第 11 部

公開鍵証明書を用いた利用者認証技術

第 1 章

はじめに

CA (Certification Authority) 技術は 通信相手が名乗ったときにそれが正しいかどうかを確認するための技術 (認証技術) の 1 つで、公開鍵暗号の公開鍵とユーザ識別子の対応を第三者が証明することで相手の正当性を保証する技術である。一度証明された公開鍵は証明書として安全かつ広範囲に配布することが可能となるため、CA 技術は認証技術の中でも広域分散環境に適した技術として注目を浴びている。

IETF で検討されてきた PKIX(Public Key Infrastructure X.509) では、インターネット上で CA を利用していくために必要な基盤技術をここ 1 年で次々と RFC 化している [65][13][93][35] [66][24]。また、インターネット上のユーザを対象とした電子決済への応用や S/MIME[9][45][46] などの電子メールのセキュリティ強化への応用、SSL[10](Secure Sockets Layer), TLS[43](Transport Layer Security) を用いた WWW のセキュリティ強化への応用が広まっている。

われわれ moCA(members oriented CA) 分科会は、CA 技術の利用を進める上で CA 運用のノウハウ収集が重要であると考え、昨年度より WIDE メンバを対象とした moCA(members only CA) を立ち上げ、運用面に焦点をあてた実験を行なってきた [129]。CA の運用にあたって特に検討すべき点として一般に以下が挙げられる。

- 立ち上げのための検討
  - 証明書に含めるユーザ識別子の検討
  - 証明書の発行手続きの検討
  - ...
- 運用維持のための検討
  - CA 証明書、ユーザの証明書の有効期限切れ対応の検討
  - CA、ユーザの鍵変更対応の検討
  - ...

昨年度は、おもに立ち上げのための検討を中心に実験を進めてきた [129][150] ため、今年度は昨年度立ち上げた CA の運用維持の検討に着目して実験を行なった。本報告では、

まず CA の運用を維持しようとしたときに解決しなければならない問題についてまとめた後、今年度行なった個々の実験の概要、結果および考察について述べる。

第 2 章

CA 運用実験

2.1 実験の体系づけ

今年度は、CA の運用を維持しようとする際の課題として、証明書の有効期限切れや鍵対変更に対する手続き、また、CA を運用する組織の運営状況の変化への対応をとりあげ、いくつかの実験を行なった。

2.1.1 証明書の有効期限切れおよび鍵対変更

まず、証明書の有効期限切れに関しては、誰の証明書かによって以下の 2 種類があり、証明書の有効期限を延長するにあたって影響を及ぼす範囲が異なる。

- ユーザやサーバといったエンドエンティティの証明書の有効期限切れ

  エンドエンティティ自身および通信相手のエンドエンティティに影響がある。

- CA の証明書の有効期限切れ

  CA 自身およびその CA が証明対象とする全てのエンドエンティティ(あるいは CA)に影響がある。

次に、鍵対の変更に関しては、証明書の期限切れと同様、誰の鍵かによって以下の 2 種類があり、証明書の再発行にあたって影響を及ぼす範囲が異なる。

- エンドエンティティの鍵対の変更
- CA の鍵対の変更

鍵対の変更に関してもっともやっかいなのは、ルート CA の場合である。ルート CA は、証明書を自己署名により作成しているため、理論上誰でもルート CA を構築することが可能である。実際の運用では、信頼点として機能させるために、ルート CA 証明書の配布にオンライン、オフラインを含めて複数の方法を提供するなどしてルート CA のなりすましを防いでいる。ルート CA の鍵対の変更をする場合には、できれば新しい鍵と古い鍵とを

関係付けて、今までに築いてきたルート CA に対する信用を引き継げるような手続きを考える必要がある。そのためには、信用しているルート CA の鍵対が変更したことを、新しい鍵と古い鍵の両方を使って誰にでも確認できる手段が必要となる。しかし、鍵対の変更理由によっては古い秘密鍵がルート CA 以外の手に渡っている場合があり、その場合には新規の立ち上げ手順に沿ってルート CA を構築しなおす必要がある。

また、ルート CA 以外の場合でも、古い鍵が使える場合と使えない場合とでは、証明書の再発行に際しての申請者の確認手続きに違いが出てくる。鍵対の変更理由を以下に示す。

- 証明書の有効期限切れに合わせた鍵対の変更 ··· 古い秘密鍵はあり、使える
- 秘密鍵の紛失による鍵対の変更 ··· 古い秘密鍵はないため、使えない
- 秘密鍵の盗難による鍵対の変更 ··· 古い秘密鍵はあるが、所有者以外の手に渡っていて使えない
- 鍵の寿命による鍵対の変更 ··· 古い鍵はあるが、公開鍵から秘密鍵が逆計算されているかも知れず、秘密鍵が所有者以外の手に渡っている可能性があり使えない

このような観点から、証明書の有効期限切れや鍵対の変更のケースを分類し、表 2.1の中で◯印をつけたケースについて作業手順およびアナウンス方法を検討し、実験を行なった。

2.1.2 CA を運用する組織の運営状況の変化

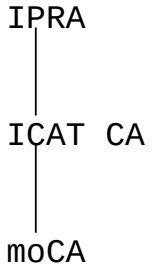
ある証明書が鍵対の変更を伴わずに再発行される理由として、証明書を発行する CA が変わるから、というケースが考えられる。具体的には、例えば以下のケースがある。

- CA を運用する 2 つの組織が合併し、片方の CA がもう一方の CA に吸収された
- CA を運用する組織が運営を終了し、別の CA へ移らざるを得なくなった
- CA を運用する組織が運営方針を大きく変えたため、別の CA へ移った

これらは、技術的な要因ではなく、現実の組織の運営状況の変化により影響を受けるケースであり、CA の運用にとって重要な、ポリシの変更を伴う可能性もある。

本分科会で昨年度から実験運用している CA, moCA(members only CA) は、IPRA(Internet Policy Registration Authority) をルート CA とする系列に位置づけてきた。厳密には IPRA の下の ICAT CA の下にあった。(図 2.1参照。)

[1]昨年度、IPRA というルート CA の系列にいたときに、実際に証明書の期限延長があり、その時点で実験は行なっている [129]。



	運営組織
IPRA	Internet Society
ICAT CA	ICAT(認証実用化実験協議会)
moCA	WIDEプロジェクト

図 2.1: 1998 年度当初の CA の位置づけ

今年度に入り、ICAT CA の運営母体である ICAT(認証実用化実験協議会) の活動が 9 月末で終了することが決定し、ICAT CA の運用継続は事実上難しくなるとわかった。現実に CA を運用する組織の運営状況に変化がおこりつつあったわけである。

そこで、ICAT の活動が終了する前に、独自のルート CA を構築し、moCA を新しいルート CA に移動する CA 系列移動実験を行なうことになった。表 2.1でいうと、(12) のケースにあてはまる。

2.1.3　今年度の実験

今年度行なった実験と表 2.1との対応関係を表 2.2に示す。1 回の実験でいくつかのケースを含んでいる場合があるが、実際の運用でもこのようにいくつかのケースが複合的に発生すると考えられる。

1998 年 9 月以降の実験では、SOI(School of Internet) 分科会が実験運用している SOI CA と共同で実験を行なった。節 2.2からは、個々の実験について述べる。

2.1.4　実験のシステム構成

昨年度と同様、実験で利用したのは既存のツールで、これらを組み合わせて図 2.2に示す構成をとった。

WIDE メンバであるユーザには、実験のアナウンスごとに、Web ブラウザへ新しい証明書を登録するなど、手順に沿った作業を依頼し、その後 SSL 対応 WWW サーバにアクセスできるかどうか、また、暗号メールのやりとりができるかどうかの確認を依頼した。

2.2　証明書更新実験

昨年度の実験で発行した個人証明書の有効期限および moCA 証明書の有効期限が 6 月 10 日に切れることをきっかけに、鍵対を変更しないで証明書の有効期限のみを変える証明

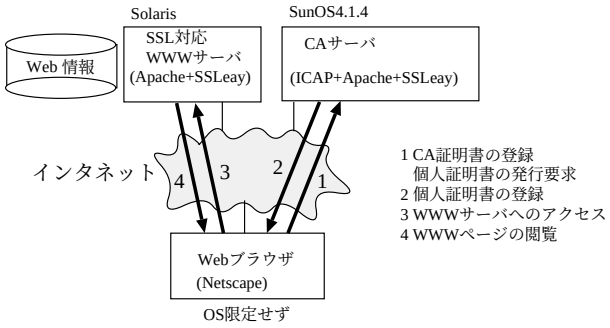


図 2.2: 実験のシステム構成

書更新実験を行なうことにした。

この実験の目的は、以下の 2 点である。

- Netscape ブラウザが、鍵対を変更しない場合の証明書の更新に対応できるのかを試すこと
- CA の証明書の更新をユーザに徹底してもらうための手続きについて考察すること

2.2.1 実験概要

Netscape ブラウザには、秘密鍵とそれに対応する証明書が対となって証明書データベースに保存されている。新しい証明書を登録しようとしたときには、証明書に対応する秘密鍵が証明書データベースにあるかどうかを確認する処理を行なっている。Netscape Navigator 4.0x では、鍵対を変えずに、有効期限とシリアル番号のみを変えた証明書を登録しなおすと、証明書が更新されたものとして、今まで使ってきた秘密鍵との対応づけを自動的に行なう機能がついた。

前回の実験で発行した個人証明書の有効期限は、4ヵ月程度と短く、鍵の寿命はまだ十分残っていると判断し、個人証明書の有効期限を延長して Navigator の機能を試すことにした。また、moCA の鍵については 1 年以上変更していなかったが、鍵の寿命はまだ残っていると判断し、鍵対を変更しないで証明書更新をすることにした。

(1) 準備作業

この実験では、証明書の有効期限が切れると困るユーザのことを考慮し、証明書の有効期限が切れる前に、証明書更新作業を行なえるようにする必要があった。そこで、次のような方針とした。

- 新旧の moCA の証明書の有効期限が一部重なるように証明書を発行する

- 個人証明書の有効期限が切れる前に更新を受け付ける

具体的には、以下のような手順に沿って作業を行なった。

1. moCA の管理者が、上位 CA である ICAT CA の管理者に対して、moCA 証明書の再署名を依頼する
2. 新しい moCA 証明書を受け取る
3. ユーザが新しい moCA 証明書を入手するための WWW ページを更新する
4. ユーザの行なうべき作業手順をまとめた WWW ページを用意する

さらに、ユーザへのプロモーション用の SSL 対応 WWW サーバの準備を行なうため、SSL 対応 WWW サーバの証明書について更新作業を行ない、WWW サーバ上で新しい moCA 証明書の設定作業を行なった。

(2) アナウンス

ユーザには、証明書の有効期限が切れるため次の作業を依頼した。

- 更新された moCA 証明書をブラウザへ登録しなおすこと
- 個人証明書更新を moCA 管理者に依頼すること

個人証明書の更新作業は、moCA 管理者が手動で行ない、電子メールの到達性を利用して新しい個人証明書の入手先を本人に通知するという方法をとった。

2.2.2　実験結果および考察

表 2.3に示すように、moCA 証明書の再署名依頼を 6 月 2 日に開始した後、アナウンスを開始したのは証明書の期限が切れる 2 日前の、6 月 8 日となった。
(1) 実験結果

前回 (1998 年 3 月) の実験参加者約 60 名のうち、証明書の有効期限が切れる前に更新作業を行なった人は 15 名、証明書の有効期限が切れた後に更新作業を行なった人は、約 10 名であった。

(2) 考察

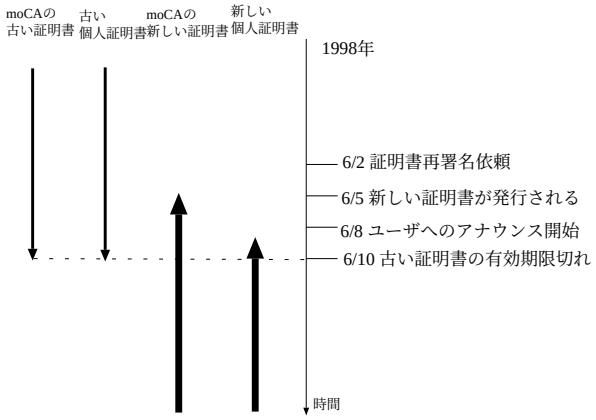


図 2.3: 証明書更新実験の経過

- 個人証明書の更新と Messenger 4.0x の機能の関係について

  個人証明書の更新後、証明書データベースから古い証明書を削除すると、古い証明書を使って暗号化された電子メールが読めなくなるということがわかった。これは、更新された証明書のシリアル番号が変わっているため、証明書をシリアル番号で参照する S/MIME では更新された証明書と古い証明書は別と判断しているためであろう。今後の実験で個人証明書を更新する際には、「古い証明書を削除すると古いメールが読めなくなる」という点をあらかじめ説明する必要がある。

- CA の証明書の更新について

  CA 証明書の更新作業を有効期限が切れる前にして、新旧の CA 証明書の有効期限を一部重なるようにした点は、個人証明書の更新と同時に新しい証明書を利用できるため、利便性からみてよい方法であったと考えられる。

- 実験アナウンスについて

  今回のアナウンスでは、個人証明書の更新を重点的に通知しようとしたため、アナウンスが個人証明書を持っているユーザ向けに閉じた内容であった。しかし、個人証明書を持っていないユーザであっても、SSL のサーバ認証や暗号化のみを目的とした WWW サーバにアクセスする場合がある。個人証明書を持っていないユーザにも CA 証明書をブラウザに登録することだけはアナウンスすべきであった。

- Messenger 4.05 の署名つきメールの機能について

  Messenger4.05 では署名つきメールを作成すると、ルート CA を含む全ての CA 証明書が自動的に添付される。実験アナウンスを署名つきメールにより行なったところ、そのメールを Messenger で読んだ人には、moCA 証明書が自動的に証明書データベー

スに登録されることがわかった[2]。CA 証明書を WWW サーバから入手する場合と手順が若干異なるため、Messenger のユーザには混乱が生じ、ツールに応じて手続きの説明を変える工夫が必要であるとわかった。

2.3　CA 系列移動実験

moCA の上位 CA は ICAT CA である。ICAT CA の運営母体である ICAT(認証実用化実験協議会) の活動が 9 月末で終了することが決定し、ICAT CA の運用継続は事実上難しくなった。そこで、ICAT の活動が終了する前に独自のルート CA を構築し、moCA を新しいルート CA に移動する CA 系列移動実験を行なった。

この実験の目的は、以下である。

- 上位 CA の変更および CA の証明書の変更をユーザに周知徹底するための手続きについて考察すること
- 上位 CA が変わることについて、CA にどのような影響があるのかを考察すること

2.3.1　実験概要

moCA の運用継続のためには以下の選択肢が考えられた。

- 既存の他の CA の下位 CA となること
- 新しくルート CA を立ち上げてその下位 CA となること
- moCA がルート CA になること

moCA は実験用の CA である。moCA の上位 CA である ICAT CA および IPRA は、いずれも実験用 CA を認めるポリシであった。実験用 CA を認める CA として広く知られたものがほとんどない状況の中で、まず 1 つの方法として、IPRA の直下に位置づけることが考えられた。しかし、以前より IPRA が 2000 年以降運用継続するかは未定と通知されていたこと[3]や、証明書の申請から発行までにかかる時間に予測がつかなかったことから、この案は採用しなかった。

また、moCA の鍵はオンラインで管理されていることから、実用上オフライン管理が望ましいとされるルート CA になるのは適さないと判断した。そこで、実験目的のルート CA を新しく立ち上げてオフライン管理とし、moCA はその下位 CA として運用継続することにした。このルート CA を WIDE ROOT CA と名付けることにして、図 2.4に示す系列移

[2]ただし、自動登録された証明書を信用するかどうかの設定は別途行なう必要がある。

[3]そして、本当に 1998 年 12 月以降は運用しない (Internet Society によるサポートが打ち切りになる) ことになってしまった。

動を行なうことになった。

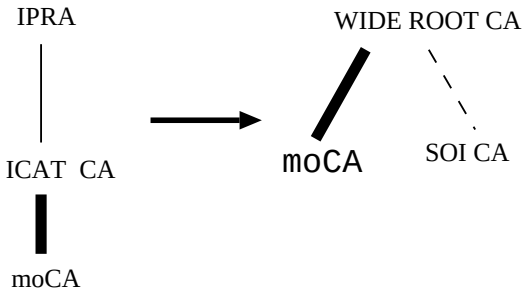


図 2.4: CA 系列移動実験

(1) 準備作業

この実験では、WIDE ROOT CA の立ち上げ作業とともに moCA 証明書の更新作業が必要となるが、技術的な問題による証明書更新ではないため、ユーザの作業はできるだけ少なくなるようにしたいと考えた。したがって、ユーザの個人証明書に変更がおこらなくてすむように、moCA の鍵対については変更しない方針とした。また、ICAT CA から発行された moCA 証明書の有効期限が 9 月 30 日と迫っていたため、その期限前にアナウンスを開始するという目標を立てた。

具体的には、以下のような手順に沿って作業を行なった。

1. WIDE ROOT CA の証明書 (自己署名) を作成する
2. ユーザが WIDE ROOT CA の証明書を入手するための WWW ページを用意する
3. ユーザが WIDE ROOT CA の証明書のフィンガープリント (証明書のハッシュ値) を確認するための WWW ページを用意する
4. moCA の管理者が、新しく上位 CA となる WIDE ROOT CA の管理者に対して、証明書への署名を依頼する
5. ユーザが新しい moCA 証明書を入手するための WWW ページを更新する
6. ユーザの行なうべき作業手順をまとめた WWW ページを用意する

さらに、ユーザへのプロモーション用の SSL 対応 WWW サーバの準備を行なうため、SSL 対応 WWW サーバの証明書について更新作業を行ない、WWW サーバ上で新しい

moCA 証明書の設定作業を行なった。

(2) アナウンス

CA 管理者からユーザへは、CA の系列を移動するため次の作業を依頼する旨のアナウンスを行なった。

- WIDE ROOT CA 証明書を登録すること
- 更新された moCA 証明書を登録しなおすこと

アナウンスの際は、ユーザを次のように分類した。

- 個人証明書を持つユーザ
- SSL のサーバ認証と暗号化の機能を設定している WWW サーバ管理者
- SSL のクライアント認証機能を設定している WWW サーバ管理者

個人証明書を持たないユーザへのアナウンスについては、SSL のサーバ認証と暗号化の機能を設定している WWW サーバ管理者から行なう、という段階的な方法をとった。

2.3.2　実験結果および考察

表 2.5に示すように、WIDE ROOT CA を 9 月 25 日に立ち上げた後、moCA の証明書を 9 月 25 日に発行し、ユーザへのアナウンスを開始したのは 9 月 29 日 (ICAT が運営を終了する日の前日、古い moCA 証明書の有効期限が切れる前日) となった。

(1) 実験結果

更新された moCA 証明書の登録を行なった人は、26 名であった。この中には個人証明書を既に持っていた人だけでなく、持っていない人も含まれている。

(2) 考察

- ユーザへの周知徹底のための手続きについて

  CA の管理者から電子メールなどで直接アナウンスできる範囲は、個人証明書や WWW サーバ証明書の発行を直接受けているユーザまでである。ユーザが S/MIME のやりとりをする相手や、SSL のクライアント認証の設定で moCA を信用している WWW

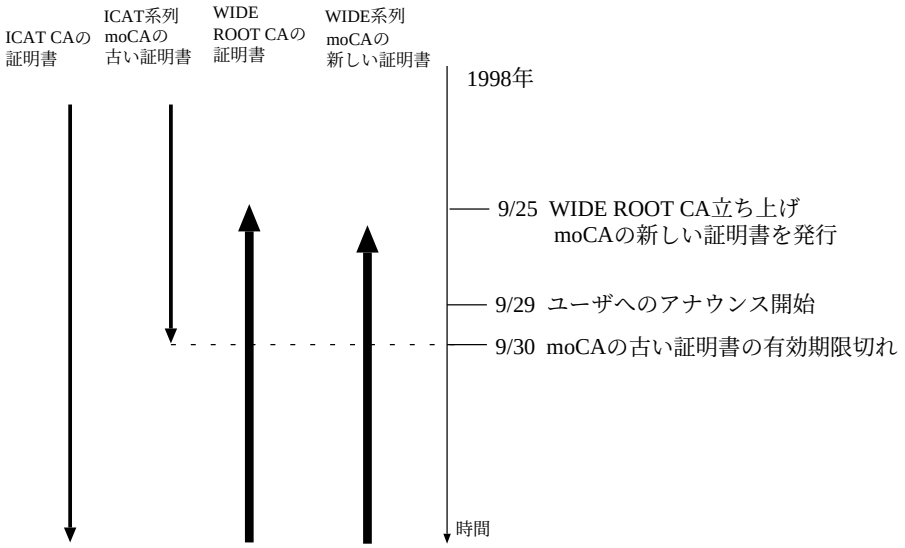


図 2.5: CA 系列移動実験の経過

サーバ管理者へは、個人証明書を持っているユーザから通知するか、CA からの通知用 WWW サーバ等を日頃からチェックしてもらうしかない。もしも組織の運営状況が証明書の有効期限と関係なく突然に変化した場合には、ユーザへの周知徹底がより困難になると予想される。

- 上位 CA が変わることによる影響について

  今回、moCA の鍵対を変えず、個人証明書を変更しない方針で実験を行なったが、個人証明書の中に含まれているポリシの記述 (Ceriticiate Policies という X.509 V3 拡張フィールド) に ICAT 依存の記述が残ったままとなってしまった (5.1.3参照)。証明書に含めるポリシの粒度にもよるが、上位 CA が変わればポリシの一部が変わる可能性があり、ポリシの変更を反映させるためには個人証明書の再発行が必要になることを考慮すべきだった。

- 系列移動の途中の問題点について

  moCA は 9 月 25 日から 9 月 30 日まで ICA 系列と WIDE 系列の両方に位置づく (マルチペアレント) 状態となった。

  今回の実験準備中、Netscape Communicator 4.05 では、ある CA 証明書を発行している上位 CA の証明書が複数存在する場合、信用するかどうかとは関係なく最新の上位 CA 証明書をカレントの上位 CA と判断する機能を持つことがわかった。しかし、最新の CA 証明書を信用する設定を行なわないと、Messenger で (moCA が証明した鍵を使った) 署名つきメールを出せなくなる、ということもわかった。

系列移動アナウンスの際は、WIDE 系列の署名つきメールを出したが、Messenger でアナウンスメールを読むユーザに対してはすぐに WIDE ROOT CA を信用する設定にするようアナウンス時に注意を呼びかけなければならなかった。

このように、CA の系列移動時には、アプリケーションツールの挙動の調査を含め、余裕を持って検討を進め、準備する必要がある。

この実験後、9 月 28 日には、SOI 分科会が実験運用する SOI CA が WIDE ROOT CA の下に入った。

2.4　SOI CA 鍵対変更実験

SOI CA の証明書の鍵が盗まれたという状況を想定し、WIDE ROOT CA から証明書の再発行を受けるとともに、ユーザへのアナウンスを行い、個人証明書の再取得を行ってもらうという実験を行った。この実験の目的は、以下の 2 点である。

- 上位 CA からの証明書再発行手続について考察すること
- CA の証明書の更新と個人証明書の再取得の手続きについて考察すること

2.4.1　実験概要

下位 CA の秘密鍵を紛失した場合に、上位 CA から証明書の再発行を受け、ユーザにアナウンスを行うための手続を実際に実験してみることによって、このような場合にいかなるトラブルが発生し、手続の完了までにどれほどの時間を要するかということを考察した。

(1) 上位 CA への証明書再発行申請
今回は突然のトラブルを想定していたため、本実験を行うことを WIDE ROOT CA の管理者に伝える以外、準備作業は一切行わなかった。よって今回行った作業が全て、実際の鍵紛失・盗難において必要な作業となる。作業の手順は、以下のとおりである。

1. WIDE ROOT CA 管理者に秘密鍵を紛失したことを伝える
2. 新しい CA の鍵を作成して、ICAP の「CA の初期設定」を行う（WIDE ROOT CA 管理者への公開鍵の送付、ポリシの送付も含む）
3. 新しい SOI CA の公開鍵の fingerprint を、電話で伝える
4. WIDE ROOT CA から新しく署名された SOICA の証明書を受け取る

5. 新しい SOI CA の証明書を組み込む

これらの作業は、殆どメールを用いて行ったため、各メールをもとに、作業を行った時間を図 2.6に示す。

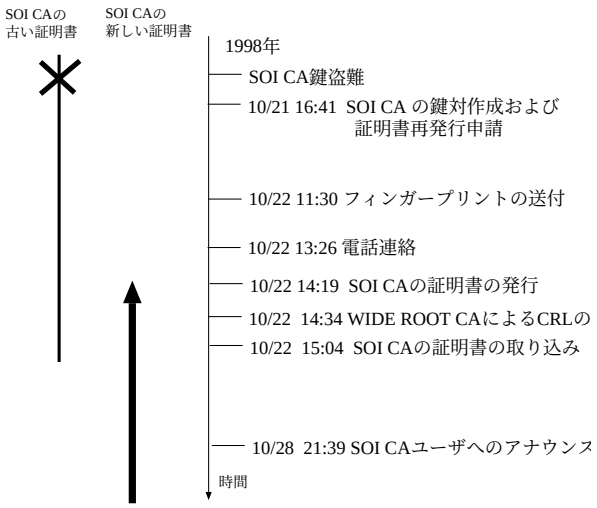


図 2.6: SOI CA 鍵対変更実験の経過

(2) ユーザへのアナウンス

ユーザには、以下の作業を依頼する旨のアナウンスを行なった。

- 更新された SOI CA 証明書を登録しなおすこと
- 証明書更新を SOI CA 管理者に依頼すること

証明書の更新作業は、SOI CA 管理者が手動で行ない、実際に本人と会って新しい証明書の入手先を本人に通知する方法をとった。作業を行った時間は図 2.6に示す通り 10/28 である。

2.4.2 実験結果および考察

(1) 実験結果
本実験までに実験に参加していた約 10 名のうち、本アナウンスを受けて更新作業を行なった人は 7 名、WWW サーバ 2 台であった。

(2) 考察

- CA の証明書の更新について

  上位 CA 管理者に鍵紛失・盗難を知らせ、証明書再発行の申請を行ってから、再発行された証明書を用いてユーザ証明書を発行できるようになるまで約 1 日の作業であった。実際の場面で要する時間は、上位 CA 管理者との連絡がどれほど迅速にできるかに大きく左右されるが、今回の手続で充分対応可能であると考えられる。

- 上位 CA へのフィンガープリントの連絡について

  上位 CA に鍵紛失・盗難を知らせ、証明書再発行の申請を行う際、真正な下位 CA からの申請であることを確認するために新しい証明書のフィンガープリントを上位 CA に伝える必要があった。しかし、新しい CA 証明書が設定途中のため、Netscape で取り込むことによってフィンガープリントを表示しようとしても、既に存在するとしてその表示が不可能であった。

  今回は SSLeay の x509 コマンドを用いて対処したが、古い CA 証明書を削除するという方法も考えられる。

  また、取出したフィンガープリントをメールで上位 CA 管理者に送ろうとした場合、既に本 CA の発行した個人証明書が使用できないため、別系列の個人証明書で署名することが必要であることが分かった。今回はこの取得が間に合わなかったため、上位 CA 管理者に電話でフィンガープリントを伝えるという方法をとった。

- 実験アナウンスについて

  今回は個人証明書を持っているユーザと、WWW サーバ証明書を持っているサーバ管理者に対して行った。内容としては、CA 証明書の更新と、各証明書の再取得を依頼するものであったが、大きなトラブルもなかったため、本内容で十分であると考えられる。

  ただ、証明書の更新が適切に行われていることを確認するために、暗号化メールを送ってみる或いは証明書を発行している WWW サーバにアクセスするよう依頼してみることも有用である。

2.5　ルート CA 鍵対変更実験 1

9 月末に WIDE ROOT CA が立ち上がり、moCA および SOI CA がその階層下に入った。ルート CA の運営も兼ねることになったため、CA を長期運用するときに本質的に最も難しいルート CA の鍵対変更実験を行なうことにした。特に、ルート CA の証明書の有効期限が切れるなど、ルート CA の古い鍵がまだ使える状況を想定し、古い鍵から新しい鍵への変更をユーザが確認できるように手順を検討し、試すこととした。

ルート CA の鍵対変更を行なう際、下位 CA が鍵対を変えるか変えないかの 2 通りが考えられることから、実験を 2 回に分けて行なうことにした。「実験 1」では、下位 CA が鍵対を変えない場合を試すことにした。

この実験の目的は、以下である。

- ルート CA の鍵対を変更したときの、ルート CA、下位 CA 各々の管理者の作業手順を検討し、よりよい手順について考察すること
- ルート CA の鍵対変更からユーザへのアナウンスまでにどの程度時間がかかるのか確認すること

2.5.1 実験概要

ルート CA をはじめて立ち上げる際には、ルート CA の証明書のフィンガープリント等を WWW サーバや印刷物など複数のメディアに記載して、ユーザへのアナウンスが正しいことを確認できる手段を提供するのが常である。ルート CA の鍵対を変更する際には、ルート CA の古い鍵が盗まれたり紛失したりしたのでない限り、古い鍵から新しい鍵に変わったということをユーザが確認できるような手段を提供すべきである。

そこで、この実験では、PKIX の CMP(Certificate Management Protocol)[13] で述べられている、NewwithOld, OldwithNew と呼ばれる特別な証明書を発行することによって、古い鍵を持っている者が確かに新しい鍵を作成したことを確認する手順を提供することにした。この方法では、古い鍵と新しい鍵を両方持っている場合にのみ、以下の証明書を作成できるという仮定から、これらの署名確認を行なうことによって、確かにルート CA の鍵が変わったことを確認しようとするものである。

OldwithOld	古い公開鍵を古い秘密鍵で署名した証明書。古いルート CA 証明書。
NewwithOld	新しい公開鍵を古い秘密鍵で署名した証明書。有効期間は、新しい鍵が作成された時点から OldwithOld の有効期限が切れるまでとする。
OldwithNew	古い公開鍵を新しい秘密鍵で署名した証明書。有効期間は、OldwithOld と同じとする。
NewwithNew	新しい公開鍵を新しい秘密鍵で署名した証明書。新しいルート CA の証明書。

CA 運用パッケージ ICAP では、NewwithOld や OldwithNew を簡単に発行する機能がなかったため、今回の実験用に開発して対応した。

(1) 準備作業

この実験では、WIDE ROOT CA の立ち上げ作業とともに moCA 証明書、SOI CA 証明書の更新作業が必要となる。具体的には、WIDE ROOT CA の管理者および下位 CA の管理者が連携しながら表 2.3に示す手順に従って作業を行なった。

さらに、ユーザへのプロモーション用の SSL 対応 WWW サーバの準備を行なうため、SSL 対応 WWW サーバの証明書について更新作業を行ない、WWW サーバ上で新しい moCA 証明書の設定作業を行なった。

(2) アナウンス

まず、WIDE ROOT CA 管理者からユーザ (WIDE メンバ全員) へ鍵対を変更した旨を通知した。NewwithOld, OldwithNew 証明書の署名確認については、WWW ページにガイドを載せて紹介した。
後日、下位 CA 管理者からユーザへは、次の作業を依頼する旨のアナウンスを行なった。

- 新しい WIDE ROOT CA 証明書を登録しなおすこと
- 更新された moCA 証明書を登録しなおすこと

アナウンスの際、ユーザを次のように分類してアナウンスを行なった。

- 個人証明書を持つユーザ
- SSL のサーバ認証と暗号化の機能を設定している WWW サーバ管理者
- SSL のクライアント認証機能を設定している WWW サーバ管理者

また、前回の実験と同様に、個人証明書を持たないユーザへは SSL のサーバ認証と暗号化の機能を設定している WWW サーバ管理者から通知する、という段階的な方法をとった。

2.5.2　実験結果

実験経過は表 2.7に示すとおりであった。
10 月 29 日に WIDE ROOT CA の鍵変更作業を開始してから、moCA のプロモーション用の SSL 対応 WWW サーバの設定が終了したのは 11 月 2 日であった。週末をはさんでいたことを考えると作業自体には約 2 日程度かかったことになる。
また、moCA のユーザの中で実際に moCA 証明書の登録をしなおした人が 26 名であった。これは、CA 系列移動実験参加者の 60%にあたる。

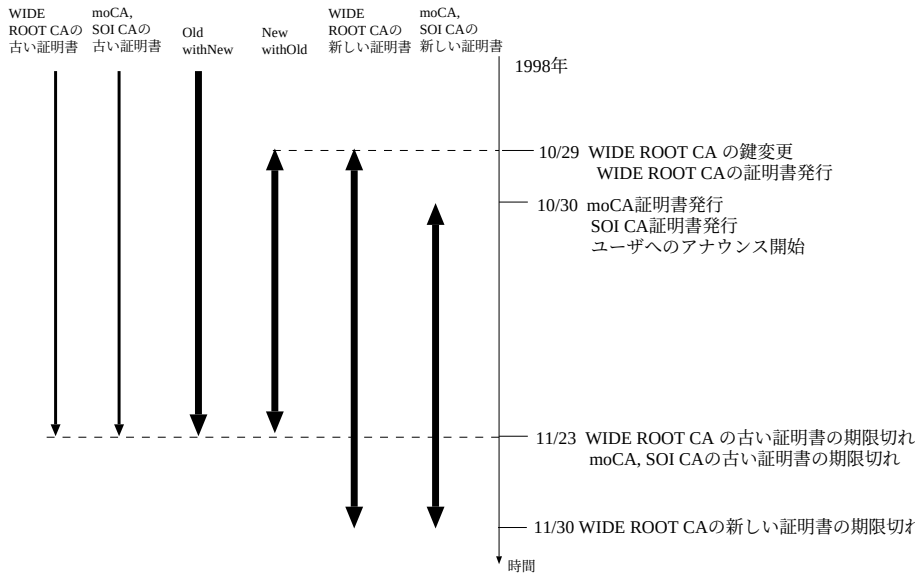


図 2.7: ルート CA 鍵対変更実験 1 の経過

2.5.3 ルート CA から見た考察

- NewwithOld, OldwithNew 証明書の確認について

  下位 CA 管理者に対しては CA パッケージ ICAP に含まれるコマンドを利用した NewwithOld, OldwithNew の証明書の署名確認を必須とした。CMP では、ユーザが通信相手の証明書を確認する際に CA 証明書の新旧を確認する方法について記述されているが、今回はユーザが確認できるような手段は提供できなかった。

- 鍵対変更からアナウンスまでにかかった時間について

  CA 管理者間で事前に連絡をとっていたことや、ルート CA 管理者は実験期間中この作業だけに時間を使えたことを考慮すると、実働 2 日間というのは短時間で鍵変更を実施できたケースと考えられる。

2.5.4 moCA から見た考察

NewwithOld, OldwithNew 証明書の検証をする立場から考えると、NewwithOld, OldwithNew の署名だけを確認しても、古い鍵を持っている者が何かに署名し、また新しい鍵を持っている者が何かに署名している、ということがわかるだけで、古い鍵から新しい鍵に移り変わったかどうかの確認にはならない。確認の過程では、NewwithOld に含まれる公開鍵が NewwithNew の公開鍵と同一かどうか、また、OldwithNew に含まれる公開鍵が OldwithOld の公開鍵と同一かどうかを確かめる必要がある。

厳密には以下の条件を全て満たすことを確認する必要があると考えられるが、管理者が本当に理解しながら確認できるのか疑問が残った。

- OldwithOld の公開鍵で NewwithOld の署名を確認し、正しいと判定されること
- NewwithOld の公開鍵で NewwithNew の署名を確認し、正しいと判定されること
- NewwithNew の公開鍵で OldwithNew の署名を確認し、正しいと判定されること
- OldwithNew の公開鍵で OldwithOld の署名を確認し、正しいと判定されること

2.5.5　SOI CA から見た考察

SOI CA の証明書更新作業は、WIDE ROOT CA 管理者への申請からユーザへのアナウンスまで約 1 日で終了した。これは、これまで使用していた鍵対を変更する必要がなく、フィンガープリントを署名付きメールで送るといった作業が比較的簡単にできたためである。

ユーザへのアナウンスについては、メールでこれを行ったが、正確にどれくらいのユーザが更新を行ったかを把握することができなかった。また、サイト証明書を発行している WWW サーバの管理者側において、具体的にどのような作業をすればよいか分からないといった質問があったため、簡単なマニュアルを用意することも必要であることが分かった。

2.6　ルート CA 鍵対変更実験 2

「実験 2」では、下位 CA(moCA のみ) が鍵対を変える場合を試すことにした。

この実験の目的は、節 2.5と同様である。

2.6.1　実験概要

節 2.5と同様に、今回もルート CA が鍵対を変更するため、NewwithOld, OldwithNew 証明書を発行し、古い鍵から新しい鍵へ変わることを確認できる方法を提供することにした。

(1) 準備作業

この実験では、WIDE ROOT CA の立ち上げ作業とともに moCA 証明書、SOI CA 証明書の更新作業が必要となる。今回は、moCA 証明書は鍵対を変更したため、moCA 発行の個人証明書は全て再発行となる。具体的には、WIDE ROOT CA の管理者および下位 CA の管理者が連携しながら表 2.4に示す手順に沿って作業を行なった。「実験 1」と比較すると、太字の部分が作業として増えている。

(2) アナウンス

節 2.5と同様のアナウンスを行なった。
ユーザへの依頼事項のうち節 2.5との違いは、下線の項目である。

- 新しい WIDE ROOT CA 証明書を登録しなおすこと
- 新しい moCA 証明書を登録しなおすこと
- 個人証明書の再発行を moCA 管理者に依頼すること

2.6.2 実験結果

実験の経過は表 2.8に示すとおりであった。

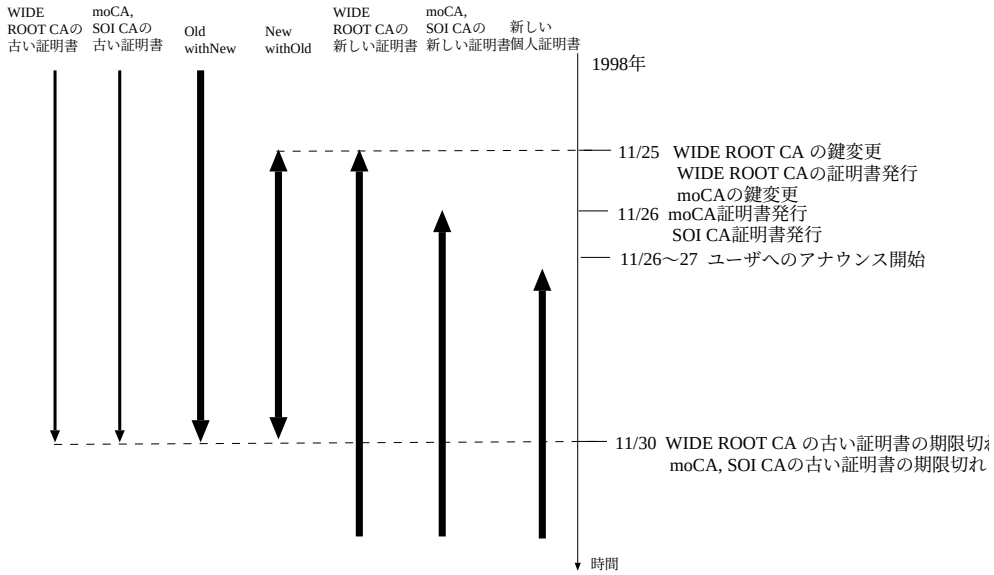


図 2.8: ルート CA 鍵対変更実験 2 の経過

11 月 25 日に WIDE ROOT CA の鍵変更作業を開始してから、moCA のプロモーション用の SSL 対応 WWW サーバの設定が終了したのは 11 月 27 日であった。「実験 1」と同様、実働 2 日程度かかったことになる。

moCA のユーザの中で実際に個人証明書を発行したのは、8 名であった。

2.6.3 ルート CA から見た考察

- Web キャッシュの問題について

新しい WIDE ROOT CA の証明書を Web ページで提供するために、古い証明書を提供していた時と同じファイル名で中身だけを更新していた。しかし、確認のためにブラウザに新しい WIDE ROOT CA の証明書を登録しようとしたところ古い証明書がロードされてしまうことがあった。

原因は HTTP proxy のキャッシュに古い証明書が残っていたためで、ブラウザの操作では解決できない問題である。現状では、古い証明書とは別のファイル名をつけて問題を回避する必要がある。WIDE ROOT CA の証明書入手先をリンクしている Web サイトが他にあれば、新しいファイル名を Web サイト管理者に伝える必要がある。

- CRL について

  古い SOI CA 証明書、および古い moCA 証明書は有効期限の途中で無効になったことから、新しい WIDE ROOT CA の鍵を使って CRL を発行した。しかし、古い SOI CA 証明書、および古い moCA 証明書は新しい WIDE ROOT CA の鍵で署名した証明書ではないため、CRL 発行対象として適していないかもしれない。

  逆に、古い SOI CA 証明書および古い moCA 証明書を持っているユーザは、古い WIDE ROOT CA の証明書を持ち続けて新しい WIDE ROOT CA の証明書に気づかない可能性があるため、古い WIDE ROOT CA の鍵が使える状況ならば、その鍵で CRL を発行すべきかもしれない。手順としては、ルート CA の鍵を変更する前に下位 CA の証明を取り消して、古いルート CA の鍵で CRL を発行することになる。

2.6.4　moCA から見た考察

「実験 1」と違い、今回は moCA の鍵対を変更し、moCA が発行する全ての証明書を再発行した。前回と今回の実験を通じ、証明書 (と秘密鍵) のバックアップと鍵変更との関係について気づいた点があった。

Netscape Communicator のエクスポート機能を利用して証明書のバックアップを取ると、個人証明書の他、CA の証明書を含んだファイル (PKCS#12 形式) が生成される。

ルート CA の鍵対が変わっても moCA の鍵対が変わらない場合は、個人証明書が変わらないため、バックアップを取ろうという意識は働きにくい。

しかし、バックアップを取っておかないと、証明書データベースが壊れてバックアップから鍵を戻したときに、ルート CA 証明書が古い状態に戻る。古いルート CA の証明書の有効期限がまだ十分に残っている状態では、なかなか気づかない。

ルート CA の鍵対が変わって moCA の鍵対も変わった場合は、個人証明書が変わるため、バックアップを取ろうとし、このような問題がいくらか防止されるはずである。

ルート CA の鍵対変更の通知徹底は、CA のシステムの中で本質的に難しい点ではあるが、バックアップから鍵を戻したときには、最新の CA 証明書を取りに行くよう促す手段

をアプリケーション側でも用意するなどの工夫が必要である。

2.6.5 SOI CA から見た考察

SOI CA においては、他 2 つの実験と同様、約 1 日で全ての作業を終了することができた。ただ、今回は SOI CA の鍵対を変更していないので、ユーザに個人証明書の再取得を要請する必要がなかったにも関わらず、誤っていくつかの個人証明書を無効にし、CRL の発行を行ってしまうというトラブルがあった。しかし、CRL を発行された個人に関して、証明書の使用に実質的な影響はなかったため、個人証明書の再発行は行わなかった。

また、ユーザへのアナウンスでは、新しい CA 証明書の取得場所を、誤って新しくサイト証明書を発行した https で指定したため、「サーバ証明書の署名が無効になっています。サイトとの接続は保護されません。」というエラーメッセージがでて、サーバにアクセスできないという苦情が送られた。これに対しては、ポート 80 番 (http) の方で取得するようアナウンスし直すことによって対処した。今回は前回の鍵の紛失・盗難を想定した実験と日程的に近かったこともあり、鍵対の変更の有無について SOI CA 管理者側に多少の混乱が生じた。CA 管理者は、各状況に応じて、具体的にどのような作業が要求されるかということを十分に把握しておくことが必要である。

表 2.1: 証明書の有効期限ぎれや鍵対変更のケース

	証明書の所有者	鍵対変更	証明書有効期限切れとの関係	古い鍵の有無	例
(1)	エンドエンティティ	する	あり	あり	鍵の寿命前に鍵を変える
(2)	エンドエンティティ	する	あり	なし	鍵の紛失と期限切れが重なる
(3)	エンドエンティティ	する	なし	あり	鍵の盗難
(4)	エンドエンティティ	する	なし	なし	鍵の紛失
(5)	エンドエンティティ◯	しない	あり	あり	証明書の期限延長
(6)	エンドエンティティ◯	しない	なし	あり	CA の鍵変更、証明書の内容変更
(7)	CA ◯	する	あり	あり	鍵の寿命前に鍵を変える
(8)	CA	する	あり	なし	鍵の紛失と期限切れが重なる
(9)	CA ◯	する	なし	あり	鍵の盗難
(10)	CA	する	なし	なし	鍵の紛失
(11)	CA ◯	しない	あり	あり	証明書の期限延長
(12)	CA ◯	しない	なし	あり	上位 CA の鍵変更、証明書の内容変更 (CA 名変更等)
(13)	ルート CA ◯	する	あり	あり	鍵の寿命前に鍵を変える
(14)	ルート CA	する	あり	なし	鍵の紛失と期限切れが重なる
(15)	ルート CA	する	なし	あり	鍵の盗難
(16)	ルート CA	する	なし	なし	鍵の紛失
(17)	ルート CA[1]	しない	あり	あり	証明書の期限延長
(18)	ルート CA	しない	なし	あり	証明書の内容変更 (CA 名変更等)

表 2.2: 今年度行なった実験

実験時期	実験名	表 2.1のうちカバーしたケース
1998 年 6 月	証明書更新	(5),(11)
1998 年 9 月	CA 系列移動	(12)
1998 年 9 月	SOI CA 鍵対変更	(9)
1998 年 10 月	ルート CA 鍵対変更 1	(12),(13)
1998 年 11 月	ルート CA 鍵対変更 2	(6),(7),(13)

表 2.3: ルート CA 鍵対変更実験 1 の手順

ルート CA		下位 CA
鍵変更の事前通知	$\Longrightarrow$	
鍵変更 (NewwithNew、 NewwithOld, OldwithNew を作成)		
証明書配布ページの準備		
下位 CA 管理者への鍵変更通知	$\Longrightarrow$	
ユーザ (WIDE メンバ) への通知		
	$\Longleftarrow$	ルート CA 管理者に電話等で確認
	$\Longleftarrow$	下位 CA 証明書の再発行を申請
	$\Longleftarrow$	証明書のフィンガープリントに 下位 CA 管理者の鍵で署名した メールをルート CA 管理者に送信
下位 CA 管理者からの申請内容の確認、		
申請された証明書のフィンガープリントを確認、 下位 CA 管理者の認証 (メールの署名確認)		
新しい下位 CA の証明書を発行		
古い下位 CA の証明書を廃棄		
新しい下位 CA の証明書を送付	$\Longrightarrow$	
		下位 CA の証明書の署名確認
		NewwithNew, NewwithOld, OldwithNew の署名確認
		新しい下位 CA 証明書の登録
		下位 CA 証明書配布ページの準備 ユーザへの通知文作成 ユーザガイド準備
		ユーザへのアナウンス

表 2.4: ルート CA 鍵対変更実験 2 の手順

ルート CA		下位 CA
鍵変更の事前通知	$\Longrightarrow$	
鍵変更 (NewwithNew、 NewwithOld, OldwithNew を作成)		
証明書配布ページの準備		
下位 CA 管理者への鍵変更通知	$\Longrightarrow$	
ユーザ (WIDE メンバ) への通知		
	$\Longleftarrow$	ルート CA 管理者に電話等で確認
		下位 CA の鍵対を変更 下位 CA の運用を停止
	$\Longleftarrow$	下位 CA 証明書の再発行を申請
	$\Longleftarrow$	証明書のフィンガープリントに 下位 CA 管理者の鍵で署名した メールをルート CA 管理者に送信
下位 CA 管理者からの申請内容の確認、		
申請された証明書のフィンガープリントを確認、 下位 CA 管理者の認証 (メールの署名確認)		
新しい下位 CA の証明書を発行		
古い下位 CA の証明書を廃棄		
新しい下位 CA の証明書を送付	$\Longrightarrow$	
		下位 CA の証明書の署名確認
		NewwithNew, NewwithOld, OldwithNew の署名確認
		新しい下位 CA 証明書の登録
		下位 CA 証明書配布ページの準備 ユーザへの通知文作成 ユーザガイド準備
		ユーザへのアナウンス
		個人証明書,WWW **サーバ証明書の廃棄**

第 3 章

アプリケーションツールの変遷と実験との関係

昨年度より 2 年間に渡る実験期間中、ユーザの利用するアプリケーションツールを Netscape Communicator に限定してきたが、それでも機能の追加や仕様の変更により、実験の進め方に少なからず影響があった。本章では、アプリケーションツールの動作の変遷と実験との関係についてまとめる。

また、Internet Explorer や Lynx など、公に実験対象とはしなかったツールに関しても、分科会内で実験対象とできるかどうかについて随時調査を進めてきた。これらの情報についても記録のためにまとめる。

3.1 Netscape Communicator

Netscape Communicator の Web ブラウザ Navigator とメールクライアントである Messenger での証明書の利用が試みられた。

以下に、実験中に発見された現象と時期、バージョン、実験への影響をまとめる。

1. 秘密鍵と証明書のエクスポート機能がなかったため、ユーザが鍵をバックアップするには Netscape Communicator の設定ファイルを直接コピーしておく他なかった。
   (1997 年 7 月 関連するバージョン：3.0)

   [実験への影響]··· バックアップの取り忘れが予想されたため、実験ではユーザ 1 名につき何通も証明書を発行できる方法をとった。

2. 利用者が複数の個人証明書を持っている場合に、クライアント認証付き Web サーバへアクセスしようとすると証明書の自動選択機能が効かない。
   (1997 年 7 月 関連するバージョン：3.0)

   [実験への影響]··· moCA 以外が発行する証明書を持っているユーザの利便性に問題が生じた。

3. 証明書が X.509 V3 でないと、S/MIME メールクライアントとして利用できない。
   (1997 年 11 月 関連するバージョン：3.0, 4.0)

[実験への影響] ··· S/MIME を利用できるようにするため、実験での個人証明書フォーマットを X.509 V1 から V3 に変更し、さらに netscape-cert-type という拡張フィールドのサポートを行なうことにした。

4. シリアル番号と有効期限だけを変えた、同じ CA 証明書を組み込むことができる。
(1998 年 2 月 関連するバージョン： 4.0)

   [実験への影響] ··· moCA 証明書の公開鍵を変更せずに有効期限を延長することができる。

5. 電子メールアドレスが証明書の Subject の PKCS9email 属性にないと Messenger での S/MIME 利用ができない。
(1998 年 3 月 関連するバージョン：4.0x)

   [実験への影響] ··· 証明書のフォーマットを変更して電子メールアドレスを Common Name から PKCS9email 属性に変更した。

6. 他人の証明書をインポート/エクスポートする機能がない。
(1998 年 3 月 関連するバージョン：3.0, 4.0)

   [実験への影響] ··· S/MIME を利用する時には、他人の証明書を組み込む為の Web ページを用意する必要があった。

7. 署名者として、前の CA 証明書を残したまま更新された証明書を組み込むことができる。
(1998 年 6 月 関連するバージョン：4.0)

   [実験への影響] ··· 利用者による、ブラウザから moCA 証明書を消す操作が必要なく、moCA 証明書の更新がスムーズにできる。

8. S/MIME の署名に更新された CA 証明書を含めると、そのメールを読んだ Communicator の証明書データベースの CA 証明書が更新される。(1998 年 6 月 関連するバージョン：4.05 以降)

   [実験への影響] ··· moCA 証明書を更新する際に、Messenger の利用者は明示的に WWW サーバから moCA 証明書を入手する必要がなかった。

9. 他の人の証明書を削除すると、データ自体が消えてもリストからは消えない場合があり、再度同じ人の証明書を組み込むことができなくなる。
(1998 年 11 月 関連するバージョン： 4.5)

   [実験への影響] ··· この状況を回避するには、key3.db および cert7.db を一度削除しなければならず、今まで登録されていたすべての証明書をバックアップなどから戻す必要がある。

10. S/MIME の署名に含まれる CA の証明書群に、ルート CA 証明書が含まれない。
(1998 年 11 月 関連するバージョン：4.5)

    [実験への影響] ··· S/MIME の署名を行ったメールを送信するだけでは、WIDE ROOT CA 証明書を配布することができず、利用者は必ず WWW サーバから WIDE ROOT CA 証明書を入手する必要があった。

11. WIDE ROOT CA および moCA 証明書をあらかじめ組み込んだ WIDE 版 Netscape が作成された。
(1999 年 1 月 関連するバージョン：4.5)

    [実験への影響] ··· WIDE 版 Netscape を利用する場合には、ユーザが CA 証明書の登録作業を省略できるようになった。

3.2 Internet Explorer

以下に、実験中に発見された現象と時期、バージョン、実験への影響をまとめる。

1. IE での秘密鍵の作成や個人証明書の登録がマニュアルに示されている通りに動かない。
(1997 年 7 月 関連するバージョン：3.0,4.0)

    [実験への影響] ··· Netscape に秘密鍵と証明書のバックアップ機能 (エクスポート機能) がなく、IE に秘密鍵と証明書のインポート機能もなかったため、IE を実験対象とすることができなかった。

2. IE では IPRA, ICAT で採用していた X.509 V1 証明書を組み込むことができず、SSL の暗号通信にも利用することはできない。
(1997 年 7 月 関連するバージョン：4)

    [実験への影響] ··· 実験に IE を使うことができなかった。

3. CA 証明書を登録していなくても、警告付きではあるが SSL 暗号機能を使った HTTP を利用できる
(1998 年 3 月 関連するバージョン：4.x)

    [実験への影響] ··· 個人証明書は使えないが、moCA が発行した証明書を持っている WWW サーバに対して SSL 暗号化通信が実現できるようになった。

4. IE に組み込んだ個人証明書の有効期限は CA 証明書の有効期限内でなければならない。(1998 年 11 月 関連するバージョン：4,5)

    [実験への影響] ··· moCA によって発行された個人証明書の有効期限 (1999 年 6 月) が moCA 証明書 (1998 年 11 月 30 日) の有効期限を越えていたため、CA 証明書の検

証に失敗する。1998 年 11 月実験以降は個人証明書の有効期限が moCA 証明書の有効期限内となり、問題は解消した。

5. Netscape を使って Export した個人証明書を IE で Import することによって、Outlook Express 4.72.3110.5 で個人証明書が正しく検証される。
(1999 年 2 月 関連するバージョン：4)

   [実験への影響] · · · Outlook Express で、moCA の発行した個人証明書を使って S/MIME を利用することができる。そして、moCA が発行した証明書を持っている WWW サーバに対して警告なしに SSL 暗号化通信が実現できるようになった。

3.3 Lynx

Lynx2.8.1 での SSL の利用について、1999 年 1 月下旬に尾上氏を中心として開発、実験及び検証が行なわれた。その結果、Lynx を使ってクライアント認証を含めた SSL 暗号通信が利用できるようになった。
同時期に OpenSSL パッケージで使われる pkcs12 プログラムの利用によって、Netscape と Lynx で用いる個人証明書の共有が可能であることが確認された。

第 4 章

おわりに

われわれは、証明書の有効期限ぎれや鍵変更といった CA の運用維持のための問題に着目し、実験を行なってきた。これらの実験を通じて、CA の運用を維持するためには、起こり得るケースごとの対応手順を事前に準備しておくことが重要であると認識することができた。実験時のユーザの作業手順から言うと、各実験内容が毎回似ているためか、実験参加者は少なく、ユーザをひきつける工夫がもう少し必要であったと反省している。しかし、このような実験では、実験に至る過程での検討が特に重要で、いくつかのケースについて対応手順が明確になってきたことが大きな成果であったといえる。

今年度も既存のアプリケーションを利用してきたが、アプリケーションの仕様変更により、ユーザの利用するアプリケーションのバージョンによってアナウンス方法を変えることが必要であった。これは CA 運用側にとっても、ユーザ側にとっても困難な状況である。しかし、CA 技術を導入し実用化するためには、アプリケーションベンダとタイアップしてユーザインターフェイス等の改善を求めていく努力を地道に続けるしかないであろう。

本分科会の立ち上げた moCA は、CA 技術をプロモートする観点が主であったが、今年度 SOI 分科会と連携して立ち上がった SOI CA は、アプリケーションの立場から CA 技術を利用しようとする観点が主であり、あるべき方向へ一歩前進した感がある。本分科会は今年度でいったん終了することになったが、今後も CA の運用は続け、他の分科会で CA が必要となったときに何らかの形でサポートできるようにしていきたい。

第 5 章

付録

5.1 moCA 仕様

平成 10 年度の WIDE members only CA の運用実験時の仕様についてまとめる。

5.1.1 CA サーバ

機種	Sun SPARCStation2
OS	SunOS 4.1.4
ディスク容量	約 800Mbyte
CA プログラム	ICAP2.42 + カスタマイズ
Web サーバプログラム	Apache1.2 +SSLpatch (SSL 対応のパッチ)
SSL プログラム	SSLeay-0.8.1b

SSL 化しているのは、SSL 暗号化機能を利用し個人情報入力時の盗聴を防止するためである。

5.1.2 CA の位置付け

図 5.1に示すように、9 月の CA 系列移動実験を経て、独自ルート CA の階層下に位置づけられている。

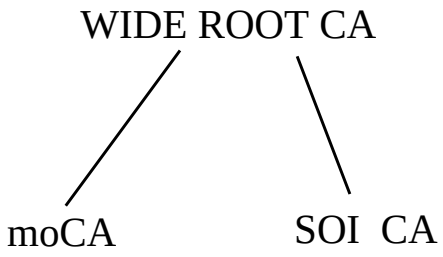


図 5.1: moCA の階層上の位置づけ

5.1.3 個人証明書フォーマット

	6 月〜 11 月まで	11 月以降
X.509 のバージョン	3	3
ユーザ識別子 (DN)		
Country	JP(固定値)	JP(固定値)
Organization	WIDE Project(デフォルト値)	WIDE Project(固定値)
Organizational Unit	(自由に記入)	(自由に記入)
Common Name	氏名	氏名
Email(PKCS#9 定義)	WIDE ML に登録されている電子メールアドレス	WIDE ML に登録されている電子メールアドレス
X.509 拡張フィールド		
basic Constraints	not CA (0)	なし
certificatePolicies	ICAT ポリシを示す番号	なし
subjectAltName		
(rfc822Name)	なし	WIDE ML に登録されている電子メールアドレス
authorityInfoAccess(*)	ICAP 固定値	ICAP 固定値
cRLDistributionPoints	ICAP 固定値	ICAP 固定値
netscape-cert-type(**)	SSLclient および S/MIME	SSLclient および S/MIME

(*) ICAT 独自

(**) Netscape 独自

11 月実験時、電子メールアドレスが 2 箇所に入っているのは、Email(PKCS#9 定義) しかサポートしていないツールと、今後標準としてサポートされるべき subjectAltName(rfc822Name) しかサポートしていないツールがあった場合でも、相互に電子メールのやりとりができるように配慮したためである。

11 月実験時の個人証明書内容例:

```
Version No = 2
Serial No = AA
Validity        from 981127032754Z
                to   000630235959Z
issuer:
        C=JP
        O=WIDE Project
        OU=members only CA
subject:
        C=JP
        O=WIDE Project
        OU=NEC Corporation
        OU=Internet Technology Labs
        CN=Mine Sakurai test(8)
        emailAddress=m-sakura@ccs.mt.nec.co.jp
signature:
        md5WithRSAEncryption
publickey:
        alg = rsaEncryption
subjectAltName:
```

```
        not critical
        rfc822Name:     m-sakura@ccs.mt.nec.co.jp
authorityInfoAccess:
        not critical
        authorityInfo:
                http://moca.wide.ad.jp/cgi-bin/calookupreq
        certStatus:
                http://moca.wide.ad.jp/cgi-bin/verifyreq
cRLDistributionPoints:
        not critical
        DistributionPointName:
                fullName:
                        http://moca.wide.ad.jp/cgi-bin/crlreq
netscape-cert-type:
        not critical
        Type: SSLclient S/MIME
```

5.1.4　1 メンバが発行できる証明書の個数

平成 10 年度は、100 個程度まで発行することが可能。ただし、2 個目から証明書発行理由の選択を必須とするようにした。

5.2　実験アナウンス・ガイド

5.2.1　WIDE ROOT CA 関連

http://www.wide.ad.jp/からたどれる WIDE ROOT CA のアナウンスページの一部を図 5.2,5.3 に示す。ちなみに、WIDE ROOT CA のフィンガープリントは、以下である。

```
55:94:21:73:e7:03:99:ab:f9:6a:eb:34:eb:54:92:a2
```

5.2.2　moCA 関連

http://moca.wide.ad.jp/にあるアナウンスページおよびガイドの一部を図 5.4,5.5 に示す。

1. はじめに
2. CA 運用実験

2.1 実験の体系づけ

2.2. 証明書更新実験

2.2.1 実験概要

2.2.2 実験結果および考察

2.3 CA 系列移動実験

2.3.1 実験概要

2.3.2 実験結果および考察

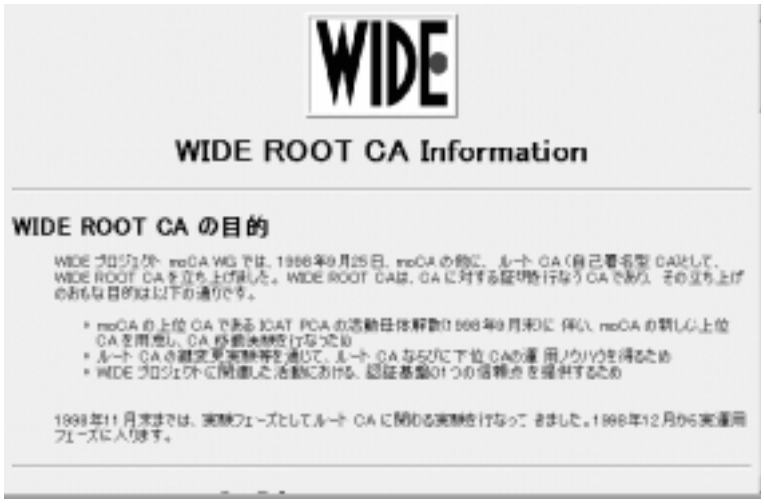


図 5.2: WIDE ROOT CA トップページ

2.4 SOI CA 鍵変更実験 → 村上さん
2.4.1 実験概要
2.4.2 実験結果および考察
2.5 ルート CA 鍵変更実験 1
2.5.1 実験概要
2.5.2 実験結果
2.5.3 ルート CA から見た考察
2.5.4 moCA から見た考察
2.5.5 SOI CA から見た考察 → 村上さん
2.6. ルート CA 鍵変更実験 2
2.6.1 実験概要
2.6.2 実験結果
2.6.3 ルート CA から見た考察
2.6.4 moCA から見た考察
2.6.5 SOI CA から見た考察 → 村上さん
2.7. 全体の考察
3. アプリケーションツールの変遷 →木村さん
3.1 Netscape Communicator
(3.x, 4.0x, 4.5 各々の機能により実験が影響したところなど ML のアーカイブから抜き出す。WIDE 版 Netscape については触れる?)
3.2 Internet Explorer

図 5.3: WIDE ROOT CA 証明書表示ページ

(3.x, 4.x, 5.0 各々の機能により実験が影響したところなど ML のアーカイブから抜き出す。)

3.3 lynx

(尾上さんによる lynx の SSL, S/MIME 対応について)

4. ?? → any other topic by anyone?

5. おわりに

6. 付録

6.1 moCA 仕様 → done (昨年度の報告書に沿って現在の仕様を書いた)

6.1.1 CA サーバ

6.1.2 CA の位置づけ

6.1.3 証明書フォーマット

6.1.4 1 メンバが発行できる証明書の個数

6.2 実験ガイド →ブラウザの画面イメージ切り貼り (ページ数との兼ね合いがあるので、最後に。)

6.1.1 WIDE ROOT CA (トップページと NewwithOld の説明ページを入れる。)

6.1.2 moCA (きれいに整理されたトップページとそれからクイックリファレンスの載っているページはどうしても入れたい)

6.1.3 SOI CA (カスタマイズされた ICAP のトップページは入れたい。SOI WG の報告書に同じものを載せるなら省略もあり得る!?)

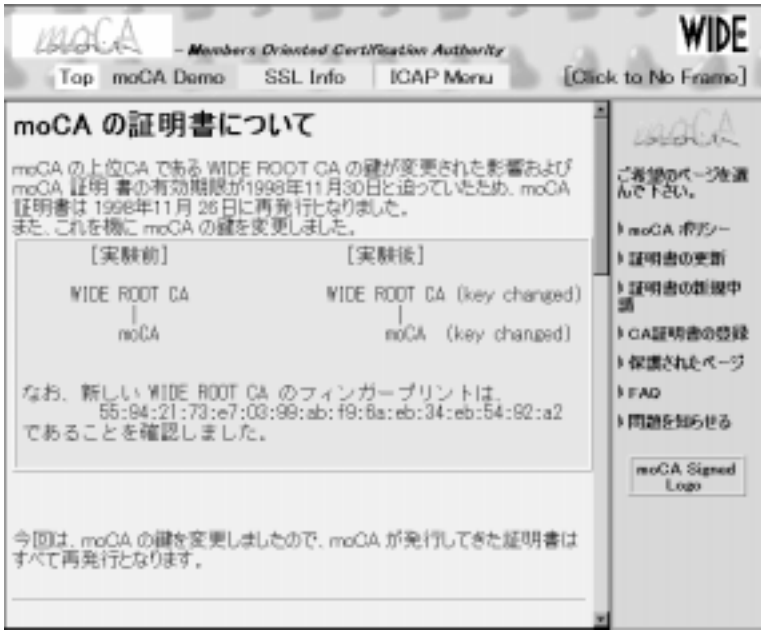


図 5.4: moCA アナウンスページ

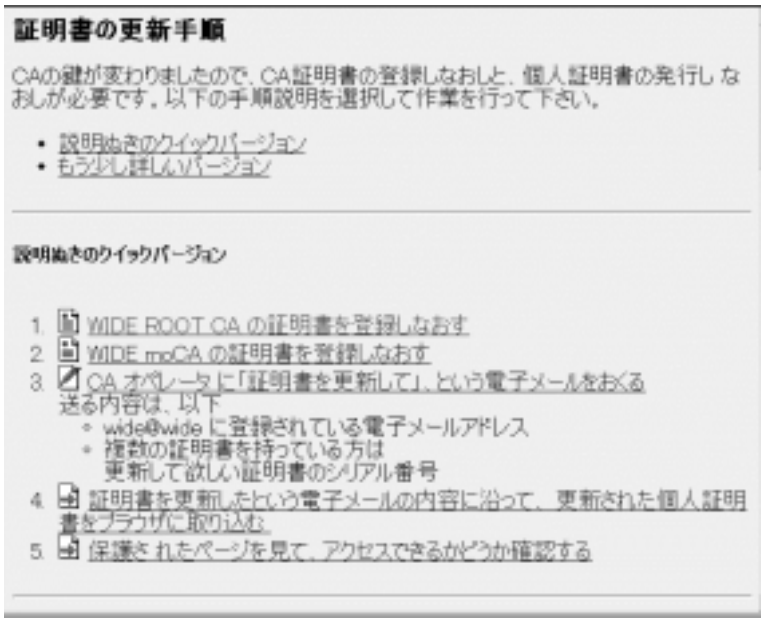


図 5.5: 証明書更新手順 (簡易版)