コンピュータセキュリティ 1－7
（1998. 5. 29）

ディレクトリを用いた過去のデジタル証明書管理

村上 美幸　桑山 雅行　山崎 重一郎†　須賀 祐治†　荒木 啓二郎†‡

† (財) 九州システム情報技術研究所 (ISIT)
‡ 九州大学

概要

X.500ディレクトリサービスでのデジタル証明書の管理方法では、有効期間内の証明書を管理対象としているので、過去に受け取ったメッセージのデジタル署名を検証しようとすると、検証不可能性や否認可能性などの問題が発生する。これらの問題を解決するには、検証者が、過去のデジタル署名の検証に必要な、署名者の署名時のデジタル証明書とその廃棄情報を入手できなければならない。本稿では、その入手先となるディレクトリサービスを考え、ディレクトリで過去のデジタル証明書とそのデジタル証明書の廃棄情報を管理する方法を提案する。

Management system of past digital certificates using the Directory System

Miyuki Murakami, Masayuki Kuwayama, Shigeichiro Yamasaki†, Yuji Suga† and Keijiro Araki†‡

† Institute of Systems & Information Technologies/Kyushu

‡Department of Information Engineering, Faculty of Engineering Kyushu University

abstract

When we use X.500 directory service to verify a message which was digitaly signed in the past, we meet two problems: non-verifiability problem and non-repudiation problem. Non-verifiabilyty problem is the problem that verifiers can't verify validity of digitaly signed messages which were signed in the past because they can't get past certificate and certificate revocation information. Non-repudiation problem is the problem that signers of digitaly signed messages can repudiate their messages when the signer's digital certificate was revoked between signing time and verifiction time. To solve these problems, we propose new certificate revocation information management system which provide past certificates and certificate revocation information using directory service.

1　はじめに

インターネットの社会への浸透とともに、電子商取引などの様々な社会的活動がインターネット上で行われるようになってきている。しかし、インターネットは安全でない通信媒体であるため、そこでのセキュリティの確保や利用者のプライバシー保護が重要な課題である。それらの課題を解決する手段のひとつとして、デジタル認証という技術がある。

公開鍵暗号方式に基づくデジタル認証の技術

[1] を用いることでデジタル署名 [2] を実現できる。デジタル署名は、デジタル署名付きメッセージの生成者が確かに本人であることと、そのメッセージに対応するデジタル署名が生成されてからそのメッセージが変更されていないことを証明する。

デジタル署名付きメッセージを受け取ってすぐにデジタル署名を検証する場合には問題ないが、過去に受け取った古いデジタル署名付きメッセージのデジタル署名を検証する場合、次のような問題が発生する。その問題とは、デジタル署名付きメッセージを受け取った人が後になってデジタル署名を検証しようとしても、デジタル署名を検証するために必要な情報を入手できないので検証できないという検証不可能性と、検証者がデジタル署名付きのメッセージを受け取った後に署名者のデジタル証明書が廃棄されると、受け取ったメッセージのデジタル署名の有効性がなくなり、署名者がメッセージの発信や内容を否認することができるという否認可能性である。

これらの問題を解決するためには、過去のデジタル署名付きのメッセージのデジタル署名を検証するために必要な情報を検証者が入手できるようにすればよい。そこで筆者は、検証者がこれらの情報を入手できるようディレクトリサービス [3] を提案することで問題が解決すると考えた。

本稿では、そのディレクトリで管理する情報の種類、ディレクトリでのデータ構造、ディレクトリへの操作の方法とそのタイミングについて考察する。

2 デジタル署名

2.1 デジタル署名とは

公開鍵暗号 (public key cryptography) を利用することで、メッセージにデジタル署名を付けることができる。メッセージに対するデジタル署名は、署名者の秘密鍵を知っているものだけしか生成できない値で、メッセージの内容に依存する。

デジタル署名は 2 つの重要な機能を提供している。その 2 つとは、メッセージの生成者を証明することと、そのメッセージに対応するデジタル署名が生成されてから、そのメッセージが変更されていないことを証明することである。検証者はメッセージのデジタル署名を検証することでこれらのことを確かめることができる。

デジタル署名の検証を行うために必要な署名者の公開鍵は、認証局から署名者に対して発行されたデジタル証明書中に含まれる。デジタル署名の検証者は、このデジタル証明書を何らかの方法で入手することにより検証を行うことができる。

デジタル証明書の入手方法としては、ディレクトリサービスを用いる方法 [1][4]、FTP や HTTP を用いる方法 [5] などがある。

X.509 ディレクトリサービスでは、現在有効なデジタル証明書やその廃棄証明書リストなどを提供している。

2.2 デジタル署名の検証を行なう際の問題

X.509 ディレクトリサービスを用いる場合、デジタル署名付きメッセージを受け取ってすぐにデジタル署名を検証するときには問題ないが、過去に受け取った古いデジタル署名付きメッセージのデジタル署名を検証するときには、以下のような問題が発生する。

検証不可能性 デジタル署名付きメッセージを受け取った人が、後になってデジタル署名を検証しようとしてもデジタル署名を検証するために必要な情報を入手できないので、検証できない。デジタル署名の検証に必要な情報とは、署名者の署名時のデジタル証明書とそのデジタル証明書の廃棄情報のことである。

過去のデジタル署名付きメッセージの否認可能性 デジタル署名は否認不能性 (non-repudiation) を提供している。否認不能性とは、一旦デジタル署名付きのメッセージを送れば、署名者はそのメッ

セージを送った事実を否定できないということを意味する。

一度デジタル署名付きのメッセージを送ると、メッセージを受け取った人は、そのデジタル署名を検証することで、

- 署名者の秘密鍵を知っている者、すなわち署名者自身が作成したメッセージであること
- 署名者がデジタル署名を行ってから、そのメッセージは改竄されていないこと

を第三者に対しても立証できる。

しかし、デジタル署名付きの古いメッセージとなると否認可能となる場合がある。検証者がデジタル署名付きのメッセージを受け取った後にデジタル証明書が廃棄されると、メッセージのデジタル署名の有効性がなくなってしまう。それは、後日にデジタル署名を検証しようとしても、署名確認に必要な公開鍵を含んだデジタル証明書が廃棄されているからである。

3 ディレクトリを用いた過去のデジタル署名の検証システム

以上で報告した問題点を解決するためには、デジタル署名付きのメッセージがある時点で存在したことが確認でき、そのデジタル署名を検証するために必要な情報を検証者が入手できればよい。

デジタル署名付きのメッセージがある時点で存在したことは、確認できるとする。

そこで筆者は、デジタル署名付きメッセージを検証する際、検証者が認証局が運営する信頼できるデジタル証明書を公開するディレクトリ（以下、証明書公開ディレクトリと呼ぶ）にデジタル証明書の識別情報と署名時刻をもとに問い合わせを行い、署名者のデジタル証明書とその廃棄情報を入手できるようにすることで問題が解決すると考える。

3.1 全体構成

システムの全体構成を図1に示す。

ユーザからデジタル証明書の発行依頼がくると、認証局は新しく発行したデジタル証明書を証明書公開ディレクトリに格納する。

また、ユーザからデジタル証明書の廃棄依頼がくると、認証局はその廃棄情報を証明書公開ディレクトリに格納する。

検証者は署名者の識別情報と時刻を基に、ディレクトリに問い合わせを行う。ここでいう時刻は、信頼できる時刻である。また、署名者の識別情報とはそのデジタル証明書を発行した認証局のDistinguished Name(以下ではDNと呼ぶ)とデジタル証明書のシリアル番号である。

ディレクトリは、検証者からの要求に応じて、条件に当てはまるものがあればそのデジタル証明書を、条件に当てはまるものが無ければその旨を検証者に返す。

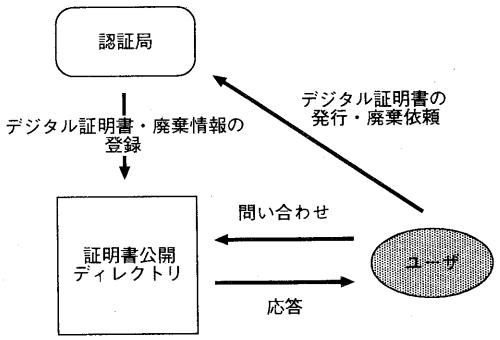

図 1: 全体構成

4 ディレクトリを用いたデジタル証明書管理

4.1 デジタル証明書とその廃棄情報を管理するディレクトリの構造

廃棄情報の管理 デジタル証明書の廃棄情報をディレクトリで管理する際、

- 認証局が発行した廃棄証明書リストを全て保管しておく方法
- デジタル証明書のエントリにそのデジタル証明書の有効期間を格納しておく方法

が考えられる。

前者の場合、検証者は署名者の識別情報と時刻を基にディレクトリに問い合わせを行い、デジタル証明書と廃棄証明書リストを返すことになる。しかし、廃棄したすべてのデジタル証明書のシリアル番号はデジタル証明書の有効期間中廃棄証明書リストに含まれるので、廃棄証明書リストのサイズは大きくなる。このため廃棄証明書リストの通信・検証・蓄積のコストは大きくなってしまう。

後者の場合、検証者は署名者の識別情報と時刻を基にディレクトリに問い合わせを行い、ディレクトリはデジタル証明書とその廃棄情報を返すことになる。ディレクトリは認証局が直接管理しており、信頼できるものであるので、デジタル証明書のエントリにそのデジタル証明書の有効期間を格納しておき、それによって検証者がディレクトリの検索を行えるようにしておくことで通信・検証・蓄積のコストは小さくなる。

そのため廃棄情報の管理には、デジタル証明書のエントリにそのデジタル証明書の有効期間を格納しておく方法を採用する。デジタル証明書の有効期間を管理するために新しく validFrom、validTo という属性型を定義する。validFrom にはデジタル証明書の発行時刻を、validTo にはデジタル証明書が有効でなくなった時刻、つまりデジタル証明書が廃棄されていなければ有効期限の時刻を、廃棄されたならば廃棄時刻を属性値として格納する。

デジタル証明書を格納するエントリ デジタル署名を検証する際のデジタル証明書の識別情報は、認証局の DN とデジタル証明書のシリアル番号である。そのため、図 2 のようにデジタル証明書のエントリは認証局の直接下位で管理し、個々のエントリの DN は格納するデジタル証明書のシリアル番号で識別する。シリアル番号は serialNumber という属性型の属性値として、デジタル証明書自身は userCertificate という属性型 [6] の属性値として格納する。

また、デジタル証明書のエントリのオブジェクトクラスとして、以下のように新しく oldCertificate を定義する。

オブジェクトクラス oldCertificate

必須属性 validFrom, validTo, serialNumber, userCertificate

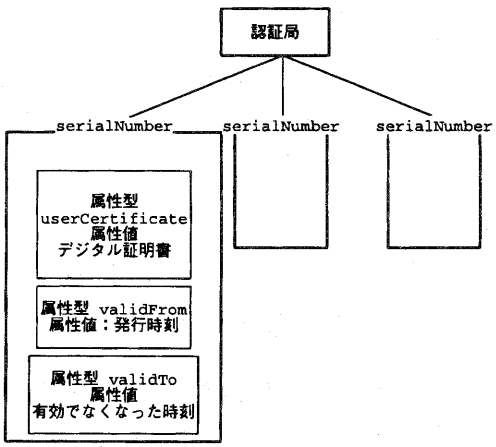

図 2: デジタル証明書のエントリ

4.2 デジタル証明書発行・廃棄時のディレクトリ操作

デジタル証明書発行時のディレクトリ操作 図 3 のように認証局は新規のデジタル証明書の発行依頼が来ると、依頼を確認しデジタル証明書の発行を行う。それと同時に、ディレクトリにオブジェクトクラスが oldCertificate であるエントリ

を新しく作成し、属性型 userCertificate の属性値として発行されたデジタル証明書を、属性型 validFrom の属性値として発行されたデジタル証明書の発行時刻を、属性型 ValidTo の属性値として有効期限を格納する。

デジタル証明書発行時に、属性型 validFrom の属性値としてデジタル証明書にあらかじめ設定された有効期限を格納しておくのは、そのデジタル証明書が有効期間内に一度も廃棄されなかった際にも属性型 validTo の属性値として、そのデジタル証明書が有効でなくなった時刻が格納されるようにするためである。

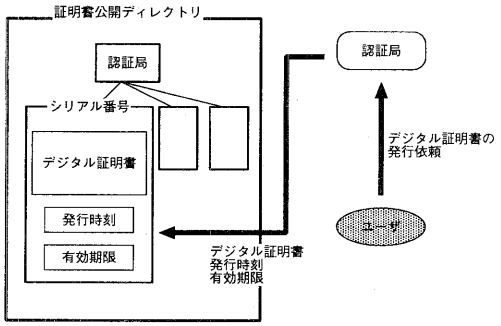

図 3: デジタル証明書発行時のディレクトリ操作

デジタル証明書廃棄時のディレクトリ操作 また、図 4 のように、認証局はデジタル証明書の廃棄依頼が来ると、依頼を確認しデジタル証明書の廃棄を行い、新しい廃棄証明書リストを発行する。それと同時に、ディレクトリの廃棄されたデジタル証明書のエントリの属性型 validTo の属性値を廃棄証明書リストに記載された廃棄時刻に書き換える。

4.3 検証者のディレクトリへの問い合わせ

検証者がデジタル署名の検証を行うには次のような条件でディレクトリを検索すればよい。

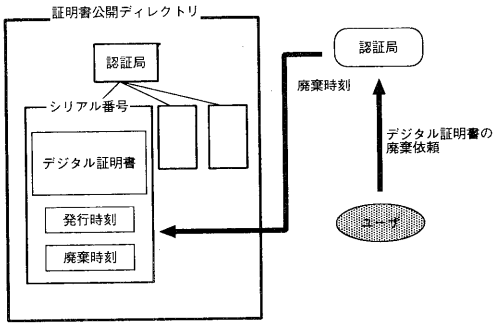

図 4: デジタル証明書廃棄時のディレクトリ操作

検索を開始する場所 検証するデジタル署名を行った秘密鍵に対応する公開鍵を含むデジタル証明書を発行した認証局のエントリの直接下位にある，シリアル番号により名前付けされるエントリ。

エントリの検索の条件 指定した時刻が validFrom よりも後であり、かつ、validTo よりも前であること。

取得したい属性値 userCertificate。

この検索により userCertificate が返されれば、指定した時刻での有効なデジタル証明書が入手できるので、それを用いてデジタル署名の検証を行うことができる。userCertificate が返されなければ、指定した時刻に有効なデジタル証明書が存在しなかったことになり、デジタル署名が正しくないことがわかる。

具体的には以下のようなコマンドおよびオプションを用いることで可能である。ここでは、ミシガン大学で開発されている LDAP ソフトウェア群を用い、1968 年 7 月 16 日の午前 8 時ちょうどに受け取った、日本の福岡県の福岡市の認証局で発行されたシリアル番号 29 番のデジタル証明書でデジタル署名されたデジタル署名付きメッセージを検証する場合の例を示す。

```
ldapsearch -b "serialNumber=29,o=CA,\
l=Fukuoka,st=Fukuoka,c=JP"  \
```

```
(&(validFrom <= 19680716080000Z)\
(validTo >= 19680716080000Z)) \
userCertificate
```

-b オプションが検索を開始する場所の指定である。次の行で指定しているのがエントリの検索の条件 [7] である。3行目で指定しているのが取得したい属性値である。

4.4 考察

第3章で挙げた問題は以下のように解決できる。

検証不可能性 認証局によって運営される、信頼できる証明書公開ディレクトリから、検証者が署名者の署名時のデジタル証明書とそのデジタル証明書の廃棄情報を入手できるようにした。検証者はこれらの情報を用いて、過去に受けとったデジタル署名付きメッセージの検証を行なうことができる。

否認可能性 署名者が故意にデジタル証明書を廃棄したとしても、過去のデジタル署名を否認することはできない。それは、検証者がデジタル署名付きのメッセージを受け取った後に署名者がデジタル証明書を廃棄したとしても、検証者はディレクトリに問い合わせることで、署名者のデジタル証明書が署名時に有効であったかどうか判断できるからである。

5 まとめ

本稿では、過去のデジタル署名の検証を行う際の問題点として、過去のデジタル署名付きメッセージの検証不可能性と否認可能性の問題について述べ、これらを解決する手段として、ディレクトリを用いたデジタル署名の検証システムを提案し、考察を行った。

参考文献

[1] ITU-T Recommendation X.509, "Information technology – Open Systems Interconnection – The Directory: Authentication framework," 1993.

[2] RSA Laboratories, "PKCS #7: Cryptographic Message Syntax Standard," 1993.

[3] ITU-T Recommendation X.500, "Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services," 1993.

[4] Tim Howes, Ph.D, Mark Smith, "LDAP:Programming Directory-Enabled Applications with Lightweight Directory Access Protocol," MACMILLAN TECHNICAL PUBLISHING, 1997.

[5] R.Housley, "Internet X.509 Public Key Infrastructure: Operational Protocols: FTP and HTTP," Internet Draft (draft-eitf-pkix-opp-ftp-http-03.txt , 1998.

[6] ITU-T Recommendation X.520, "Information technology – Open Systems Interconnection – The Directory: Selected attibute types," 1993.

[7] T. Howes, "A String Representation of LDAP Search Filters," RFC1960, 1996.