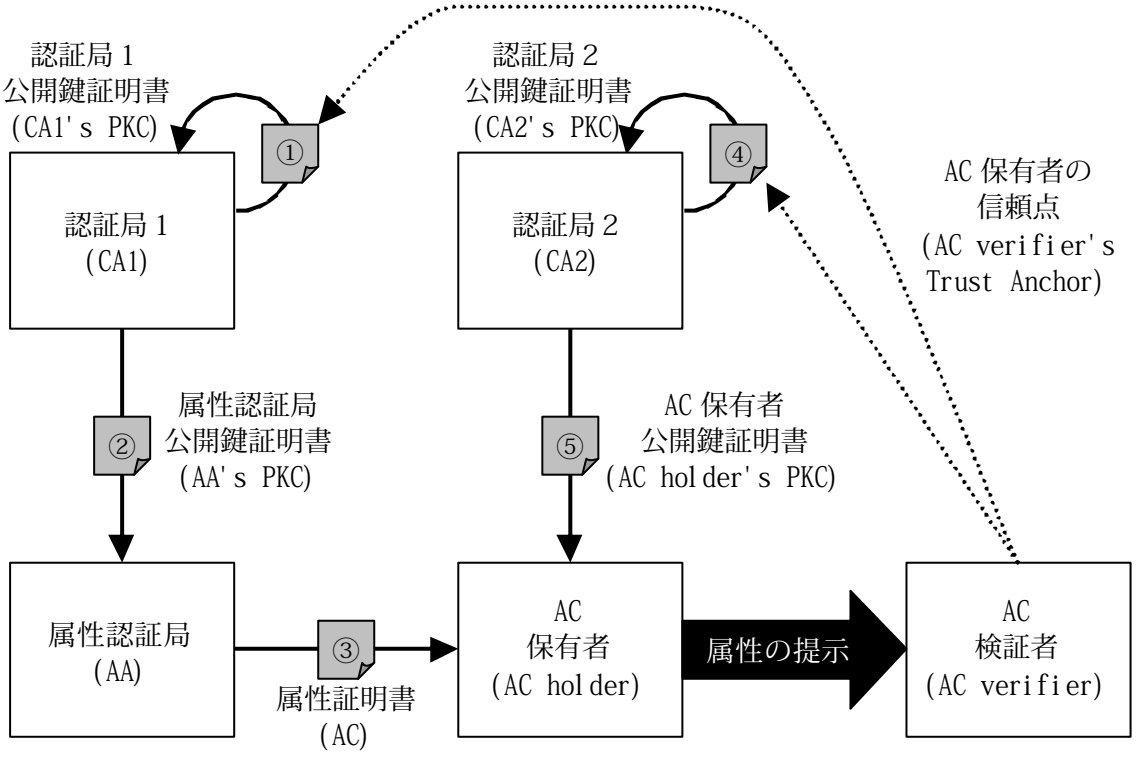


図 4-8 AA と AC 保有者が異なる CA から PKC の発行を受けている場合の属性認証モデル

このような属性認証モデルにおいて、AC 保有者が、AC 検証者に対して属性情報の提示を行った場合（すなわち、図 4-9 の AC③、当該 AC に関わる PKC 群①・②・④・⑤等、および、要求メッセージ⑥を AC 保有者から受信した場合）に、AC 検証者が行う AC の検証方法について説明する。

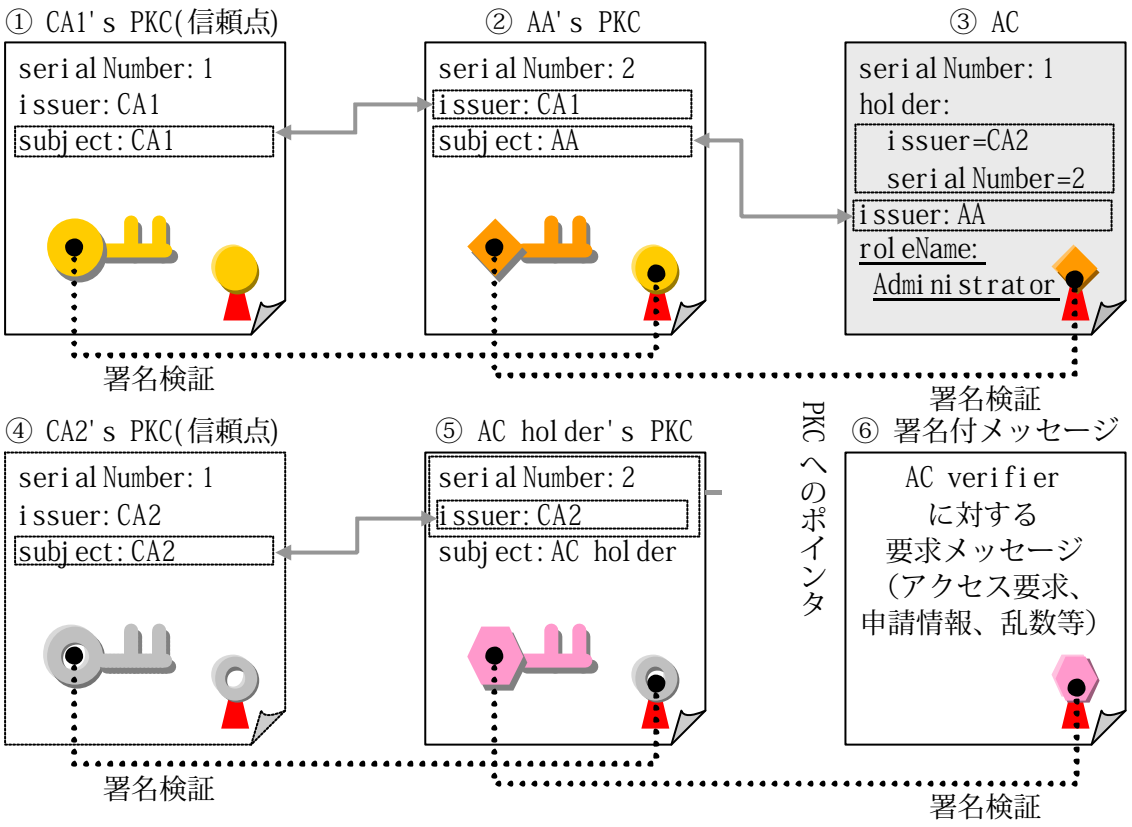


図 4-9 AA と AC 保有者が異なる CA から PKC の発行を受けている場合の認証パス

(1) AC 保有者の正当性確認

4.4.1.2(1)と同様、AC 検証者は、AC 保有者から受信した要求メッセージ⑥に付与されている電子署名を、AC 保有者の PKC を用いて検証する。AC 検証者の PKC は、AC ③の holder フィールドから特定する。図 4-9 では、holder フィールドに baseCertificateID 型を使用した場合の例を記載しているが、本フィールドが示す issuer=CA2、serialNumber=2 である PKC⑤を AC 保有者の PKC とする。

AC 保有者の PKC⑤の正当性を検証するため、RFC3280 の検証手順に則り、AC 検証者の信頼点である CA の PKC①から、AC 保有者の PKC⑤までの認証パスを構築し、署名チェーンの検証と有効性確認を行う。本モデルでは、PKC④の直接配下に PKC⑤が存在するため、PKC⑤に付与されている電子署名を、PKC④を用いて検証する。PKC⑤の有効性確認については、CA が発行した失効情報(CRL、OCSPResponse 等)を取得して検証を行う。

これらの処理を行うことによって、AC 保有者の正当性を確認することになる。

(2) AC の正当性確認

4.4.1.2(2)と同様に、AC 検証者は、AC③に付与されている電子署名を、AA の PKC を用いて検証する。AA の PKC は、AC③の issuer フィールドから特定する。図 4-9 では、issuer フィールドに v2Form の issuerName 型を使用した場合の例を記述しているが、subject フィールドが CA である PKC②を AA の PKC として特定する。

AA の PKC②の正当性を検証するため、RFC3280 の検証手順に則り、AC 検証者の信頼

点である CA の PKC①から、AA の PKC②までの認証パスを構築し、署名チェーンの検証と有効性確認を行う。本モデルでは、PKC①の直接配下に PKC②が存在するため、PKC②に付与されている電子署名を、PKC①を用いて検証する。PKC②の有効性確認については、CA が発行した失効情報(CRL、OCSPResponse 等)を取得して検証を行う。これらの処理を行うことによって、AC の内容の正当性を確認することになる。

(3) AA の適合性確認

4.4.1.2(3)と同様に、上記(2)で特定した AA の PKC②が RFC3280 に準拠していることを確認する。それに加えて、②の PKC の拡張項目として keyUsage 拡張が含まれている場合は、digitalSignature のビットが 0 となっていないことを確認する。また、basicConstraints 拡張が含まれている場合は、cA フィールドが TRUE となっていないことを確認する。

(4) AA の信頼性確認

4.4.1.2(4)と同様に、上記(2)で検証を行った AC③の発行者が、当該 AC に記載されている属性を付与する組織・人として信頼できるかどうかを確認する。図 4-9 では、AC に role 属性を割り当て、その roleName フィールドに Administorator という属性値を設定している。本項の前提として、AC 検証者は、role 属性を含んだ AC の発行組織として AA を信頼しているので、当該 AC を受け入れることができる。
この処理によって、AA が属性の割り当て機関として信頼できるかどうかを確認することになる。

(5) AC の有効期間チェック

4.4.1.2(5)と同様に、AC の検証日時が、上記(2)で特定した AA の PKC②の attrCertValidityPeriod フィールドに記載されている有効期間の範囲内であることを確認する。

(6) AC 受入資格の確認

4.4.1.2(6)と同様に、AC③に targetInformation 拡張が含まれている場合は、当該拡張の値に記載されている名前もしくは組織が、AC 検証者と一致していることを確認する。
この処理を行うことにより、当該 AC は、意図した検証者にのみ提示されるようになる。

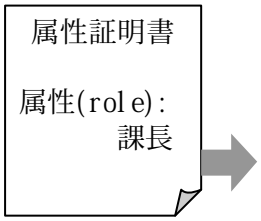
(7) クリティカルな拡張に関するチェック

4.4.1.2(7)と同様に、AC 検証者の使用しているアプリケーションでサポートしていない拡張が AC③の中に含まれており、かつ、当該拡張が critical である場合は、その AC の受け入れを拒否する。

4.4.1.4　属性の検証方法

AC の検証が完了し、AC の内容が適切なものであると確認できた場合には、当該 AC の中に記載されている属性情報（資格・権限等）と、AC 検証者側で事前に設定しているルールやアクセス制御ポリシーのような情報とを比較し、属性を提示した者に対してどのような行為を許可（もしくは制限）するのかを判断することになる。図 4-10 および図 4-11 に属性の検証例を示す。

図 4-10 は、社員用ポータルサイト（AC 検証者に相当）が、role 属性を含んだ AC を受領した場合に、role の値によって実行可能な業務を判断する例である。図 4-10 の場合、AC の role 属性には「課長」という値が設定されているので、事前に社員用ポータルサイトに設定されているルールに基づいて判断を行い、出張旅費申請業務と出張旅費承認業務については実行可能であるが、部内人事管理業務や社長印押印業務は実行できないというように Web 上の業務システムに対する実行を制限する。



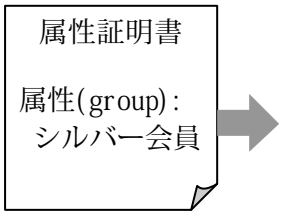
ルール

role	出張旅費申請業務	出張旅費承認業務	部内人事管理業務	…	社長印押印業務
社長	○	○	○		○
部長	○	○	○		×
課長	○	○	×		×
社員	○	×	×		×

○：実行可能

図 4-10 属性の検証例（その１）

また、図 4-11 は、クレジットカード等の会員専用 Web サイト（AC 検証者に相当）が、group 属性を含んだ AC を受領した場合に、group の値によって実行可能な業務を判断する例である。図 4-11 の場合、AC の group 属性には「シルバー会員」という値が設定されているので、事前に Web サイトに設定されているアクセス制御ポリシーに基づいて判断を行い、ページ A、ページ B、ページ C については閲覧可能であるが、ページ D はゴールド会員専用ページのため閲覧できないというように Web ページへのアクセスを制限する。



アクセス制御ポリシー

group	ページ A	ページ B	ページ C	…	ページ D
ゴールド会員	○	○	○		○
シルバー会員	○	○	○		×
ブロンズ会員	○	×	○		×

○：閲覧可能

図 4-11 属性の検証例（その２）

以上のような形で、属性の検証は行われるようになる。属性の検証は、アプリケーションによって検証に必要な属性の種類やルール等が異なるため、アプリケーションレベルで実装することとなる。

AC をある 1 つの閉じた組織内で使用する場合には、属性の値の解釈に差異は発生しないが、AC を汎用的に広い世界で使用しようとした場合、例えば、A 社が A 社の課長に対して発行した AC と B 社が B 社の課長に対して発行した AC では、その位置づけは必ずしも同格ではない。極端な話をいえば、全く無関係な赤の他人が、ある特定の人に対して課長という属性値を割り当てた AC を発行することもできる。そのため、AC に課長という属性値が割り当てられていても、それをそのまま信用して使用することはできない。属性値は、どういう人・組織によって割り当てられていて、その属性値の割り当て基準がどういうものであって、その属性値をどういうふうに解釈すればよいのかという部分が属性認証の重要なポイントとなってくる。

このようなことから、AC を広く使用するには、AC を発行する側と AC を検証する側の間で属性の値に関する共通の認識を持つ必要がある。X.509(2000)では、属性や役割の定義を行うために、attributeDescriptor 拡張や Role Specification Certificate を用いる仕様も規定されているが、属性情報の割り当て基準や意味に関する標準化や AA の認定制度等によって各利用者が共通で使用できる属性情報を定義していくことが今後の課題となる。

4.4.2 汎用製品での実装

本項では､既存の PKI 製品をベースにして AC の検証を実現しようとした場合に必要な機能について検討を行う。

4.4.2.1 想定環境

図 4-12 は、AC 検証者システムの概観の一例を示す図である。図 4-12 において、アプリケーションとは、後述の属性認証プログラムを利用して、属性情報による資格や権限を検証する必要のあるアプリケーションプログラム（例えば、アクセス制限された Web サービスプログラム、電子申請プログラム等）である。また、AC 検証プログラムとは、AC 保有者から送信されてきた AC の検証を行うために必要な各種処理を実現するプログラムである。PKI プログラムは、署名データの生成・検証や PKC の解析・検証を行うためにプログラムであり、既存の PKI 製品や暗号ライブラリで実装されているものである。

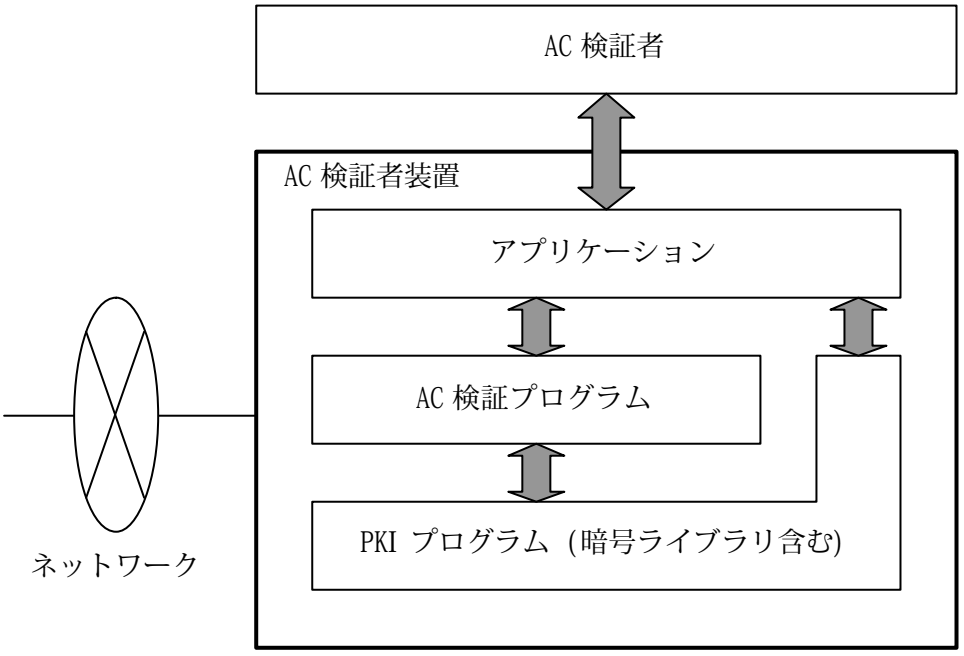


図 4-12 AC 検証者システムのモデルの概観

既存の PKI プログラムでは、具体的に、以下に列挙するような機能を実装しているものと想定する。ここで列挙した機能が、既存の PKI プログラムで実装されていない場合には、属性認証プログラムで実装することとなる。

- 署名データ(RFC3369 CMS v4 SignedData 等)の生成・解析・検証
- X.509 公開鍵証明書の解析
- X.509 公開鍵証明書の検証（認証パスの構築・検証、有効性確認）
- 署名値(RFC2437 等)の生成・検証
- ディレクトリサーバから特定のエンティティの属性に関する値の取得

4.4.3　各プログラムにおける機能要件

属性認証プログラムに必要とされる機能について、前項で記載した AC の検証手順をもとに、push 型と pull 型に分けて検討を行った。各プログラムにおいて必要と想定される機能は表 4-5 および表 4-6 のようになる。但し、本検討結果は、あくまでも実装の一例を示したものであり、各プログラムの仕様を限定するものではない。

「手順」の欄は、「4.4.1.1　RFC3281 における検証手順」における手順を示している。前節の手順に加えて、事前設定フェーズ、属性の検証フェーズ、および、その他の時点で実施される機能についても列挙した。「機能」および「機能概要」の欄は、前節手順を実施するために必要な機能とその内容を説明している。「利用者による操作」、「アプリケーション」、「AC 検証プログラム」、「ＰＫＩプログラム」の欄は、それぞれの機能がどのプログラムで実装されることになるかを示すものである。その機能が実装されると想定した欄には○を、場合によって実装される欄には△を記入している。

表 4-5 AC 検証者側に必要な機能一覧(push 型の場合)

手順	機能	機能概要	利用者による操作	アプリケーション	AC検証プログラム	PKIプログラム
(0)事前設定	①検証ポリシーの設定	認証パスの信頼点や信頼する証明書ポリシなど、検証に必要なポリシーを AC 検証者が設定する。但し、ある組織内に複数の利用者が存在し、当該組織において統一のポリシーを持たせたいというような場合においては、当該組織のシステム運用管理者がポリシーを設定して各利用者には触らせないようにするというような実装が望まれる。	○	○		
	②ルールの設定	本表(8)の属性の検証時において、どの属性にどのような権限を与えるかということを、AC 検証者が設定する。但し、ある組織内に複数の利用者が存在し、当該組織において統一のルールを持たせたいというような場合においては、当該組織のシステム運用管理者がルールを設定して各利用者には触らせないようにするというような実装が望まれる。	○	○		

(1)	①署名データの解析および署名検証	AC 保有者から送付されてきた署名データの解析を行う。また、署名データに付与されている署名者の PKC を用いて、署名値の検証を行う。 既存の PKI 製品において、RFC3369 CMS v4 SignedData のような AC を含む署名データを取り扱うことができない場合には、AC 検証プログラムで本機能を実装する必要がある。			△	○
	②署名者の PKC の解析	本表(1)①にて取り出した署名者の PKC について、各フィールドの値の解析を行う。				○
	③AC の解析	本表(1)①にて取り出した AC 保有者の AC について、各フィールドの値の解析を行う。			○	
	④署名者と AC 保有者の同一性確認	本表(1)②にて取り出した署名者の PKC の issuer フィールドおよび serialNumber フィールドと、本表(1)③にて取り出した AC の holder フィールドとを比較する。これにより、署名者が AC 保有者であるということが確認できるため、以降では署名者を AC 保有者として記述する。 （上記記述は、holder フィールドに baseCertificateID を適用していることを前提としているが、holder フィールドにそれ以外の型を使用している場合には、それに応じた同一性の確認を行う。）			○	
	⑤AC 保有者の PKC の検証	本表(1)①にて取り出した AC 保有者の PKC について、認証パス構築・検証および有効性確認を行う。				○
(2)	①AA の PKC の取得	本表(1)③にて取り出した issuer フィールドの値をもとに、本表(1)②にて取り出した署名データ中の certificates フィールドから AA の PKC を取得する。署名データ中に AA の PKC が含まれていないような場合には、リポジトリから AA の PKC を取得する。リポジトリから PKC を取得する機能は、既存の PKI プログラムの機能を流用できるものと想定される。			○	△

	②AA の PKC の解析	本表(2)①にて取得した AA の PKC について、各フィールドの値の解析を行う。				○
	③AC の署名検証	AC に付与されている AA の署名値を AA の公開鍵で検証する。具体的には、本表(1)③で取り出した AC の署名値を、本表(2)②で取り出した AA の公開鍵にて復号し、本表(1)③で取り出した AC の署名対象部のハッシュ値と比較する。				○
	④AA の PKC の検証	本表(2)①にて取得した AA の PKC について、認証パス構築・検証および有効性確認を行う。				○
	⑤AC の有効性確認	AC に失効の可能性がある場合(noRevAvail 拡張が使用されていない場合)には、本表(1)③にて取り出した内容をもとに、AC の失効情報を取得し、当該 AC が有効性であるかどうかを確認する。(ACRL の取得・解析・検証、OCSP の送受信、解析・検証においては、既存の PKI プログラムを流用できる部分もあると想定される。)			○	○
(3)	①AA の PKC のプロファイル確認	本表(2)①にて解析した AA の PKC の keyUsage 拡張や basicConstraints 拡張などの値が RFC3281 の規定通りとなっているかどうかを確認する。			○	
(4)	①AC の有効期間チェック	本表(1)③にて取り出した AC の有効期間と(AC 検証者装置のシステム時間などから取得した)検証日時とを比較し、当該 AC が検証時点において有効期間内であることを確認する。			○	
(5)	①AC の信頼点の確認	本表(1)①にて取り出した AC が、当該 AC に記載されている属性を割り当てるにふさわしいエンティティであるかを、AC の信頼点をもとに確認する。			○	
(6)	①Targeting Information 拡張の確認	本表(1)③にて取り出した AC の Targeting Information 拡張の内容をもとに、AC 検証者が当該 AC 受領してもよいかどうかを判断する。			○	

(7)	①AC の拡張のクリティカルフラグの確認	本表(1)③にて取り出した AC の拡張において、クリティカルフラグが TRUE となっている拡張ついては、当該拡張値の解析や拡張に伴う処理をサポートしていることを確認する。（拡張に伴う処理はこの時点で実行されるかもしれない。）			○	
(8)属性の検証	①属性の検証	本表(1)③にて解析された AC の属性情報と、本表(0)②にて設定されたルールとを比較し、AC を提示した者（AC 保有者）の要求を許可するかどうかを判断する。		○		
その他	AC の内容の表示	AC 検証者が、AC 保有者から送付されてきた AC の内容を確認したいという場合には、AC の内容を画面に表示する。 AC の解析には、本表(1)③で記載した AC 検証プログラムの機能を利用する。	○	○	○	
	属性認証に関わるログの記録および表示	AC 検証プログラムでは、各処理（特に検証ポリシー等の保護資産情報にアクセスを行う場合）の実施状況をログとして出力する。アプリケーションでは、属性認証プログラムから出力されたログを保存しておき、AC 検証者からログの内容を確認したいという要求があった場合にはそれを表示する。 (PKI プログラムに関するログの記録・表示については、別途実装されているものと想定する。)	○	○	○	

表 4-6 AC 検証者側に必要な機能一覧(pull 型の場合)

手順	機能	機能概要	利用者による操作	アプリケーション	AC検証プログラム	PKIプログラム
(0) 事前設定	①検証ポリシーの設定	認証パスの信頼点や証明書ポリシなど、検証に必要なポリシーを AC 検証者が設定する。但し、ある組織内に複数の利用者が存在し、当該組織において統一のポリシーを持たせたいというような場合においては、当該組織のシステム運用管理者がポリシーを設定して各利用者には触らせないようにするというような実装が望まれる。 AC の取得先(リポジトリの URI など)が AC 保有者から送付されてくるデータに含まれないシステムの場合には、AC の取得先も設定しておく必要がある。	○	○		
	②ルールの設定	本表(8)の属性の検証時において、どの属性にどのような権限を与えるかということを、AC 検証者が設定する。但し、ある組織内に複数の利用者が存在し、当該組織において統一のルールを持たせたいというような場合においては、当該組織のシステム運用管理者がルールを設定して各利用者には触らせないようにするというような実装が望まれる。	○	○		
(1)	①署名データの解析および署名検証	AC 保有者から送付されてきた署名データの解析を行う。また、署名データに付与されている署名者の PKC を用いて、署名値の検証を行う。 既存の PKI 製品において、RFC2630 CMS v3 SignedData のような AC を含む署名データを取り扱うことができない場合には、AC 検証プログラムで本機能を実装する必要がある。			△	○

	②署名者の PKC の解析	本表(1)①にて取り出した署名者の PKC について、各フィールドの値の解析を行う。				○
	③AC の取得	(0)①にて設定した AC の取得先や(1)①にて解析した署名データの内容をもとに、リポジトリから AC 保有者の AC を取得する。AC の取得においては、既存の PKI プログラムの機能を流用できる場合もある。			○	△
	④AC の解析	本表(1)③似て取得した AC 保有者の AC について、各フィールドの値の解析を行う。			○	
	⑤署名者と AC 保有者の同一性確認	本表(1)②にて取り出した署名者の PKC の issuer フィールドおよび serialNumber フィールドと、本表(1)④にて取り出した AC の holder フィールドとを比較する。これにより、署名者が AC 保有者であるということが確認できるため、以降では署名者を AC 保有者として記述する。 （上記記述は、holder フィールドに baseCertificateID を適用していることを前提としているが、holder フィールドにそれ以外の型を使用している場合には、それに応じた同一性の確認を行う。）			○	
	⑥AC 保有者の PKC の検証	本表(1)①にて取り出した AC 保有者の PKC について、認証パス構築・検証および有効性確認を行う。				○
(2)	①AA の PKC の取得	本表(1)④にて取り出した issuer フィールドの値をもとに、本表(1)②にて取り出した署名データ中の certificates フィールドから AA の PKC を取得する。署名データ中に AA の PKC が含まれていないような場合には、リポジトリから AA の PKC を取得する。リポジトリから PKC を取得する機能は、既存の PKI プログラムの機能を流用できるものと想定される。			○	△
	②AA の PKC の解析	本表(2)①にて取得した AA の PKC について、各フィールドの値の解析を行う。				○

	③AC の署名検証	AC に付与されている AA の署名値を AA の公開鍵で検証する。具体的には、本表(1)④で取り出した AC の署名値を、本表(2)②で取り出した AA の公開鍵にて復号し、本表(1)④で取り出した AC の署名対象部のハッシュ値と比較する。				○
	④AA の PKC の検証	本表(2)①にて取得した AA の PKC について、認証パス構築・検証および有効性確認を行う。				○
	⑤AC の有効性確認	AC に失効の可能性がある場合(noRevAvail 拡張が使用されていない場合)には、本表(1)④にて取り出した内容をもとに、AC の失効情報を取得し、当該 AC が有効性であるかどうかを確認する。(ACRL の取得・解析・検証、OCSP の送受信、解析・検証においては、既存の PKI プログラムを流用できる部分もあると想定される。)			○	○
(3)	①AA の PKC のプロファイル確認	本表(2)①にて解析した AA の PKC の keyUsage 拡張や basicConstraints 拡張などの値が RFC3281 の規定通りとなっているかどうかを確認する。			○	
(4)	①AC の有効期間チェック	本表(1)④にて取り出した AC の有効期間と(AC 検証者装置のシステム時間などから取得した)検証日時とを比較し、当該 AC が検証時点において有効期間内であることを確認する。			○	
(5)	①AC の信頼点の確認	本表(1)③にて取得した AC が、当該 AC に記載されている属性を割り当てるにふさわしいエンティティであるかを、AC の信頼点をもとに確認する。			○	
(6)	①Targeting Information 拡張の確認	本表(1)④にて取り出した AC の Targeting Information 拡張の内容をもとに、AC 検証者が当該 AC 受領してもよいかどうかを判断する。			○	
(7)	①AC の拡張のクリティカルフラグの確認	本表(1)④にて取り出した AC の拡張において、クリティカルフラグが TRUE となっている拡張ついては、当該拡張値の解析や拡張に伴う処理をサポートしていること			○	

		を確認する。（拡張に伴う処理はこの時点で実行されるかもしれない。）				
(8)属性の検証	①属性の検証	本表(1)④にて解析された AC の属性情報と、本表(0)②にて設定されたルールとを比較し、AC を提示した者（AC 保有者）の要求を許可するかどうかを判断する。		○		
その他	AC の内容の表示	AC 検証者が、AC 保有者から送付されてきた AC の内容を確認したいという場合には、AC の内容を画面に表示する。 AC の解析には、本表(1)④で記載した AC 検証プログラムの機能を利用する。	○	○	○	
	属性認証に関わるログの記録および表示	属性認証プログラムでは、各処理（特に検証ポリシー等の保護資産情報にアクセスを行う場合）の実施状況をログとして出力する。アプリケーションでは、AC 検証プログラムから出力されたログを保存しておき、AC 検証者からログの内容を確認したいという要求があった場合にはそれを表示する。 (PKI プログラムに関するログの記録・表示については、別途実装されているものと想定する。)	○	○	○	

表 4-5 および表 4-6 より、属性認証プログラムとして、以下の機能を実装する必要があると考えられる。

- 署名データの解析および署名検証(AC を含む署名データの処理系が実装されていない場合)
- AC の解析
- 署名者と AC 保有者の同一性確認
- AA の PKC の取得(署名データに AA の PKC が含まれている場合)
- AC の有効性確認(noRevAvail 拡張が使用されていない AC を取り扱うシステムの場合)
- AA の PKC のプロファイル確認

- AC の有効期間チェック
- AC の信頼点の確認
- Targeting Information 拡張の確認
- AC の拡張のクリティカルフラグの確認
- AC の内容の表示
- 属性認証に関わるログの記録および表示
- AC の取得(pull 型を採用する場合)

4.4.4 検証手順からみた性能および運用性における配慮

本項では、属性認証に関するシステムを開発・構築するにあたってどのような点に配慮すればよいのかを、属性の検証における観点から述べる。

4.4.4.1 性能面における配慮

AC の検証は、AC の署名検証や有効性確認に加えて、AC 保有者の PKC と、AA の PKC という２つの認証パスの検証を行う必要があるため、処理手順は多い。そのため、上記の２つの認証パスがどれだけ複雑なものであるかによって処理性能が影響してくる。検証にあまり時間を費やさないようにするためには、できるだけ単純な PKI のモデルで属性認証を実現することが望ましい。検証に必要な PKC 群を署名データに含める形式を採用することや、noRevAvail 拡張を使用することによって、リポジトリにアクセスする処理を省略することができるため、結果として処理時間を減らすこともできる。

また、push 型と pull 型で比較した場合、push 型は、AC 検証者が AC を取得する必要のない分、pull 型に比べ、処理は高速である。但し、pull 型では、AC 保有者が誤った AC を送付してしまうことも考えられ、この場合、AC 保有者は AC を受け取ってもらうまでリクエストを再送し続けるなど、トラフィックが増加する恐れがある。pull 型は、AC 検証者側に AC を取得する機能が必要となる。また、当該機能により、外部ネットワークへの通信が頻繁に発生するため、負荷がかかることを考慮に入れておく必要がある。

4.4.4.2 運用面における配慮

属性の検証を行うにあたり、AC 検証者が行うべき手続きや運用について配慮する事項について整理してみる。

まず、AC 検証者は、認証パスの信頼点や証明書ポリシなど、検証に必要なポリシーを設定しておく必要がある。AC の取得先(リポジトリの URI など) が AC 保有者から送付されてくるデータに含まれないシステムの場合には、AC の取得先も設定することになる。また、属性の検証時において、どの属性にどのような権限を与えるかというルールについても設定しておく必要がある。但し、ある組織内に複数の利用者が存在し、当該組織において統一の検証に関するポリシーやルールを持たせたいというような場合においては、当該組織のシステム運用管理者がポリシーやルールを設定して各利用者には触らせないようにすることが望まれる。さらに push 型の場合には、AC 検証者が提供している業務においてどういう AC であれば受け入れることが可能であるかを、AC

保有者側に明示しておく必要がある。pull 型の場合には、AC がリポジトリから取得できるようにしておくよう、手筈を整えておく必要がある。

4.4.4.3　即時性を実現する上での留意事項

AC 検証者が、CA および AA から提供された失効情報をリアルタイムに取得・検証するためには、失効における即時性要件を満たすとともに、検証における即時性要件を満たす必要がある。

この即時性を実現する上においては、留意事項すべき事項が存在する。

- 即時性と応答性（検証時間）は、トレードオフの関係にあるということである。即時性を重視すると、以下の理由により、CA, AA, AC 検証者での処理が非常に大変になる。そのため、応答性（検証時間）は長くなる。また、逆に応答性を重視した場合、即時性が失われる。

- 即時性を満たすためには、AC の検証に必要となる AC や PKC すべてについての有効性確認を行う必要がある。また、それらの有効性確認に必要な失効情報を取得するために CA, AA, VA にアクセスする必要がある

そのため、実際の適用システムにおいては、設計者がどの程度の即時性が必要かを考慮した上で、有効性確認方法などを決める必要がある。

付録：公開鍵証明書/属性証明書の有効性確認方法

公開鍵証明書および属性証明書の有効性確認の方法としては、以下の２種類が存在する

１）OCSP：Online Certificate Status Protocol。RFC2560 に記載。認証局から CRL を Client 側で解析し証明書が失効されているかどうかを調べる処理の負担をなくして証明書の状態を Online で確認するために規定されたプロトコル。

２）CRL：Certificate Revocation List。ITU-T 勧告 X.509, RFC3280 に記載。CA により発行される証明書の破棄リスト。

<OCSP を AC の有効性確認に使用する場合の注意点>

OCSP を AC の有効性確認のために使用する場合、以下の点について注意が必要である。

RFC2560(OCSPv1)で規定されている OCSPRequest において、検証対象となる証明書を指定する情報として、以下を指定する必要がある。

・発行者の識別名のハッシュ値
　属性証明書の発行者フィールドから算出可能。
　但し、issuerName を使用していること（それ以外の型であると識別名がわからなくなるため)、かつ、当該エンティティが CA でないこと（検証対象が PKC なのか AC なのかがわからなくなるため）

・発行者の公開鍵情報のハッシュ値
　属性証明書の AuthorityKeyIdentifier 拡張の keyIdentifier フィールドに値が入っていれば、設定可能。
　そうでない場合は、属性証明書の発行者（すなわち属性認証局）の公開鍵証明書から算出することになる。すなわち、属性証明書の発行者の公開鍵を事前に取得している必要あり。

・検証対象証明書のシリアル番号
　属性証明書のシリアル番号から取得可能。

以上のことから、検証要求者が事前に証明書発行者の公開鍵証明書を取得している（場合によっては認証パスの構築を行う）必要があることから、検証者自身が属性証明書発行者の公開鍵証明書を入手していない場合には属性証明書の有効性確認を OCSP で行うことはできない。

なお、検証者自身が属性証明書発行者の公開鍵証明書を入手していない場合においても検証が可能となるように OCSPv2(draft-ietf-pkix-ocspv2-ext-01.txt)という方式も提案されている。この OCSPv2 における検証対象証明書の指定方法では、属性証明書そのものを指定することが可能となっている。

4.5　属性証明書の失効

本節では、公開鍵証明書の失効と属性証明書の失効に関して、属性証明書の有効性確認の観点から以下について述べる。

- 属性証明書の失効の必要性

  属性証明書および公開鍵証明書の失効理由，属性証明書の有効期間の視点から、属性証明書の失効の必要性について述べる。

- 属性証明書の失効と検証の即時性とその実現性

  即時性が求められる状況に適用するための属性証明書の失効と検証の即時性要件を抽出し、その実現性について述べる。

4.5.1　属性証明書の失効の必要性

属性証明書は、その失効理由，属性証明書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属性証明書失効要否は分類される。

以降、属性証明書および公開鍵証明書の失効理由，属性証明書有効期間の視点から、属性証明書の失効の必要性について見ていく。

4.5.1.1　失効理由から見た属性証明書失効の必要性

属性証明書の失効理由としては、以下のものが挙げられる。

表 4-7 属性証明書の失効理由

失効理由	失効手順	
1）属性証明書自身の失効		
・AC 保有者が、自身の判断に基づいて属性証明書を失効させる場合	3.2.3 章の(a)	
・属性認証局および属性の管理主体が、自身の判断に基づいて AC 保有者の属性証明書を失効させる場合	3.2.3 章の(b)	
2）公開鍵証明書の失効による属性証明書の失効		
・AC 保有者が、自身の判断に基づいて自身の公開鍵証明書を失効し、その情報を受けて、属性証明書が失効される場合	3.2.3 章の(a)	AC 保有者の公開鍵証明書の失効後、左記失効手順により属性証明書を失効
・属性認証局および属性の管理主体が、自身の判断に基づいて AC 保有者の公開鍵証明書を失効し、その情報を受けて、属性証明書が失効される場合	3.2.3 章の(b)	AC 保有者の公開鍵証明書の失効後、左記失効手順により属性証明書を失効

・属性認証局が、自身の判断に基づいて AA の公開鍵証明書を失効し、その情報を受けて、属性証明書が失効される場合	3.2.3 章の(c)	
・認証局が、秘密鍵の漏洩などの理由により、自身の判断に基づいて CA の公開鍵証明書を失効し、その情報を受けて属性証明書が失効される場合	3.2.3 章の(d)	CA の公開鍵証明書を失効後、左記失効手順により属性証明書を失効
・認証局が、自身の判断に基づいて属性認証局，AC 保有者の公開鍵証明書を失効し、その情報を受けて、属性証明書が失効される場合	3.2.3 章の(d)	属性認証局，AC 保有者の公開鍵証明書を失効後、左記失効手順により属性証明書を失効

(1) 属性証明書自身の失効

属性証明書は、主にアクセス制御を目的とした資格確認のために用いられる。そのため、例えば、医師の違法処置による医師免許の失効、委任者が変更になったことによる電子委任状の失効など何かの理由により資格自体が剥奪される場合には、属性証明書を失効する必要がある。この失効の方法として、属性認証局あるいは運用主体が属性証明書を失効する場合とユーザ自身が属性証明書の失効依頼を属性認証局に対して行う場合がある。例えば、違法処理による医師免許の失効は前者であり、委任者変更による電子委任状の失効は後者で用いられる。

(2) 公開鍵証明書の失効による属性証明書の失効

属性証明書には、公開鍵証明書とのリンク情報が格納される。また、その属性証明書を発行した属性認証局の署名がうたれる。そのため、資格確認を行うためには、属性証明書の検証および有効性確認のみならず、属性証明書と関連づけられた公開鍵証明書の検証および有効性確認が必要になる。以上により、秘密鍵が危殆化した，証明書情報の変更などの理由による公開鍵証明書が失効された場合には、以下の理由により属性証明書に格納された属性を用いた属性認証が失敗する。そのため、AC 保有者あるいは運営団体から属性証明書に関する失効申請を受け取らない限り、必ずしも属性証明書を失効する必要はない。

- AA 公開鍵証明書の失効：AC の権限自体が無効になる
- CA 公開鍵証明書の失効：AA（および/あるいは）AC 保有者の本人確認ができない
- AC 保有者公開鍵証明書の失効：AC 保有者の本人確認ができない

ただし、PKC とのリンク方法によっては、秘密鍵が危殆化した場合において、AC も失効する必要のある場合がある（注意 1 参照)。

4.5.1.2　有効期間から見た属性証明書失効の必要性

属性証明書はその証明書内に有効期間を持つ。この有効期間の長さによっても、属性証明書失効の必要性を分類することができる。

1）有効期間が公開鍵証明書の有効期間より長い属性証明書（医師免許証など）
：属性証明書の失効（および検証時の AC の有効性チェック）を行うべき→有効期間が公開鍵証明書の有効期間より長い属性証明書は、医師免許証などその属性を持つ人が一生持ちまわる属性であるのが大半である。しかし、一方でこれらの属性は、医師免許取り消しなどその効果が取り消される場合がある。そのため、本来は属性証明書の失効（および検証時の AC の有効性チェック）を行う。

2）有効期間が公開鍵証明書の有効期間より比較的短い属性証明書（運転免許証，転居届，住民票など）
：属性証明書の失効（および検証時の AC の有効性チェック）を行うべき→本来は属性証明書の失効（および検証時の AC の有効性チェック）を行う必要がある。しかし、現実では、転居届や住民票などは現状発行日からの利用期限を決め、特に失効はしていないといった運用も存在しており、そのような運用を行う領域では、属性証明書の失効（および検証時の AC の有効性チェック）を行う必要はない。

3）有効期間が非常に短い属性証明書（(本人確認の必要な）サービスチケットなど）
：属性証明書の失効（および検証時の AC の有効性チェック）を行わなくてもよい→属性証明書の発行から有効期限切れまでの有効期間が非常に短いため、失効情報を発行して配布するために必要な時間よりも属性証明書の失効すると有効期間の方が早く終了してしまう。そのため、属性証明書の失効（および検証時の AC の有効性チェック）を行わなくてもよい。

なお、有効期間が公開鍵証明書の有効期間より長い属性証明書を発行する場合、属性証明書に格納する属性証明書の所有者を示す情報である holder 要素は、baseCertificateID を使用するより entityName や objectDigestInfo を使用するとよい。これにより、公開鍵証明書の更新時に属性証明書も更新する必要がなくなる。また、公開鍵証明書が失効された時でも、再度公開鍵を登録し、公開鍵証明書を発行することで、同じ AC を用いることが可能になる。(詳細：RFC3281 の 4.2.2 章/7.3 章参照)

＜注意 1＞
上記属性証明書に格納する属性証明書の所有者を示す情報である holder 要素として、entityName を利用した場合、以下の条件下において他人がもとの所有者の属性を使用することが可能になってしまう。

- 他人（悪意人）が、AC 保有者の entityName で AC 保有者とは異なる CA から公開鍵証明書を発行
- 作成した公開鍵証明書と AC 保有者が所持する AC を AC 検証者に対して送る
- AC 検証者が信頼する CA 公開鍵証明書にたどり着くことができる

→これは、現在の entityName の一意性が CA 単位で実施されていることが原因であり、①AC を使用する範囲内における全 CA 間での entityName の一意性を保証する機関を設ける，あるいは②AC 内 entityName 要素に AC 保有者の名前だけでなく、PKC 発行者名を付けるなどの対策

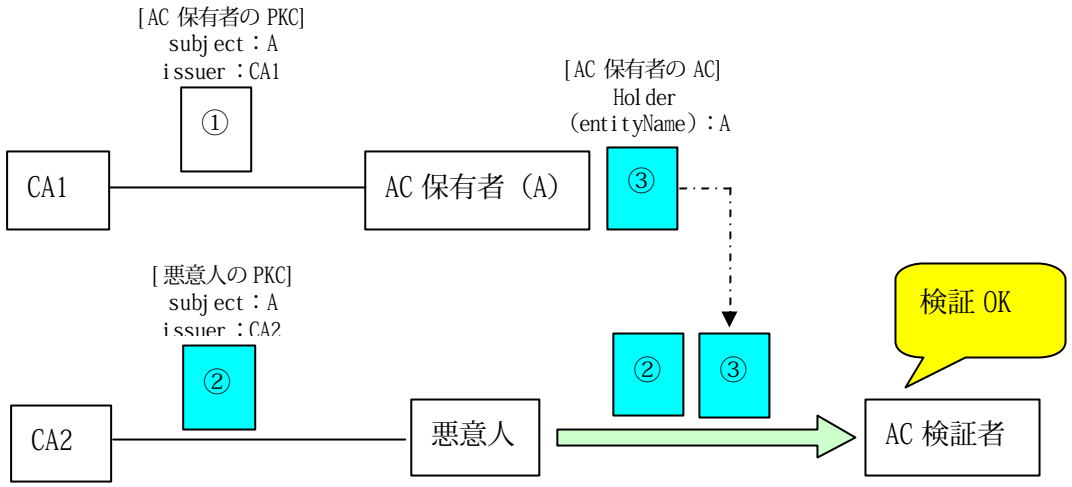


図 4-13 entityName 使用時の問題

＜注意２＞

上記属性証明書に格納する属性証明書の所有者を示す情報である holder 要素として、objectDigestInfo を利用し、かつ objectDigestInfo に格納する値として、RFC3281 の 7.3 章で述べられる“公開鍵ハッシュを使用する方法”を用いた場合、以下の条件下において他人がもとの所有者の属性を使用することが可能になってしまう。

・他人（悪意人）が、AC 保有者の鍵を盗み、その鍵で CA から公開鍵証明書を発行

・作成した公開鍵証明書と鍵を盗んだ AC 保有者が所持する AC を AC 検証者に対して送る

・AC 検証者が信頼する CA 公開鍵証明書にたどり着くことができる

→これは同じ公開鍵を持つ公開鍵証明書を作成することができてしまうことが原因である。そのため、objectDigestInfo を使用する際、上記のように鍵が危殆化した場合には、PKC だけでなく AC も失効させる必要がある。

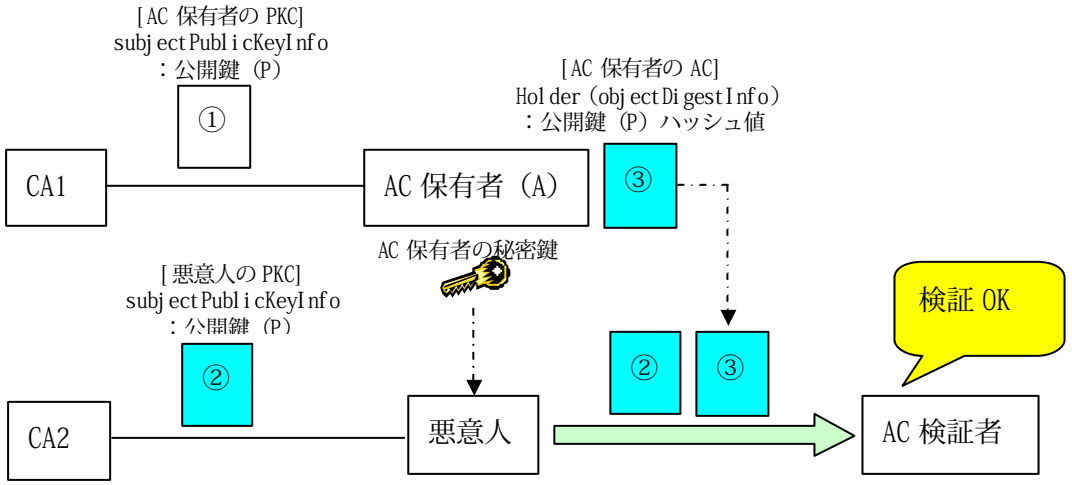


図 4-14 公開鍵ハッシュ使用時の問題

4.5.2 属性証明書の失効と検証の即時性要件

属性証明書の失効の必要性について上述したが、実際の運用においては、高額決済など即時性

（リアルタイム性）が求められる状況がある。

即時性を実現する上では、失効時の即時性要件だけでなく、検証時の即時性要件も満たすことで初めて AC 検証者が求める即時性が実現できる。

そこで、以下では、即時性が求められる状況に適用するための属性証明書の失効の即時要件だけでなく、検証の即時性要件についてもまた抽出する。

(1)　失効の即時性要件

即時性（リアルタイム性）を発揮するためには、属性認証局，認証局，AC 保有者，AC 検証者などすべての登場人物が責任を持ち、自身の鍵や属性に関して問題が発生した場合には、即座に対応をとる必要がある。

以上より、失効の即時性要件としては、次の要件があげられる。

- 属性の変化により属性証明書内容に変化が発生した際､即座に属性証明書の失効申請を行うこと
- 秘密鍵の危殆化など公開鍵証明書に関する変化が発生した際､即座に公開鍵証明書の失効申請を行うこと
- 認証局および属性認証局において、AC 保有者および運営主体などからの失効申請を常時受け付けることができること。また、これらの失効申請を即座に処理できること

(2)　検証の即時性要件

AC 検証者は、属性によるアクセス制御を行う前に、以下の検証を行う必要がある。

- AC 保有者の属性証明書の署名検証，属性認証局証明書から AC 検証者の信頼ポイントまでの認証パス検証（認証パス構築，署名検証，有効性確認など）
- AC 保有者の公開鍵証明書の署名検証，AC 保有者の公開鍵証明書発行者から AC 検証者の信頼ポイントまでの認証パス検証（認証パス構築，署名検証，有効性確認など）

上記を実施する際、リアルタイムな検証を行うためには、属性証明書や公開鍵証明書の有効性が主なポイントとなる。

以上により、検証の即時性要件としては、次の要件があげられる。

- 認証局および属性認証局が失効情報をリアルタイムに提供すること
- AC 検証者が上記提供された失効情報を取得・検証できること

<u>付録：属性証明書の失効情報の配布要否による属性認証局の運用の変化</u>

属性証明書の失効情報の配布要否により、属性認証局の運用が変化する。この運用に関して、RFC3281 において２つの方式が定義されている。

- NeverRevoke 方式

：失効情報を入手することはできないことを利用者（AC 保有者，AC 検証者など）に知らせる方式。属性証明書内の noRevAvail エクステンションを用いる。この方式を使用することを

示すためには、必ず本エクステンションが属性証明書内になければならない。

- Pointer in AC 方式
  ：失効情報の入手元を利用者（AC 保有者，AC 検証者など）に知らせる方式。属性証明書内の authorityInfoAccess エクステンションあるいは crlDistributionPoints エクステンションを用いる。

noRevAvail エクステンションがない場合は、失効情報の確認がサポートされていないことを暗に示す。そのため、属性証明書の失効をサポートしていない属性認証局，および失効をサポートはしているが、失効情報の配布は行わない属性認証局は、NeverRevoke 方式を使用する必要がある。一方、属性証明書の失効をサポートし、かつ失効情報の配布を行う属性認証局は、Pointer in AC 方式をサポートすることでよい（詳細は、RFC3281 の 6 章参照)。

5. 適用ガイド

2 章から 4 章では、属性の認証の要件とその実現技法を検討した。本章では、要件に応じた技法の選択についてその要点をまとめ、さらに今後に残された課題を提示する。

「5.1　適用技法の選択」では、属性証明書（AC）を利用する技法とその他の技法の選択指針をまとめる。「5.2　属性証明書を利用した属性認証の適用指針」では、AC を利用する技法における適用上の留意事項をまとめる。また「5.3　属性証明書による属性認証の今後の課題」では、今後 AC 利用の普及に際して解決すべき主要な課題を提示する。

5.1　適用技法の選択

属性を認証する技法には、属性証明書を用いる技法の他にも、公開鍵証明書（PKC）を用いる技法や属性認証サーバを用いる技法もある。それぞれに運用管理面、システム構築・保守面や利用者の利便性の面で以下に示す特徴があるので、業務の特徴に応じた技法を採用する必要がある。特徴の技術的な詳細は、2 章を参照されたい。

(1)　属性証明書（AC）を用いる技法

公開鍵証明書（PKC）を保有する利用者に対して、属性及び属性値を記載した属性証明書（AC）を発行する技法である。

この技法は、PKC を用いる技法と同様に、電子商取引や電子申請等、情報を送信する者の属性及び属性値を受信する者が確認する場合のような「End-to-End モデル」のアプリケーションに適している。送信者がその PKC を用いて送信情報と AC に電子署名を付与し、受信者がこれを検証する。

この技法では識別、認証のための一つの PKC に対して、属性ごとに AC を持つことができ、一般に有効期間の異なる複数の属性に柔軟に対応できる。また、属性によってはその有効期間が短く、失効処理が実際上意味を持たないものもあるが、AC による技法ではこれも想定している。汎用ソフトウェアが未整備であるためシステム構築が負担となることや、利用者の理解が必要なことなど、課題は少なくない。しかし、属性を認証する仕組みとして認知され、属性の種別やその表現が共通化されるなどその利用環境が整えば、社会において種々の属性を扱うための技法として利点が発揮されるものと考えられる。

(2)　公開鍵証明書（PKC）を用いる技法

公開鍵証明書（PKC）の利用は、識別、認証の手法として標準が確立しており、また実現技術も成熟度が高い。既存の PKC 発行サービスが利用でき、また、汎用のソフトウェアを利用して構築した認証局で PKC を発行することもできる。現状の PKC の利用を見ると、その機能は識別、認証のためだけでなく、属性の認証も含むものが多い。PKC 発行時に、利用者がある属性及び属性値を持つこともあわせて審査して、PKC に「会員証」や「資格証明」のはたらきを持たせるような場合である。すなわち、属性の認証は、PKC を用いる技法では既に実現されている。

この技法は、電子商取引や電子申請等、情報を送信する者の属性及び属性値を受信する者が

確認する場合のような「End-to-End モデル」のアプリケーションに適している。この点は、前項の AC を用いる技法と同様である。送信者がその PKC を用いて電子署名を付与し、受信者がこれを検証する。

この技法では一つの PKC で識別、認証と属性の認証をあわせて表現するので、利用者は、属性ごとに PKC を持つ必要がある。PKC の保持や、属性及び属性値の追加／変更に伴う再発行、失効を利用者が意識する必要がある。

(3) 属性認証サーバを用いる技法

属性認証サーバが、サーバ上の属性データベースを用いて属性認証を行う技法である。

この技法は、情報を保有している主体がそのアクセス管理を行う場合のように、多くの利用者の管理を特定の主体が集中的に行う場合に適している。電子商取引や電子申請等において、一つの受付窓口で多くの送信者の資格を確認する場合への適用も考えられる。属性や属性値の追加／変更／失効を管理者の判断で機動的に行えることも特徴である。

5.2 属性証明書を利用した属性認証の適用指針

本節では、属性証明書（AC）を用いる技法で属性認証を実現する場合の指針と留意事項をまとめる。

(1) システム設計

システム設計においては、当事者である AA、AC 保有者、AC 検証者のそれぞれに関して、運用における実現性、利便性、コストに配慮する。

- AA を認証する CA を選定する。ただし、AA の利用自体がまだ一般的でなく、したがって AA を認証する CA の選定も難しいのが実情である。そこで、既存の CA から選定するのではなく、この目的に CA を構築することも検討する。
- AC の前提とする AC 保有者の PKC と CA を決定する。（将来、様々な AC が広汎なアプリケーションで利用される場面を想定すると、個人を識別、認証する汎用的な PKC が利用できることが望ましい。しかし、現在はそのような PKC が存在しないため、当面の目的に PKC を発行するとともに、将来、汎用的な PKC の利用に移行することも現実的な解であると思われる。）
- AA の運用を想定し、その設計を行う。
  - AA の信頼の根拠として、CA における CP/CPS に相当するポリシーを策定する。（「4.1 属性認証局の信頼の根拠とポリシー」）
  - AC の発行、失効の手続を定める。（「3.2.1 属性証明書の発行手順」「3.2.3 属性証明書の失効手順」）
  - AC 保有者及び AC 検証者による AC の扱いに push 型と pull 型があり、その方式を選択する。また、選択した方式に必要な AA の仕組み（pull 型における AC のリポジトリ）を設計する。（「4.4.4 検証手順からみた性能および運用性における配慮」）

- AC 検証者の運用は、操作性と性能の面で実現方式が大きく影響する。AC 検証者の運用を想定し、その設計を行う。(「4.4.4 検証手順からみた性能および運用性における配慮」)

(2) 属性認証局（AA）の運用
- AA の管理者は、AA をそのポリシーに沿って運用する。(「4.1 属性認証局の信頼の根拠とポリシー」)
- AA の管理者は、ポリシーの遵守状況に関して、あらかじめ定めた監査を受ける。

(3) 属性証明書保有者（AC 保有者）の運用
- AC 保有者は、PKC と AC を利用するときに、保有する複数の PKC あるいは AC から正しいものを選ぶよう注意する。(「3.2.2 属性証明書の利用手順」「4.3 属性証明書の利用」)
- AC 保有者は、AC の失効事由が生じた場合には、速やかに AA に通知する。(PKC に失効事由が生じた場合と同様である。)(「3.2.3 属性証明書の失効手順」「4.5 属性証明書の失効」)

(4) 属性証明書検証者（AC 検証者）の運用
- AC 検証者は、検証のための事前設定を正しく行う。事前設定には、検証ポリシーの設定とルールの設定がある。(「表 4-5 AC 検証者側に必要な機能一覧（push 型の場合)」「表 4-6 AC 検証者側に必要な機能一覧（pull 型の場合)」)
- AC 検証者は、表示される AC の内容を確実に確認する。(同)

(5) ソフトウェア開発
- 属性証明書を利用する技法では、AA、AC 保有者及び AC 検証者の各システムに、このための機能を実現するソフトウェアが必要である。このソフトウェアは汎用の製品の場合と、個別に作成する場合が考えられる。いずれの場合にも、AC 保有者及び AC 検証者の利便性を特に重視する必要がある。適用業務における AC 利用の重要性を考えると、操作性がよく、また誤りを誘発しにくいことが重要である。
  - 属性認証利用者プログラムは、AC の選択に関して AC 保有者に分かりやすいものであること。(「4.3 属性証明書の利用」)
  - 属性認証検証者プログラムは、AC 検証者が行う事前設定に関して分かりやすいガイドと提示し、また結果も分かりやすく確認できること。(「4.4 属性の検証」)
  - 属性認証検証者プログラムは、検証の操作や属性証明書の表示が AC 検証者に分かりやすいこと。(「4.4 属性の検証」)

5.3 属性証明書による属性認証の今後の課題

属性証明書（AC）による属性認証は、汎用性やグローバルな相互運用性を持つ手法として今後の展開が期待される。その一方で、現時点では制度面などにおいて以下のような重要な課題もある。AC による属性認証が広く利用されるためには、これらの解決が必須であると思われる。

(1)　社会的受容

電子認証や PKC は、行政機関や企業では道具として受け入れられつつあるが、個人にはまだ馴染みの薄いものではないだろうか。まして、本報告で取り上げている属性認証や AC は、一般にはその言葉もまだ知られていない。AC の特徴が十分に生かされるのは、個人を識別、認証する汎用の PKC があり、その上で多様な AC が利用される場面であることを考えると、属性認証や AC は私人としての個人に理解され、受け入れられるるものでなければならない。そのためには、電子認証と属性認証に関する啓蒙が重要であると共に、「属性証明書」「検証」等の専門用語ではなく、「資格証明書」あるいは「社員証」等の日常のことばで説明する努力がベンダーと技術者に強く求められている。将来、個人にとってさまざまな手続きや取引が電子化され、そこに属性認証が適用された際に、一部の利用者だけが理解できるような技術であってはならない。

(2)　前提となる公開鍵証明書

AC を利用するには、属性を付与される対象を識別・認証するための PKC が必要である。しかし、AC の適用対象によっては、現時点で必ずしもこの条件がみたされていないことに留意する必要がある。その場合は、AC だけでなく PKC もあわせて用意する必要がある。

本来は、AC での必要性から PKC もあわせて用意するのではなく、汎用的な PKC が別途存在することを期待したい。このような PKC は、民間の事業者が発行する PKC が考えられる。また、行政機関が発行する PKC が、住民としての識別、認証の観点から望まれる場合も少なくないと思われる。行政機関が発行する PKC の用途に関して、民間の AC とあわせた利用も許すような合意が必要であると思われる。

(3)　属性の相互運用性

複数の AA のドメインにわたって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 の調整が必要になる。しかし、現状ではそれらの業界標準化／国際標準化は課題として残されている。例えば、病院・医療システムにおいて医師、看護士、技師などを属性として扱うにしても、属性や属性値について国際的に共通の分類が可能か否か自明ではなく、したがってグローバルに共通のオブジェクト ID 付与の可能性も検討課題である。

(4)　属性の検証ソフトウェア

属性と一口に言ってもさまざまであり、属性を検証する汎用的なソフトウェアを予め用意するのは容易ではない。市販の検証ソフトウェアを利用するにしても、現状では、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ことに留意する必要がある。

6. ＸＭＬ技術と属性

本報告書では、前章までで属性証明書を用いた属性管理手法の分析、まとめを行った。一方、現在の e ビジネスでは Web サービスが主流になって来ておりそこではユーザの認証のみならず属性やアクセス権限を一元管理し、シングルサインオンを実現した SAML(Security Assertion Markup Language)がある。本章では、まず 6.1 節で SAML そしてそこで使われている属性、アクセス管理を行う言語 XACML(eXtensible Access Control Markup Language)を紹介する。その他にも XML をベースとしたアクセス管理言語は多々あるが、本章では 6.2 節でそのいくつか主だったものを紹介し、6.3 節では本報告書で述べてきた属性証明書を利用した管理との比較を行う。最後に 6.4 節では、セキュリティ機能を実現するための XML の技術を簡単に紹介する。なお、XML に関する詳細技術は述べないため関連仕様を参照されたい。

6.1 SAML

6.1.1 概要

SAML(Security Assertion Markup Language)は、シングルサインオンを実現するために複数の Web サービスサーバ間でユーザの認証情報を引継ぎ、再ログインなしでセキュアな資源をアクセスさせるためのスキームであり OASIS で規定されている。

「SAML Authority」として、「Authentication Authority」、「Attribute Authority」、「Policy Decision Point(PDP)」が存在し、それぞれ、アクセス要求者 Subject を認証し認証した事（行為）を証明するための「Authentication Assertion」、Subject に関する属性情報である「Attribute Assertion」、特定の Resource(情報システム上のデータや提供されるサービス)に対してアクセスする事の認可を証明するための「Authorization Decision Assertion」を発行する。認証には X.509 PKI の公開鍵証明書(PKC)、属性認証、認可チェックには X.509 PMI の属性証明書(AC)を利用する事もできる。

OASIS の SAML 仕様では、Assertion の要素定義と Assertion のメッセージ交換を行う際のプロトコルの要素（Request/Queries/Response の各要素）が規定されている。

以下に Assertion の要素定義を示す。

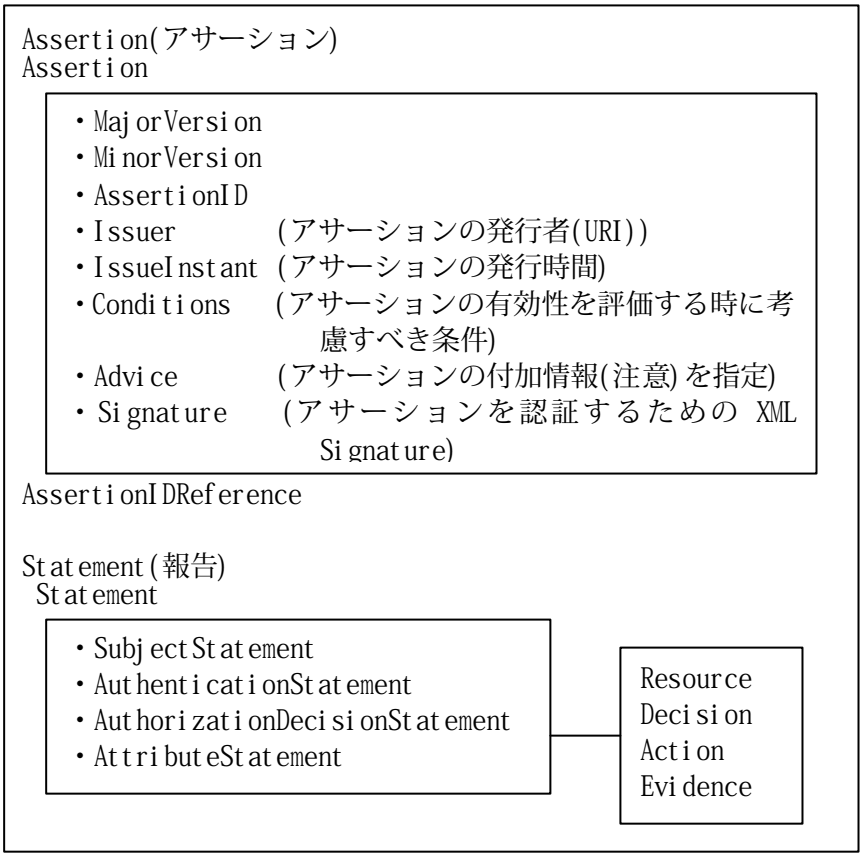


図 6-1 Assertion 定義内容

6.1.2 XACML

XACML(eXtensible Access Control Markup Language)は、XML データに関するアクセス要求(参照、更新など)について、要求元の情報、要求の内容、アクセス対象の組み合わせから、アクセスが許可されるか否かを判断するためのルールを記述するためのものであり、OASIS で規定されている。SAML でのアクセス制御を行う仕組みに利用されている。

アクセス制御を実施するためのデータフローモデルと、アクセス制御を行うためのルール、ポリシー(Policy Statement)を規定している。ポリシーは、ルール（ターゲット、条件、結果）と責務から成る。アクセス対象情報は、アクセス要求者、アクセス対象、操作から成り、通常複数のポリシー(Policy Statement Set)として定義される。

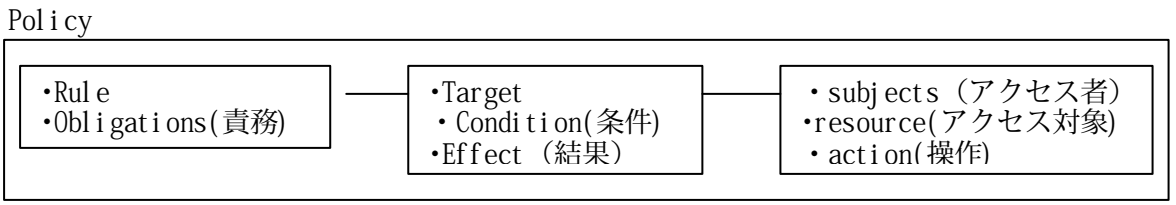


図 6-2 XACML 構造図

例：「患者であれば全てのレコードを読むことができる」

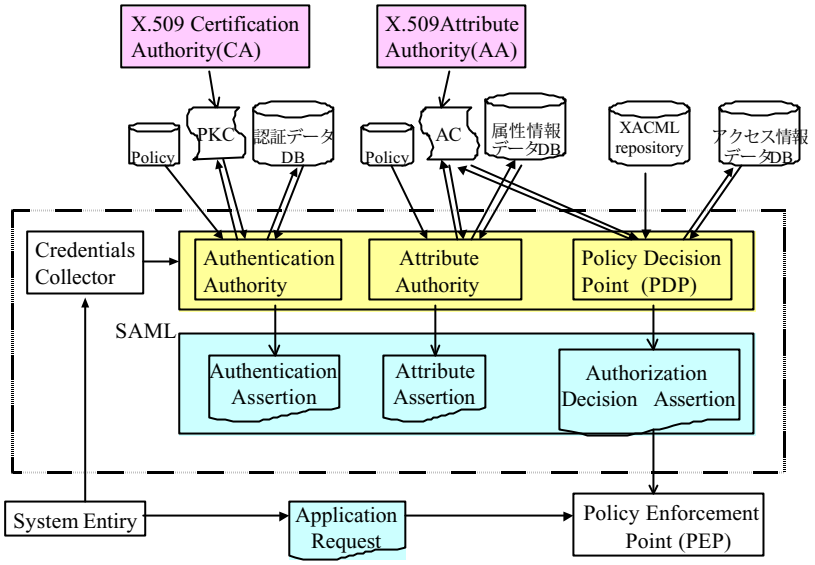
```
<rule ruleId="//medico.com/rules/rule1"
effect="Permit"
xmlns="…”  xsd 指定
      <target>
       <subjects>
         <saml:Attribute AttributeName="RFC822Name"
AttributeNamespace="//medico.com">
           <saml:AttributeValue>*
       <resources>
<saml:Attribute AttributeName="documentURI"
AttributeNamespace="//medico.com">
           <saml:AttributeValue>//medico.com/record.*
<actions>
<saml:Action>read</saml:Action>
      <condition>
         <equal>
         <saml:AttributeDesignatorAttributeName="requestor" AttributeNamespace="…/"/>
         <saml:AttributeDesignator AttributeName="patientName"
                  AttributeNamespace="…/"/>
```

※説明上、終了タグは省略している

6.1.3 PKI/PMI との関連

SAML での「AuthenticationAuthority」は、認証したことを証明する Assertion を発行するエンティティであり、PKI で言う「Certification Authority」とは異なる。「Authentication Authority」が、実際の認証には「Certification Authority」の発行した証明書を用い PKI 認証を行うかもしれないという関連である。SAML の「Attribute Authority」と PKI の「Attribute Authority」の関係も同様である。

図 6-2 に SAML の概念図と X.509 PKI、PMI で使われる CA（公開鍵証明書を発行する認証局)、AA(属性証明書を発行する属性認証局) との関連を示す。



OASIS SAML Spec(http://www.oasis-open.org/committees/security/docs/cs-sstc-core-00.docより引用

System Entity: SAML により認証、認可を受ける対象

Credentials Collector: パスワードや鍵など認証に必要なもの

Authentication Authority: Authentication Assertion を発行する主体

Attribute Authority: Attribute Assertion を発行する主体

Policy Decision Point: ポリシーに基づき要求者にリソースに対するアクセス権限が与えられているかを判断する主体

Policy Enforcement Point: ポリシーに基づいて、アクセス制御を実施する

Authentication Assertion: 認証した事を示すためのデータ

Attribute Assertion: 属性情報を示すためのデータ

Authorization Decision Assertion: アクセス制御情報を示すためのデータ

Application Request: リソースに対するアクセス要求

図 6-3 SAML フレームワークと関連エンティティイメージ

6.2 XML ベースの表現言語

本節では、前節で述べた XACML 以外の XML ベースの属性、アクセス管理言語のいくつかを紹介する。

6.2.1 XrML

XrML(eXtensible rights Markup Language)は、ContentGuard によって、デジタル資源に関する権利と条件を記述する XML ベースの言語として 2001 年 11 月に開発された。

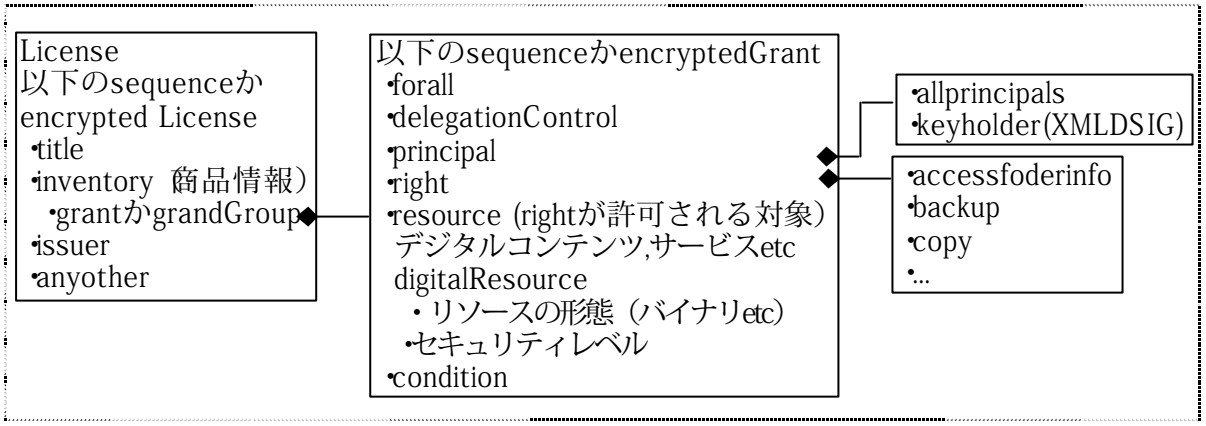


図 6-4 XrML 構造図

例：URL で指定されたデジタルデータを print できる

```
<license>
  <grant>
    <keyHolder>              許可されるサブジェクトの情報
      <info>
        <dsig:KeyValue> digital signature で規定されている署名鍵情報のデータ
    <cx:print/>              許可される操作（print）
    <cx:digitalWork>
      <cx:locator>           リソース位置情報
        <nonSecureIndirect URI="http://www.contentguard.com/sampleBook.spd"/>
```

※ 説明上、終了タグは省略している

6.2.2 ODRL

ODRL(Open Digital Rights Language)は、IPR Systems によって、無形・有形の資産に関する権利の表現、解析等を行うための言語として 2001 年 11 月に開発された。

XrML が資産と権利(Rights)が顧客に渡されてからの資産供給管理にフォーカスを当てているのに対し、ODRL は資産管理, 表示機能, コンテンツディストリビュータに渡すまでの範囲にフォーカスを当てている。

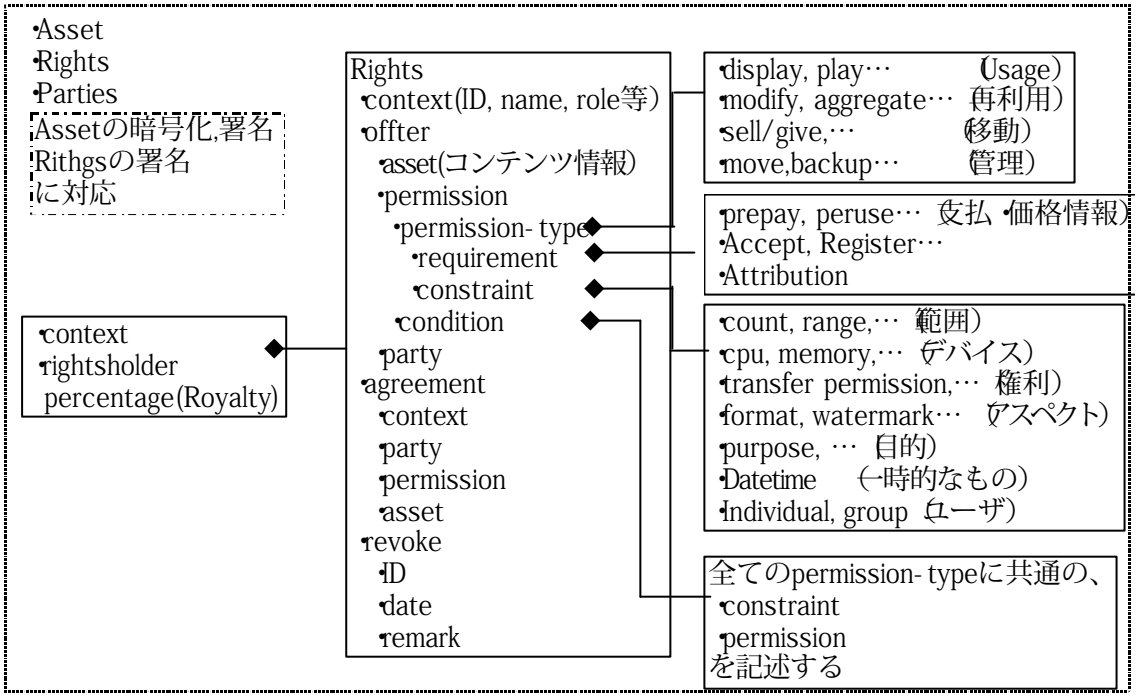


図 6-5 ODRL 構造図

6.2.3 XMCL

XMCL(the eXtensible Media Commerce Language)は、RealNetworks 社を中心に XMCL Initiative によって 2001 年 6 月に開発されたものであり、マルチメディアコンテンツに対する利用ルールを記述するフォーマットである。

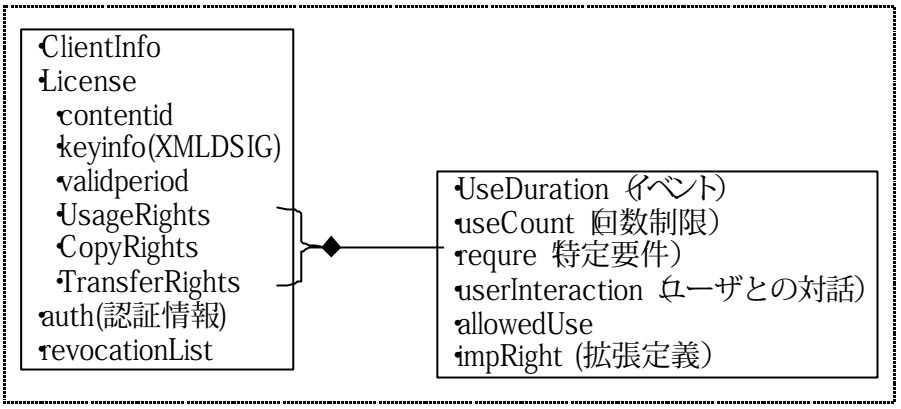


図 6-6 XMCL 構造図

6.3 属性証明書利用との比較

本節では、属性および許可されるアクセス・操作・権利・権限等の管理を、属性証明書を利用して表現する場合と XML ベースの言語を利用した場合の比較を行う。

表 6-1 に、簡単に表現イメージを掲載する。

表 6-1 属性アクセス制御の管理比較

	属性証明書利用例	XML ベースの言語例

属性	属性値フィールド利用
	□ServiceAuthenticationInformation フィールド
	service 名、識別名、認証に必要なアクセス者情報
	を格納する
	
	OID: id-aca 1
	SvceAuthInfo ::= SEQUENCE {
	service GeneralName,
	ident GeneralName,
	authInfo OCTET STRING OPTIONAL
	}
	
	□Access Identity フィールド
	service 名、識別名、認証情報を格納する
	OID: id-aca 2
	SvceAuthInfo ::= SEQUENCE {
	service GeneralName,
	ident GeneralName,
	authInfo OCTET STRING OPTIONAL
		■XACML
		<Target><subject> tag にアクセス者に関する情報、<condition>に条件を格納する
		
		■XrML
		<principal> tag にアクセス者に関する情報、<condition>に条件を格納する
		
		■ODRL
		<party>tag にアクセス者に関する情報、<condition>に条件を格納する
		
		■XMCL
		<auth>tag にアクセス者に関

	}
	
	□Charging Identity フィールド
	サービス利用に関する課金情報（課金先、課金のた
	めの ID）を格納する
	OID: id-aca 3
	IetfAttrSyntax ::= SEQUENCE {
	policyAuthority[0] GeneralNames OPTIONAL,
	values SEQUENCE OF CHOICE {
	octets OCTET STRING,
	oid OBJECT IDENTIFIER,
	string UTF8String
	}
	}
	
	□Group フィールド
	所有者が属するグループの情報を格納する
	OID: id-aca 4
	IetfAttrSyntax ::= SEQUENCE {
	policyAuthority[0] GeneralNames OPTIONAL,
	values SEQUENCE OF CHOICE {
	octets OCTET STRING,
	oid OBJECT IDENTIFIER,
	string UTF8String
	}
	}
		する認証情報を格納する

権限・ 権利	■属性値フィールド利用 1)-1ServiceAuthentication Information, Access Identity, Charging Identity に アクセス ID やキーを含める事で包含 1)-2 Group に対して暗黙の権利を割り当て る	■XACML <Target><resource>, <Target><action>, <Condition> tag に権限や権利、条件を記 述する

	2) Role 属性+Role extension(X.509 のみ：Role Specification Certificate を指定)を利用
	
	□ロールスペック証明書に役割を定義する
	　属性値には権限判断の基となるロールの属性を記述する
	
	□属性証明書（ロール割当証明書）は役割：ロール
	をユーザ（属性証明書保有者）に割り当てるためのものである。
	この証明書の属性値に Role Authority、RoleName を記述する事でロールスペック証明書との対応が取れる。
	X.509 仕様では、extension 領域に、
	「roleSpecCertIdentifier」としてロールスペック証明書の識別子(ロール名、発行者、シリアル番号、証明書格納場所)を指定する事もできる。
	
	role ATTRIBUTE ::= {
	WITH SYNTAX RoleSyntax
	ID id-at-role
	}
	RoleSyntax ::= SEQUENCE {
	roleAuthority [0] GeneralNames OPTIONAL,
	roleName [1] GeneralNames
	}
	roleSpecCertIdentifier Extension ::= {
	SYNTAX RoleSpecCertIdentifierSyntax
	IDENTIFIED
	BY {id-ce-roleSpecCertIdentifier}}
	
	RoleSpecCertIdentifierSyntax ::=
	SEQUENCE SIZE(1…MAX) OF
	RoleSpecCertIdentifier
		■XrML
		<right>, <condition> tag に権利、条件を記述する
		
		■ODRL
		<permission> tag に許可される権限を記述する
		
		■XMCL
		<UsageRights>
		<CopyRights>
		<TransferRights>
		tag にそれぞれ権利を記述する

	RoleSpecCertIdentifierSyntax ::=
	SEQUENCE {
	roleName [0] GeneralName,
	roleCertIssuer [1] GeneralName,
	roleCertSerialNumber [2]
	CertificateSerialNumber OPTIONAL,
	roleCertLocator [3]
	GeneralNames OPTIONAL
	}	

	3) 別で管理（アクセス制御テーブル利用等）	

XML ベースの言語を用いた表現にて管理を行う場合、以下の利点があると考える。

- アクセス対象が XML 文書の場合、データへアクセス情報を埋め込む事やリンクを張る事が可能である
- 属性とアクセス・操作・権利・権限等が同じスキームで記述できるため、アプリケーションが処理しやすい
- 後述する SAML をはじめとする Web サービスでのセキュリティ特に認証との相性が良い

6.4 関連 XML 技術

6.4.1 XML 署名

前節までで述べた一連の認証(行為)、属性、アクセス許可等を記述するにあたってはデータ作成者の署名付けを行う。XML データへの署名方法は、W3C にて定義されている。

XML 署名では、署名範囲、署名者情報、正規化手法、ダイジェスト計算情報、ダイジェスト値、署名アルゴリズム、署名値、検証鍵（公開鍵）情報を指定する。署名方式には、署名対象データを署名要素から参照の形でポイントする Detached 署名、署名対象データの中に署名要素が含まれる Enveloped 署名、署名要素に署名対象文書を含む Enveloping 署名の３パターンがあり、それぞれ用途に応じて使い分けられる。

6.4.2 XKMS

前節で述べた XML 署名、およびここでは記述しなかったが XML 暗号を利用する際に公開鍵系を用いる場合にその公開鍵を管理する仕様として、XKMS(XML Key Management Specification)がある。仕様は同じく W3C にて規定されている。

XKMS は、公開鍵登録のプロトコル XML Key Registration Service Specification (X-KRSS).と公開鍵に対する情報参照（検証も含む）のプロトコル XML Key Information Service Specification(X-KISS)から成る。

X-KRSS は、鍵管理機関(XKMS サーバ)に公開鍵を登録する際の仕様、X-KISS は XKMS サーバに対する鍵情報の問い合わせ、鍵情報の有効性確認を行うための仕様である。

XKMS 自体も XML Signature を用いて電文への署名付けが行われている。

本書の用語集

1. ＰＫＩ（Public Key Infrastructure）

公開鍵インフラ／公開鍵基盤とも呼ばれる。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。認証局（CA：Certification Authority）、登録局（RA：Registration Authority）、ディレクトリ（公開鍵証明書などを保管・開示する手段）、証明書有効性検証機関（VA：Validation Authority）、証明書を利用する利用者側のシステムなどが要素として含まれる。利用者側のシステムで行われる電子署名は、署名対象となる電子文書、あるいはそのハッシュ値を署名者の秘密鍵で暗号化する行為である。

2. GPKI、LGPKI（Government PKI, Local Government PKI）

各種申請・届出等の行政事務の電子化において基盤となる、政府及び自治体の PKI システム。

1999 年 12 月にミレニアム・プロジェクトが発表され、電子政府（行政事務の効率化、申請手続き軽減、情報公開、電子商取引促進）及び教育情報化に関する計画が明らかにされた。政府と民間の間のやり取りはインターネットが前提となっており、セキュリティを確かなものとすることは不可欠となっている。これら各種システムを安全に運用するために、GPKI 及び LGPKI の構築が進められている。

3. 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。

公開鍵暗号方式では「鍵ペア」と呼ばれる対となった二つの鍵が利用され、これらは公開鍵と秘密鍵と呼ばれる。公開鍵は、広く一般に開示する鍵で、検証者が署名検証を行う際に使用し、秘密鍵は署名を行う者自身が秘密に保持する鍵で、電子署名する際に使用する。

4. 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。

運用に際しては、認証局運用規定という、証明書発行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を作成する。それをＣＰＳ（Certification Practice Statement）とも呼び、一般に、利用者等に対して開示される。

5. リポジトリ(Repository)

加入者の証明書やＣＲＬおよびこれらに関連するその他の情報を保管しておき、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。

6. 登録局、登録機関

公開鍵証明書発行の申請者の本人性を確認し、主として登録業務を行う機関。

7. 発行局

電子証明書の作成･発行を主として発行業務を行う行う機関。

8. 公開鍵証明書

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。電子証明書、あるいは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があれば検出される形式となっており、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、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。

また、秘密鍵の危殆化、紛失等が生じた場合、証明書の所有者（署名者、または認証局）の指示に基づいて､有効期間内であっても証明書の効力を失わせることがあるが､これを証明書の失効と呼ぶ。

9. 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。ＣＲＬは、署名の検証者が、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場合に用いる。この他に､OCSP（Online Certificate Status Protocol）と呼ばれるプロトコルを使用してオンラインで失効確認する方法もある。

10. 相互認証

２つの認証機関（ＣＡ）が他方の認証機関を信頼することを証明し、安全に鍵情報を交換できるプロセス。

11. 電子署名法

2000 年 5 月に成立し､2001 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る旨が記述されている他､認証業務のうち一定要件を満たすものを特定認証業務と定義し認定を受けることができる、任意的な認定制度について記述されている。

12. 署名生成、署名検証

署名生成とは、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。紙の文書に押印する場合に相当する。

署名検証とは、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。

13. 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。

14. 属性証明局

公開鍵証明書の保有者が持つ属性を確認、審査し、属性証明書を発行する機関。AA(Attribute Authority）とも呼ぶ。

15. 属性証明書

公開鍵証明書が主に証明書利用者の特定に利用するのに対して、属性証明書はアクセス制御等に利用する。公開鍵証明書に記載された証明書所有者の名前によるアクセス制御も可能であるが、それのみならず、組織・団体における役職や役割などの属性情報によりアクセス制御を行うことも大いに考えられる。これら証明書所有者の属性情報を記載した証明書を、属性証明書と呼ぶ。これは、属性認証局（AA：Attribute Authority）により発行される。

16. 電子公証

電子認証と並んで、電子申請・企業間取引・電子文書長期保存等を支えるプラットフォームであり、一般的には、第三者（TTP：Trusted Third Party）による電子的記録の原本性を保証するサービス、として捉えられている。電子公証の意味合いは立場により解釈が異なる場合もあるが、共通的な認識としては、電子的記録の非改竄を保証し証拠能力を担保する為の一要素、非改ざんの保証は当事者ではなく第三者により行われる、電子公証の提供者が誰であるかは特に問わない、といった特徴を持つと言える。

本書の参考文献

[X.509-2000]

ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8:2001,
"Information technology - Open Systems Interconnection -
The Directory: Public-Key and Attribute Certificate Frameworks".

[RFC2437]

B. Kaliski and J. Staddon, "PKCS #1: RSA Cryptography
Specifications Version 2.0", RFC2437, October 1998.
http://www.ietf.org/rfc/rfc2437.txt

[RFC2560]

M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams,
"X.509 Internet Public Key Infrastructure -
Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.
http://www.ietf.org/rfc/rfc2560.txt

[RFC3280]

R. Housley, W. Polk, W. Ford and D. Solo,
"Internet X.509 Public Key Infrastructure
Certificate and Certificate Revocation List (CRL) Profile",
RFC3280, April 2002.
http://www.ietf.org/rfc/rfc3280.txt

[RFC3281]

S. Farrell and R. Housley, "An Internet Attribute Certificate
Profile for Authorization", RFC3281, April 2002.
http://www.ietf.org/rfc/rfc3281.txt

[RFC3369]

R. Housley, "Cryptographic Message Syntax (CMS)",
RFC3369, August 2002.
http://www.ietf.org/rfc/rfc3369.txt

[OCSPv2]

M.Myers, A. Malpani and D.Pinkas,
"X.509 Internet Public Key Infrastructure
Online Certificate Status Protocol, version 2
draft-ietf-pkix-ocspv2-ext-01.txt", December 2002.
http://www.ietf.org/internet-drafts/draft-ietf-pkix-ocspv2-ext-01.txt

[XACML]

http://www.oasis-open.org/committees/xacml/index.shtml

http://www.xmlconsortium.org/websv/kaisetsu/C11/content.html

[SAML]

http://www.oasis-open.org/committees/security/docs/cs-sstc-core-00.doc

http://www.xmlconsortium.org/websv/kaisetsu/C10/content.html

http://www.atmarkit.co.jp/fsecurity/rensai/webserv04/webserv01.html

[XrML]

http://www.contentguard.com/

[XMCL]

http://www.xmcl.org/

[ODRL]

http://odrl.net/

[関連]

「代理申請の制度的・技術的課題について
～電子申請における代理申請のあり方について」
代理申請に関する制度的・技術的課題研究会

「電子申請業務における X.509 属性証明書を用いた資格確認技術の開発」
日立ソフトウェアエンジニアリング株式会社

メンバーリスト

事務局

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

リーダー

山下　真　　富士通株式会社

TF1/2 メンバー（編集メンバー）

氏名	会社名
高村　昌興	株式会社ＮＴＴデータ
小黒　博昭	株式会社ＮＴＴデータ
今枝　直彦	日本電信電話株式会社
手塚　優	エントラストジャパン株式会社
松山　科子	ソニー株式会社
洲崎　誠一	株式会社日立製作所
笈川　光浩	株式会社日立製作所
金谷　延幸	株式会社富士通研究所
佐伯　正夫	三菱電機株式会社
坂上　勉	三菱電機株式会社
鍛治俊彦＊	株式会社日本電子貿易サービス

（注）＊はオブザーバー

SWG1 メンバー（参加メンバー）

氏名	会社名
荻原　利彦	NTT コミュニケーションズ株式会社
宍倉　勝仁	シャチハタ株式会社
中原　康	株式会社東芝
島田　毅	株式会社東芝
中村　逸一	株式会社ＮＴＴデータ
立石　広治	株式会社ＮＴＴデータ
河田　悦夫	株式会社エヌ・ティ・ティ・ドコモ
関野　公彦	株式会社エヌ・ティ・ティ・ドコモ
内海　雅俊	川鉄情報システム株式会社
鈴木　良信	コンピュータ･アソシエイツ株式会社
佐藤　正康	コンピュータ･アソシエイツ株式会社
雨宮　隆征	セイコーインスツルメンツ株式会社
米倉　昭利	（財）日本品質保証機構（JQA）
星野　　理	株式会社帝国データバンク
岩本　光恵	日本電気株式会社
本間　史夫	日本認証サービス株式会社
塚田　孝則	日立ソフトウェアエンジニアリング株式会社
手塚　　悟	株式会社日立製作所
西谷　研次	株式会社 UFJ 銀行
保田　昌宏	中央青山監査法人

平成 14 年度
ＥＣ技術基盤の相互運用性に関する調査研究事業
（取引相手先の属性認証技術等の調査）
属性認証の適用ガイドライン
平成 15 年３月発行

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

この資料は再生紙を使用しています。