コンピュータセキュリティ　1－10
（1998．5．29）

# 公開鍵証明証の検証法式の考察[1]

榊原裕之[2]　吉武 淳[3]

三菱電機株式会社 情報技術総合研究所[4]
〒247-8501 鎌倉市大船 5-1-1

近年インターネット上での通信において、公開鍵暗号システムの利用が益々要求されている。公開鍵暗号システムを利用した安全な通信においては、認証局が発行した“公開鍵証明証”が必要となる。公開鍵証明証は、認証局が公開鍵とその持ち主の結びつきを認証局の公開鍵暗号の秘密鍵でデジタル署名を付加して証明したものである。公開鍵を安全に取得するために、公開鍵は“証明証パス”と共に検証された公開鍵証明証から取得するべきである。“証明証パス”はある認証局により署名されたエンティティの証明証と、0 個、又はそれ以上の数の、別の認証局が発行した認証局の証明証から構成される。加えて、証明証パス”上の証明証は失効していないことをチェックされなくてはならない。認証局が発行する CRL を失効のチェックに利用することは有効な手法である。

本稿では、CRL の運用が“証明証パス”の検証を複雑化することを示し、解決策を述べる。

# [1] Verification of public key certificates

[2] Hiroyuki Sakakibara　[3] Jun Yoshitake
[4] Information Technology R&D Center　Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura,247-8501, Japan

Recently requirement of public key cryptosystem has been increased on the Internet communication. A “public key certificate” issued by a Certification Authority(CA) is needed for secure communication with public key cryptosysytem.

A public key certificate is data structure which binds public key value to the public key owner digitally signed with the CA's private key. A public key of an entity should be safely obtained from the certifiicate verified through “certification path”. A certification path comprises a certificate of the entity signed by one CA, and zero or more addtional certificates of CAs signed by other CAs. Also, certificates on a certification path should be verified that they are not revoked .Using CRLs issued by CAs is an effective method of certificates revocation verification.

This paper describes that management of CRLs by CAs makes certification path complex and a solution to it.

## 1.はじめに

近年インターネット上の通信での安全性の確保が重要視されている。情報の秘匿、認証に対して、公開鍵暗号が利用されているが、公開鍵の取得は、認証局という信頼される第三者機関がその正当性をデジタル署名で保証した“証明証”[2] という形態で行われる。従って、証明証の正当性は公開鍵暗号の運用の安全性の基盤である。本稿では証明証の正当性を検証するにあたり、運用の形態によって処理が複雑になることを示す。

## 2.証明証について

### 2.1.公開鍵証明証

インターネットの普及に伴い情報の保護が重要視され、暗号技術が注目されている。中でも、鍵の配送と相手認証、メッセージ認証に利用できる公開鍵暗号は必須の技術である。公開鍵はその名の通り公開する情報であるが、悪意を持った人間が、他人の公開鍵を自分のものとして偽って公開したり、又、逆に自分の公開鍵を他人のものとして公開する恐れがある。この問題を防ぐために、公開鍵と持ち主の結びつきを、認証局と呼ばれる信頼のおける第三者機関がデジタル署名を用いて証明した“証明証”という形式で利用する。

### 2.2.認証局の構成

認証局の構成は図 1 に示す様なパターンが考えられる。図 1-(1)は SET[3]に代表される厳密な階層構造を持つものであり、認証局への証明証の発行は 1 つ上位の認証局が行い、完全なトップダウン型の証明証の発行形式となっている。失効管理については、各認証局が自身が発行した証明証についての CRL を管理する。完全なトップダウン型の管理形式のため、証明証の生成、失効管理の指針が明確であるが、RCA の証明証が失効すると、そのパス上にある証明証はすべて無効となる。図 1-(2)では、A、B2 つのドメインにおいて、各ドメインにおける認証局同士が、証明証を発行しているパターンを示している。この図では、CA2a と CA2b がお互いに証明証を出し合うことで、User_A と User_B が互いの証明証を検証することが可能である。

### 2.3.証明証の失効の確認

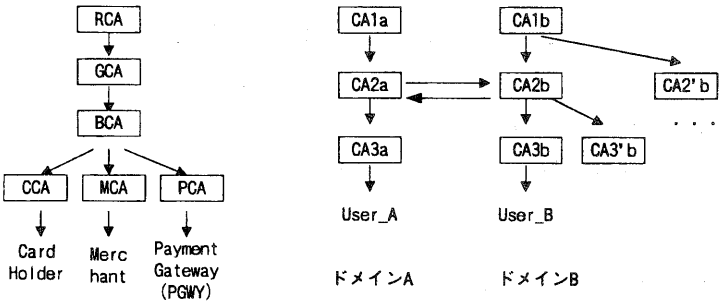


図 1

証明証の失効の確認方法としては、X.509 ver3[2][3]で定義されている CRL(Certificate Revocation List)を利用する方法や、認証局に失効状態をオンラインで直接問いあわせる方法がある。 CRL は、認証局が自身が発行している証明証で有効期限が切れる前に失効したものをリストにしたもので、認証局がデジタル署名を付加して配布する。検証者は CRL の中に対象の証明証が記載されていないことを確認する。

代表的なセキュア電子メールの規格である S/MIME[4]においては、CRL を採用しており、最近の X.509 ver3 をベースとした証明証の運用の提案でも CRL の使用を提唱しているので、本稿では、失効の確認は CRL を利用することを前提とする。

## 3.証明証の基本的な検証方法

### 3.1.証明証のパス

証明証のパスとは、ある証明証を検証しようとした場合、その証明証を検証するために必要な公開鍵が含まれる証明証を取得し、さらにその証明証を検証するために必要な公開鍵が含まれる証明証を取得するという処理を、自分が信頼している証明証にたどりつくまで繰り返したときに得られる順序性を持った証明証の並びである[2] 。図1の(1)、(2)の構成両方において、証明証のパスは発生する。証明証のパスの検証においては、FPKI[2]、SET[3]等 殆どがX.509 ver3 の検証方法を基盤としている。X.509 ver3 では、証明証のパスを検証する証明証から最上位の信頼する証明証まで決定し、そのパスに対して、最上位の信頼する証明証から、トップダウンで検証を行う。以下にトップダウン型の検証例を挙げる。図2-(1)は、CA_0 がルートの認証局（CA）であり、各認証局が一つ下の認証局に証明証を発行しており、CA_2 は EndEntity に証明証を発行する。CA_0 から EndEntity までの証明証のパスは、図 2-(2)に示されており、Cert_CA0, Cert_CA1, Cert_CA2, Cert_EE となる。

1. Cert_CA0 の正当性を確認後、含まれる公開鍵を用いて、Cert_CA1 の署名を検証する。
2. Cert_CA1 の正当性を確認後、含まれる公開鍵を用いて、Cert_CA2 の署名を検証する。
3. Cert_CA2 の正当性を確認後、含まれる公開鍵を用いて Cert_EE の署名を検証する。

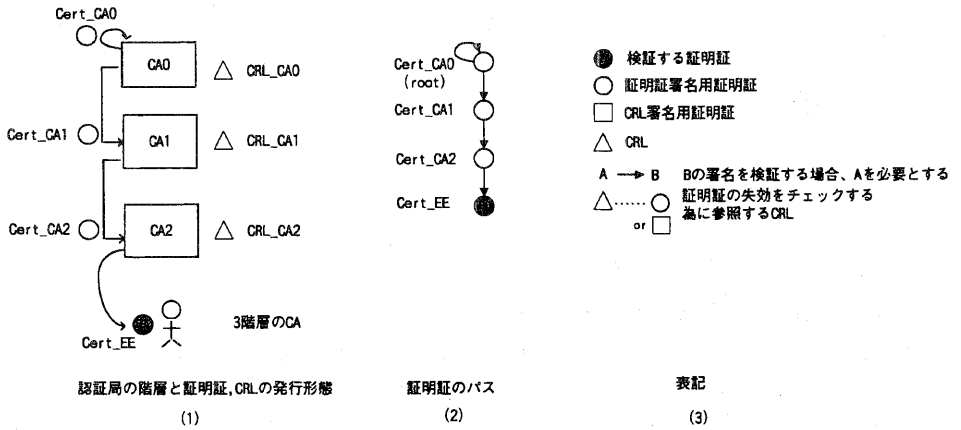


図 2

という順序で行われる。途中、検証している証明証とその署名を検証するための上位認証局の証明証間の整合性も随時チェックする必要がある。例えば、Cert_CA1 の Issuer が、Cert_CA0 の Subject になっているか、Cert_CA1 の有効期限は、Cert_CA0 の有効期限の範囲に収まっているか等をチェックする。なお、証明証のパスの信頼性は、Cert_CA0 の正当性に依存するので、Cert_CA0 の正しさは、別方法でチェックする必要がある。一般的な手法としては、CA0 の公開鍵のハッシュ値を信頼できる別ルートで入手し、Cert_CA0 内の公開鍵のハッシュ値を自分で計算した値と比較し、一致を確認する方法がある。すでに、検証済みの Cert_CA0、Cert_CA1、Cert_CA2 を保持している場合は、Cert_CA0 から Cert_EE のパスを決定後、Cert_CA0, Cert_CA1, Cert_CA2 の検証は省略し、Cert_CA2 と Cert_EE の間で、項目の整合性と署名のチェックを行えばよい。

### 3.2.CRL を利用した失効の確認

証明証のパスの検証において重要なチェック事項として、3.1 で述べたチェックの他に、パス上のすべての証明証について失効していないか CRL を利用して確認することが挙げられる。

CRL の発行形態は認証局の運用ポリシーに左右されるが、本節では SET における CRL の運用を例に挙げる。SET においては各認証局が、自分が発行した証明証に関する CRL を発行するので、各認証局につき、1 つの CRL が存在する。X.509 ver3 においては、証明証が保証する鍵の用途を示す keyUsage という extension が存在し、証明証署名用と CRL 署名用を別々又は同時に指定することが可能である。

各認証局において、証明証署名用の証明証と CRL 署名用の証明証が同じであるとき、検証すべきパス上の証明証と CRL の構成は、図 3-(1)で示される。各 CRL は、認証局が証明証を発行する時に使用する秘密鍵を CRL の署名にも利用するので、証明証のパスは 1 本であり、このようなパス構成になる。各記号の表記の意味は図 2-(3)に示してある。又、証明証署名用の証明証と CRL 署名用の証明証を分けて発行している場合は、図 3-(2) に示された構成になる。この場合は、証明証のパス自体は 1 本であるが、CRL 署名用証明証の署名の検証が必要であり、検証個所が図 3-(1)に比べて増えるので、証明証のパス全体の検証に時間がかかる。

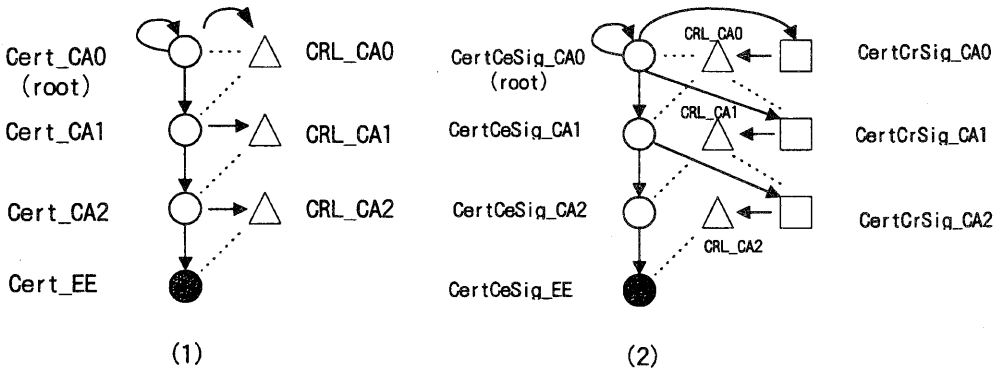


図 3

## 4.証明証のパスの構成の複雑化

### 4.1. 証明証署名用証明証と CRL 署名用証明証が同一の場合

認証局の運用の形態により証明証のパスが複雑化することがある[1]。例えば、図 3-(1)の様に、証明証署名用と CRL 署名用の証明証が同じであった場合、以下の様な運用が実施されたとする。

証明証：Cert_CA0：notAfter = 98/04/30, Cert_CA1：notAfter=98/04/15, Cert_CA2：notAfter=98/04/01,
Cert_EE ： notAfter = 98/03/31 が発行されている状態で、

1. 98/03/29 現在、CA0 が、次の時期の運営用の鍵対を新たに生成し、notBefore=98/03/29 である証明証 Cert_CA0'を発行した。又、CRL は新しい鍵で署名し、thisUpdate=98/03/29 である CRL_CA0' として発行した。
2. CA1 は Cert_CA0'の発行に対し、次の時期の運営用鍵対を新たに生成し、CA0 の新しい秘密鍵で新しい証明証 Cert_CA1'を生成してもらう。notBefore は 98/03/30 とする。又、CRL は新しい鍵で署名し、thisUpdate=98/03/30 である CRL_CA1'として発行した。

この場合、98/03/30 の時点で Cert_EE の検証を行った場合 Cert_CA0,Cert_CA1, Cert_CA2, Cert_EE は検証を行った場合有効期限内にあり、CRL_CA0', CRL_CA1', CRL_CA2 も有効期限内なので、図 4-(1)のようなパスの構成になる。Cert_EE に関する Root までのパスと、CRL_CA1', CRL_CA0'を検証するために必要な Cert_CA1', Cert_CA0'に関するパスが異なるという状態が生じる。両者とも Root が異なるので、Cert_CA0,

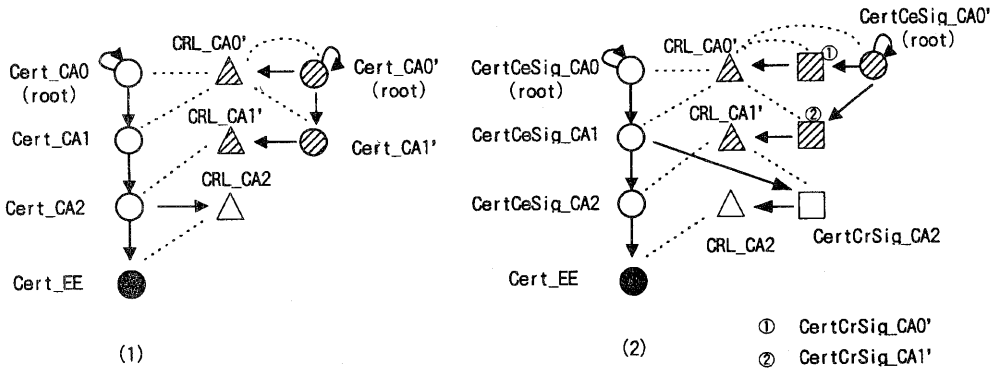


図 4

Cert_CA0'両方の正当性を確認する必要がある。

**4.2.証明証署名用証明証と CRL 署名用証明証が別の場合**

証明証署名用証明証と CRL 署名用証明証が別れている場合の例を挙げると、図 3-(2)の形態で、以下の運用がなされたとする。

1. CA0 は CertCeSig_CA0 が有効な期間に鍵対を新たに生成し、証明証署名用証明証 CertCeSig_CA0' を発行した。又、新たな鍵対を生成し、CRL 署名用証明証 CertCrSig_CA0'を生成し、CRL はこれと対の新しい鍵で署名し、CRL_CA0'として発行した。
2. CA1 の CertCrSig_CA1 に対する秘密鍵が漏洩したため、CA 1 は鍵対を新たに生成し、CRL 署名用証明証 CertCrSig_CA1'を CA0 に発行してもらった。この場合、CA0 は CertCeSig_CA0'と対の鍵で署名を行った。CRL は新しい鍵で署名し、CRL_CA1'として発行した。

この場合は、証明証のパスは図 4-(2)の形態になり、図 4-(1)よりも複雑化している。

**5.証明証パスの複雑化への対処**

以上の様に、CRL を検証するためにさらに証明証のパスが存在することが、証明証のパス全体を複雑化する要因である。 CRL の署名の検証に利用する証明証の世代が、検証したい証明証のパス上の世代とずれれば、その分検証しなくてはいけないパスが増える。従って、CRL の署名の検証箇所、或いは CRL の署名を検証するための証明証の検証箇所を如何に減らせるかが焦点となる。以下に解決手法を挙げる。

（1）証明証署名用証明証と CRL 署名用を兼用とする

複雑さの違いは図 4 に示してある通りで、両用途を兼用にした方が軽減する。

（2）X.509 ver3 に記載されている手法

X.509 ver3 においては、解決方法として、CRL の個数を減らす方法を提示している。1 つのエンティティが、複数の CA の証明証の失効状態を CRL として発行することを例として挙げている。CA の証明証が End Entity の証明証に比べて頻繁に失効することはないと予想されるので、有効であると思われる。

（3）有効 CRL 情報（ Valid CRL Information (VCI) )の利用

SET においては、Brand CA が、その Brand に関連する CRL において最新のものを特定するために Brand CRL Identifier(BCI)という情報を発行している。BCI には、Brand CA 以下の認証局と Root が発行した CRL のうち、最も新しいものを識別する情報が含まれている。BCI は最新の CRL を特定するだけの機能だけを持った情報である。又、最近の PKIX のドラフトでは同じ機能を持った情報が CRL List Attribute として定義されている。本稿では、これらの機能を拡張し、CRL の署名に関する証明証パスを短縮する“有効 CRL 情

報：Valid CRL Information(VCI)" を提案する。"CRL 情報管理機関：CRL Information Management Authority( CIMA)" という機関を仮定し、自身が置かれているドメイン内の認証局が発行している CRL 全てを常に把握し、署名の検証を含めて有効性を確認する。その上で、それら CRL の識別子とハッシュ値をリストアップして署名をつけたものを VCI と定義する。 VCI は CIMA の証明証 Cert_CIMA と対の鍵で署名されている。この様子を図 5 に示す。以下に、VCI の利用手順を示す。

1. 検証者は VCI を取得し、 Cert_CIMA を利用して署名を検証する。
2. VCI 内部に記載される CRL の識別子を利用して必要な CRL を収集する。
3. 各 CRL のハッシュ値が VCI 内部に存在するので、この値と取得した CRL のハッシュ値を突き合わせて一致を確認する。一致しなかったら、取得した CRL は無効である。
4. 各 CRL の署名は CIMA により検証済みなので、省略することができれる。従って CRL の署名検証用の証明証パスの発生の問題は解決される。

しかし、この方法でも Cert_CIMA の署名を検証するための証明証のパスが発生する。従って、Cert_CIMA は self-sign にするか、Root が直接発行する等、パスを短縮化する必要がある。SET の BCI の発行の様に、Root に近い認証局、或いは Root が CIMA を兼務しても良い。その場合は、VCI を署名するための鍵を、証明証署名用証明証又は CRL 署名用証明証に対応したものと兼用することで、これらのパスと VCI を検証するためのパスを共通化する。

CIMA が、CRL を検証する代わりに各 CRL 署名用証明証を検証し、Issuer, SerialNumber, ハッシュ値をリストアップして署名を付けて "有効 CRL 署名用証明証情報 Valid CRLSignCert Info" として発行し、VCI 同様の運用をすれば CRL の署名におけるパスの短縮化に貢献する。

## 6.おわりに

CRL の利用はオープンな環境では有効であるが、証明証のパスが複雑化するという問題がある。本稿では解決方法をいくつか示し、VCI を提案した。今後は Cert_CIMA の署名に関するパスの短縮化、Cert_CIMA の失効管理等が課題である。

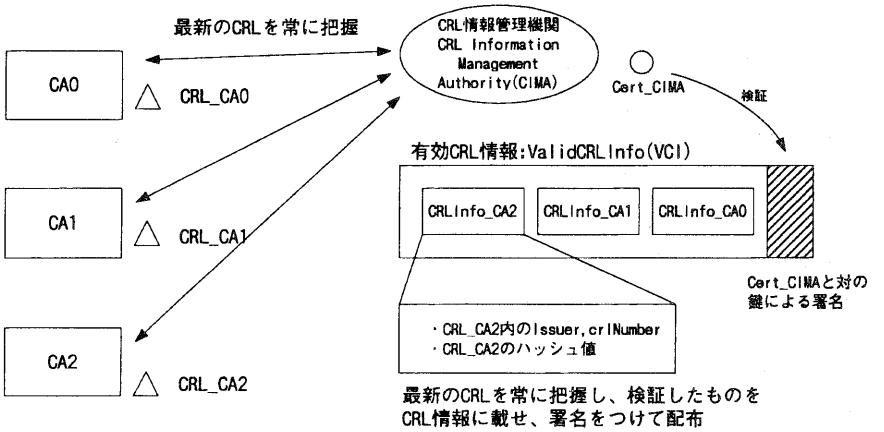


図 5

[参考文献] [1] 榊原、吉武： SET 他の公開鍵証明証の検証方式の考察 情報処理学会, 第 55 回全国大会講演論文集 5M-07,1997

[2] Federal Public Key Infrastructure(PKI) X.509 Certificate and CRL Extension Profile, NIST PKI-TWG March 9, 1998

[3] Secure Electronic Transaction Specification Book2:Programmer's Guide, Ver1.0, May 31, 1997

[4] S/MIME Version2 Certificate Handling, RFC 2312, March 1998