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 3 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。最近、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[104][105][106] などの電子メールのセキュリティ強化への応用、SSL[107](Secure Sockets Layer), TLS[108](Transport Layer Security) を用いた WWW のセキュリティ強化への応用が見られる。特に、Netscape 社製や Microsoft 社製など Web ブラウザのように一般的に広く受け入れられているツールが標準で SSL 対応になってきており、かなり多くのユーザが CA 技術を利用できる環境になりつつある。

しかし、CA 技術が既に一般に普及しているかというと、まだそのような段階にはない。現状はほとんどのシステムが実験段階である。しかも、CA やアプリケーションについての機能確認やパフォーマンス評価を目的とした実験が主体であり、CA の運用面に焦点をあてた実験はほとんど行なわれていない。CA 運用に関する各種ツールの開発が進んでも、CA が発行する証明書内のユーザ識別子を具体的に何にするか、ユーザ識別子と公開鍵の対応を証明するのにどのような情報を根拠とすべきか、といった点については CA を運用する各組織が証明対象に応じて検討しなければならないことであり、運用に関するケーススタディが強く求められている。members only CA (moCA) 分科会は、WIDE メンバを対象とした具体的なケーススタディを行なう目的で立ち上がった。

WIDE プロジェクトでは、メンバ間が情報共有するための手段の 1 つとして Web サーバを運用しているが、メンバ限定の情報に対する保護方法はパスワード認証のみであり、さらなるセキュリティ強化について検討すべき時期にきている。例えば、SSL 対応 Web ブラウザや Web サーバを利用し、SSL のクライアント認証機能を用いることはセキュリティ強化方法の 1 つとして提案できる。SSL 対応 Web ブラウザが多くのユーザに利用できることを考えると、WIDE プロジェクトに参加する全メンバを対象にすることが可能であり、WIDE プロジェクトという組織全体のスケールを考慮した実験が行なえる。また、既存のツールを利用することで新しいアプリケーションの開発を必要とすることなく、CA 運用面に重点を置いた実験が可能である。

そこで、WIDE プロジェクトの参加メンバ限定の Web 情報へのアクセスコントロールを SSL のクライアント認証を用いて行なうために、セキュリティレベルと組織構造を考慮した CA の運用実験を行なった。

本報告では、まず証明書発行手続きを決定する上で考慮しなければならない WIDE プロジェクトの組織としての特徴についてまとめ、実験の前提となる Web のアクセスコントロールについて簡単に述べる。続いて、2 回行なった各実験の概要、結果および考察について述べる。

第 2 章

CA 運用実験

2.1 WIDE プロジェクトの組織的特徴

インタネットの研究プロジェクトである WIDE プロジェクトでは実験基盤となる WIDE インタネットの実運用、および、さまざまな分野と期間にわたる運用実験を行なっている。

2.1.1 構成員

WIDE プロジェクトは、おもに WIDE インタネットに接続している様々な大学や企業からの、学生や技術者から構成され、全体では 400 名を超える。WIDE プロジェクトのメンバとしての ID は、WIDE プロジェクト内のメーリングリストに登録される際の電子メールアドレスである。メンバには体制的に以下のような分類がある。

ボード 研究活動の他、WIDE プロジェクト全体の運営を行なう役割を持つメンバ。約 20 名。

一般メンバ ボードの会議で WIDE プロジェクトへの参加が了承されたメンバ。一般メンバには、入会、所属変更や脱会などの際に連絡役となるボード (担当ボード) が必ず一人つく。

2.1.2 研究活動の形態

WIDE プロジェクトメンバは、おもに技術分野ごとのグループ (ワーキンググループ、WG) に分かれて活動している。ワーキンググループによっては時々オフラインミーティングを開いているが、WIDE プロジェクト全体で顔を合わせる機会はおおよそ以下の場合である。

- 1 日構成の定例研究会 (5,7,11,12 月)
- 合宿形式の研究会 (3,9 月)

ミーティングや研究会以外では、ワーキンググループ単位や、全 WIDE メンバのメーリングリストを運用して情報交換や議論を行なっている。また、Web サーバを運用して、研究会の詳細プログラム等の情報をメンバ間で共有している。

2.2 Web のアクセスコントロール

2.2.1 アクセスコントロールのレベル

Web サーバへのアクセスコントロールを実現するには、以下の 2 つのレベルが考えられる。

- クライアントコンピュータ単位でのアクセスコントロール
- ユーザ単位でのアクセスコントロール

ユーザ単位でのアクセスコントロールにおいて、ユーザが本物かどうかを識別する情報として、以下の 2 つがある。

- パスワード
- 証明書

証明書を利用したアクセスコントロールは、たとえば Web サーバおよび Web ブラウザ間の通信を SSL を用いることで可能となる。

2.2.2 SSL のクライアント認証機能を用いたアクセスコントロール

SSL は、Socket レベルでのデータの暗号化機能や、サーバ認証機能およびクライアント認証機能を提供するプロトコルである。

SSL は、実際のデータの暗号化に先立って、公開鍵暗号を用いてセッション鍵をサーバ・クライアント間で共有する。サーバおよびクライアントそれぞれの公開鍵情報は、X.509 証明書フォーマット [109][110] で交換する。X.509 証明書は、図 2.1 に示すように、単に公開鍵ばかりでなく、その所有者、発行者、有効期限などの情報も含む。また、発行者 (CA) によって内容が保証され、CA の秘密鍵による署名が付いている。クライアントとサーバはあらかじめ信頼する CA を決めておくという前提の元で、受け取った証明書がその CA によって署名されているかによって検証を行なう。

SSL プロトコル上で Web のデータをやりとりする際には、Web サーバとブラウザに対して CA が証明書を発行し、Web サーバ側でアクセスを許可する証明書の情報をアクセスコントロールリストに登録することで、証明書を用いたアクセスコントロールが実現できる。例えば、SSLeay をベースとした SSL 対応 Web サーバでは、アクセスコントロールリストに図 2.2のように記述する。

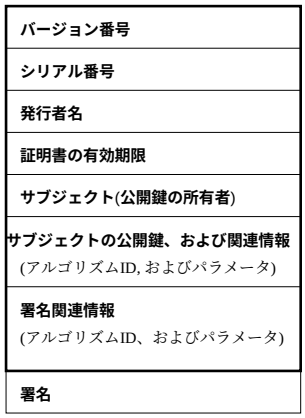


図 2.1: 証明書に含まれるデータ

（例）/usr/local/etc/httpd/conf/.htpasswd ファイル

```
/C=JP/O=WIDE Project/CN=abc@def.meiji.ac.jp:xxj31ZMTZzkVA
```

これは、クライアントから提示された証明書の所有者名 (サブジェクト) が、
/C=JP/O=WIDE Project/CN=abc@def.meiji.ac.jp
であるならばアクセスを許可するという意味である。

図 2.2: アクセスコントロールリストの例

2.3　実験目的

WIDE プロジェクト内の CA を対象にした運用実験では、あらかじめ各メンバに証明書を発行し、メンバ限定の Web 情報提供サービスの中で証明書を利用するという前提をおく。この実験の目的は以下の 2 点であるが、特に前者に重点を置いた。

- 証明書発行において、メンバ限定の Web 情報提供サービスに求められるセキュリティレベルと、WIDE プロジェクトという組織の構造とを考慮した最適な手続きを提案し、実証すること
- 既存のツール (ブラウザ、Web サーバ、CA 運用パッケージ) を組み合わせて実験し、その機能限界と問題点を明確化すること

前者で、最適な手続きとはアプリケーションに必要なレベルの本人確認が達成でき、かつ、一連の作業がユーザに受け入れられる手続きである。前者に重点をおくことによって、CA 運用ツールだけでは実現困難なノウハウを収集し、他の組織で同様のサービスを行なう場合にも参考になるような一般的なノウハウを得ることを目標とする。

2.4 実験概要

メンバ限定の Web 情報として 9 月の合宿研究会関連の情報をターゲットにした関係で、実験期間は 7 月の研究会直後から 10 月末までの 3ヵ月間とした。そして、実験期間の証明書発行ユーザ数の目標を合宿参加者数 (= 約 230 名) とした。

本節では、実験に対する考え方、実験システムの具体的な構成、証明書発行手続きの詳細について述べる。

2.4.1 考え方

証明書を用いた合宿情報へのアクセスコントロールを実現するためには、アクセスを許可する者 (= WIDE プロジェクトメンバ) に対し、事前に証明書を発行する必要がある。証明書発行時に最も重要となるのは、証明書を発行してよいユーザの本人確認手続きである。

今回のシステムではメンバ共通の情報を扱うため、メンバであることを確認できる程度の本人確認を行えばよい。その本人確認レベルは、組織でのメンバ登録時の本人確認レベルと同一となるべきである。メンバ登録時にメンバ証が発行されている場合はそれを確認するのがもっとも容易である。ない場合はメンバ登録手続きの再現を必要とする。再現の容易さは、本人確認に必要な情報の準備の容易さとその情報を確認する関係者の揃いやすさに影響を受けると予想される。WIDE プロジェクトでは、1) メンバ証はない、2) メンバ登録時の本人確認では特定の書類を必要としないが、メンバと担当ボードが顔を合わせるのが原則である。したがって、本人確認レベルは厳密ではないが、メンバの分散性の高さから、確認作業は困難なレベルにあると予想され、いくつかの手続きを提案し試すことにする。

2.4.2 システム構成

実験で利用した既存のツールは以下の通りで、これらを組み合わせて図 2.3に示す構成をとった。

- CA 運用サーバ

  ICAP1.0 [111]+ Web サーバ

- SSL 対応 Web ブラウザ

  Netscape Navigator 3.0 以上または Netscape Communicator 4.0 以上

- SSL 対応 Web サーバ

  Apache 1.2.+ SSLeay 0.6.6 ベースの SSL 対応パッチ

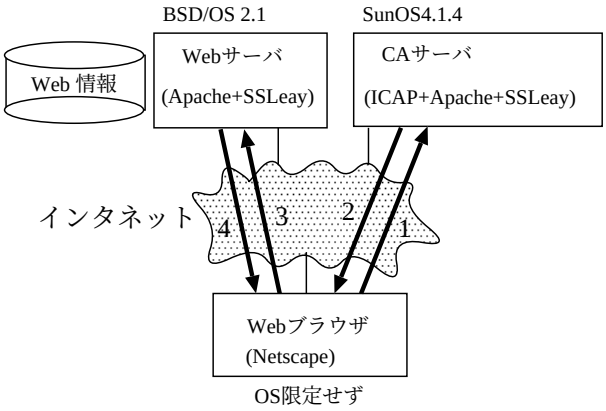


図 2.3: 実験のシステム構成

2.4.3　実験のステップ

一連の実験は 2 つのステップからなる。

ステップ 1 ICAP を利用して CA サーバを運用し、X.509 形式の証明書を発行する (図 2.3 の 1,2)

ステップ 2 Web サーバの SSL クライアント認証機能を用いて、証明書つきの SSL 対応 Web ブラウザから、アクセスコントロールをかけた Web ページにアクセスする (図 2.3 の 3,4)

各ステップにおいてメンバが行なった手続きを図 2.4、2.5 にまとめる。

ステップ 1 において、一般の発行手続きでは、本人確認後にその場で証明書を発行し配布する場合が多い。しかし、以下のようなツールの制約や特徴を考慮して、本人確認と同時に証明書発行用アカウントとパスワードを配布し、後日そのアカウントを使って各自が証明書を発行する手続きにした。

- Web ブラウザ用の証明書を発行するためにはブラウザ側で鍵を作成する必要がある[1]が、本人確認するその場でブラウザを持っているとは限らない

[1] 本人確認後に秘密鍵と証明書の対を渡したとしても、Netscape ブラウザにはそれを取り込む機能がまだなかった。

- CA サーバ (ICAP) にはアカウント登録機能があり、登録されたユーザがパスワードを使って各自で証明書を発行することが可能である

したがって、メンバの本人確認を行なった後に、安全な方法で 担当ボードから CA の運用者 (CA オペレータ) へ証明書発行用アカウントとパスワードを伝えて登録することにした (図 2.6)。

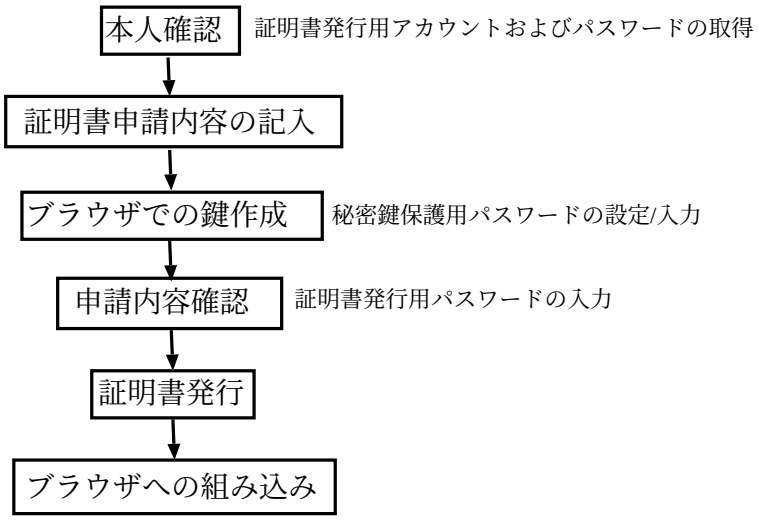


図 2.4: ステップ 1 の流れ

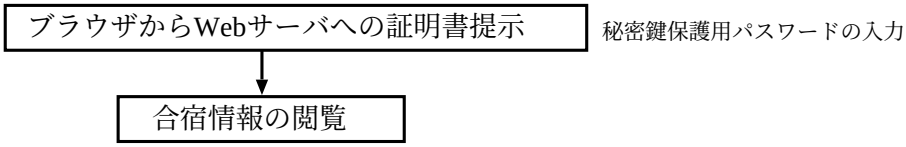


図 2.5: ステップ 2 の流れ

2.4.4 本人確認手続き

実験にあたって、1) 誰が、2) いつ、3) どのような方法で「WIDE メンバである」ことを確認するかを決める必要があった。

1) については、節 2.1.1 を考慮し、各メンバの担当ボードが最適であると判断した。2) については、節 2.1.2で述べた定例研究会を利用して本人確認を行なうのが最も効率のよい機会である。しかし、全員が参加するわけではないため、メンバと担当ボードが随時本人

確認を行なえるように決めた。3) については、WIDE プロジェクトのメンバ登録時に確認すべき書類が特に決められていないため、原則写真つき身分証明書類で確認することにしたが、詳細は各担当ボードに任せた。

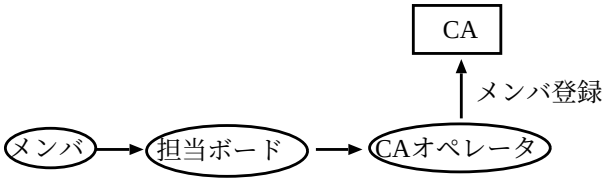


図 2.6: 本人確認の流れ

実際に行なった本人確認手続きは以下の 3 通りである。

(1) 7 月の定例研究会にて

パスワードの書かれた用紙 (節 6.2参照) に氏名、電子メールアドレスなどを記入し、担当ボードがメンバの身元と用紙の記入事項を確認した後、CA オペレータの目の前で用紙に署名してから CA オペレータに手渡す。

(2) 9 月合宿前日までの随時

メンバが担当ボードに電子メールアドレスとパスワードを何らかの方法で渡し、担当ボードがメンバを何らかの方法で確認し、担当ボードから CA オペレータには PGP を利用してデータを暗号化し、オンラインで情報を渡す[2]。

(3) 9 月合宿研究会にて

合宿会場受け付けや会議室にいるメンバがパスワードの書かれた用紙に氏名、電子メールアドレスを記入し、CA オペレータが身元と用紙の記入事項を確認する。

(1) はメンバ登録時の手続きに最も近い手続き、(2) は CA オペレータを含むメンバの分散性を考慮した手続き、(3) は手続きの行なわれる場所の物理的なセキュリティレベルを加味して本人確認を簡単にした手続きである。

2.4.5　証明書発行手続き

本人確認手続きの違いによらず、以下に示す手順で証明書を発行した。

1. 証明書発行アカウント登録通知を受けたメンバが、後日 SSL 対応 Web ブラウザを利用して CA サーバにアクセスする

[2](2) を実施する前には、あらかじめ CA オペレータがボードの身元確認を行ない、各ボードの PGP の公開鍵が正しいことを確認した。

2. ガイドにしたがって CA サーバの証明書発行メニューを実行し、必要事項の記入や鍵作成を行なう

3. 証明書発行アカウントとパスワードを用いて各自で証明書を発行し、ブラウザに証明書を格納する

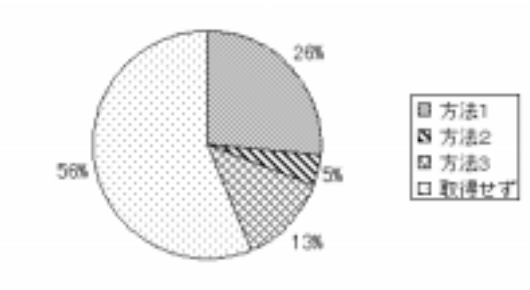
2.5 実験結果

2.5.1 本人確認手続き

方法別の集計を表 2.1に示す。合計数は 180 であり、目標数の 230 には満たなかった。

節 2.4.4 の方法 (1) は多かったが、担当ボード本人が立ち会うことについてはうまくいかず、担当ボード以外のボードが代理で署名することが多かった。(2) はメンバからボードにパスワード情報を渡す方法を決めなかったことから、逆にメンバを戸惑わせることになり、予想以上に数が少ない結果となった。(3) は担当ボードを通さないため最も簡単な手続きであり、(2) よりも多かった。

表 2.1: WIDE メンバ数 (411) に占める証明書発行用アカウント取得の割合

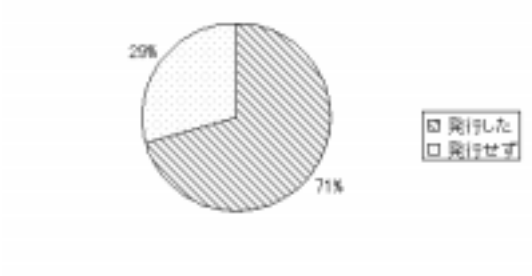


2.5.2 証明書発行手続き

実験の結果、実際に証明書を発行したメンバ数を表 2.2 に示す。本人確認手続きを済ませたメンバのうち、証明書発行手続きまで至ったメンバは 約 7 割 の 127 名となった。

証明書発行が最も集中した際には 1 日で 56 件の証明書発行処理が行なわれたが CA サーバでの性能的な支障は特になかった。

表 2.2: 証明書発行用アカウント総数 (180) に占める証明書発行者数の割合



2.6　考察

2.6.1　手続きに関する考察

A。本人確認手続き

節 2.5.1 で述べたように、「WIDE メンバであること」を確認するだけの比較的簡単な確認であったにもかかわらず、当初に決めた通りには実行できなかった。特に、担当ボードでない他のボードに本人確認の代行をお願いするケースが多く見られたが、ボードがメンバのことを知らない場合には身元確認ができても WIDE メンバかどうかを判定することは難しい。

WIDE のメンバ証があれば、メンバと CA オペレータの間のみでも確認手続きが行なえたはずである。しかし、メンバ証があったとしても、担当ボードか CA オペレータのどちらかがメンバと顔を合わせていないと確認は行なえない。メンバが物理的にも、所属組織的にも分散している環境では、本人確認手続きを短期間で行なうのは予想通り難しかった。

結果的には、組織構造が柔軟である場合、本人確認手続きの実行が困難であることが示唆された。しかし、今後新規に参加するメンバの本人確認手続きについては、メンバ登録手続きと連動させれば比較的容易になると考えられる。

B。証明書発行手続き

節 2.5.2 より、本人確認手続きを行なったメンバ数とその後証明書発行手続きまで行なったメンバ数にはかなりの差が見られる。実験期間中のメンバからの問い合わせ内容や、メンバに対して行なったアンケート調査から、証明書を発行する手順を示すガイドがわかりにくいという指摘が目立った。

特に、ブラウザの秘密鍵を保護するためのパスワードと ICAP に登録した証明書発行用アカウントのパスワードの使い分けが最も混乱を招いた点である。ガイドや ICAP イン

ターフェイスを改善すれば使い分けることが可能なのか、証明書発行用アカウントを使って各自で証明書を発行するという方法自体を見直した方がよいのか、切り分けて検討する必要がある。

2.6.2 既存のツールの機能に関する考察

A。ICAP

ICAP は全てのソースを公開しているわけではないが、ユーザインターフェイスなどを変更することは容易である。[3]

結果として、100 人規模の CA を運用することができ、一日あたり最大 56 件の証明書発行処理が行なえることが確認できた。定常運用を考えると以下のような改善が必要である。

- CA の秘密鍵のセキュリティ管理
- 設定機能の充実化

  証明書の有効期限など組織の事情に合わせて変更すべきパラメータを簡単に変えられるようにするなど

B。SSL 対応 Web ブラウザ

Windows、UNIX を問わず様々な OS での動作が確認されたが、以下のような不安定な面がみられた。

- OS とブラウザの日本語モードの組み合わせによっては鍵の作成時にブラウザが無反応になる
- 複数の証明書を持っている場合に、複数の証明書から自動的に提示する証明書を選択する機能があったが、正しく動作せず強制終了する

C。SSL 対応 Web サーバ

機能的に不十分な点として、証明書の廃棄リスト (CRL) を処理する機能がない点が挙げられる。証明書は有効期間内に廃棄されることがあり、CA が定期的に CRL として発行することになっているが、Web サーバ側に CRL を処理する機能がないと、無効となるはずのクライアント証明書を有効と判定することになる。この問題は、CRL のチェックを行な

[3] われわれは ICAP の開発にも携わっており、ユーザインターフェイス等の問題のいくつかは ICAP 次版で改良した。

うように、プログラムの改良を行なうことによって解決した。ただし、今回は試作のみで運用までには至らなかった。

また、今回 Web サーバの一部の情報に「WIDE の CA が発行した証明書を持つユーザは全てアクセス可能」というルールでアクセスコントロールをしようとした際に、以下のような不十分な点があった。

1. アクセスコントロールのポリシーが、Web ページ毎に設定できない
2. アクセスコントロールリストに証明書の所有者名 (サブジェクト) しか記述できず、複数の CA を信用している場合にどの CA が発行した証明書なのかを厳密に指定できない
3. アクセスコントロールリストに正規表現が使えず、アクセスを許可する人数分の DN 情報を記載しなければならない

i) については、アクセスコントロールをかけたい情報のために専用の Web サーバを運用して回避した。ii) については、Web サーバが信用する CA を 1 つに限定して問題を回避した。iii) については、Web サーバ管理者にとっては負担となることが予想されたため、図 2.2の中で正規表現を利用できるよう、プログラムの改良を行なって解決した。

第 3 章

追加実験

われわれは、7 月から 3ヵ月間行なった CA 運用実験の反省点を元にして、ターゲット情報を 1998 年 3 月の合宿に関するメンバ限定情報に変更し再度実験を行なうことにした。追加実験の期間は 1998 年 3 月から 6 月にかけてであり、1998 年 4 月現在実験途中であるが、7 月の実験内容との違いを中心に現時点での考察を述べる。

3.1 実験目的

基本的には、7 月実験と同一の目的 (節 2.3) のもとで、具体的な手順を変更し、比較を行なうことにした。

まず、証明書発行手続きについては、一連の証明書発行手続きに 2 つ以上のパスワードを使用するとユーザが混乱する点が明らかとなった。そのため、証明書発行手続きを改善するとともに実験ガイドを強化して 7 月実験の手続きと比較することを実験の第一の目的とした。

また、7 月の実験では、証明書を 2 個以上発行したケースが予想以上に多かったが、証明書を 5 個まで自由に発行できたため、証明書を 2 個以上発行した理由をアンケート以外の方法で調査するのは不可能であった。そのため、追加実験では理由を調査できるような仕組みを用意して再実験し、ユーザの実際の鍵管理上の特性や CRL の発行タイミングに関するヒントを得ることにした。

さらに、既存のツールについては、Web ブラウザのバージョンアップが進んだことによって、同一の証明書を利用して SSL クライアント機能と S/MIME 機能の両方に利用可能となった。そこで、SSL クライアント機能の利用を軸としつつ、S/MIME にも使えるような証明書を発行し、実験することとした。(付録 6.1.3に 7 月実験時の証明書フォーマットとの違いを示す。)

追加実験の目的についてまとめると、以下のようになる。

- 証明書発行手続きを比較し、最適な手続きについて模索すること
- 2 個以上証明書を発行する理由について調査し、ユーザの鍵管理の特性や、CRL の発行タイミングに関するヒントを得ること

- SSL クライアント機能 に加え S/MIME 機能について試行し、新しい機能を評価すること

3.2　実験のステップ

節 2.4.3と同様に、ステップ 1 とステップ 2 に分かれるが、ステップ 1 の具体的な手続きは、以下のように 2 通りを試すことにした。

- 図 3.1 に示すように、本人確認直後に証明書発行用アカウントを配布し、後日そのアカウントを使って証明書を発行する手続き
- 図 3.2に示すように、本人確認直後には証明書発行用アカウントを配布しない代わりに、ブラウザに証明書を組み込むためのパスワード (アクセスキー) を証明書発行後に電子メールで配布する手続き

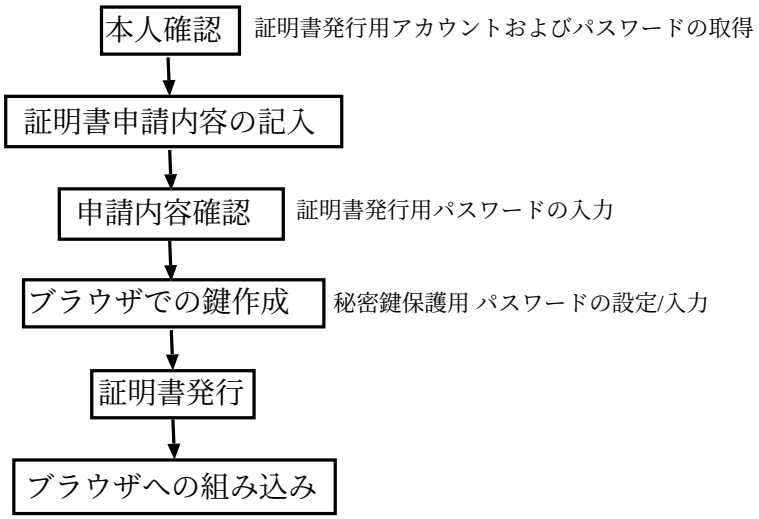


図 3.1: 追加実験におけるステップ 1 の流れ (その 1)

図 3.1 は、図 2.4と似ているが、パスワードを利用する順番が異なる。図 3.1では、あるパスワードが配布されたらそれを利用するまでは別のパスワードに関連した手続きが出てこないようにしている。これによって、複数のパスワードがあっても混乱なく使い分けられるのではないかと予想した。

図 3.2 では、本人確認手続きの際に申請された電子メールアドレスが正しいこと、およびその電子メールアドレス宛ての電子メールが本人のみに到達することを仮定している。証明書発行用パスワードをオフラインで配布するのに比較するとセキュリティレベルが落

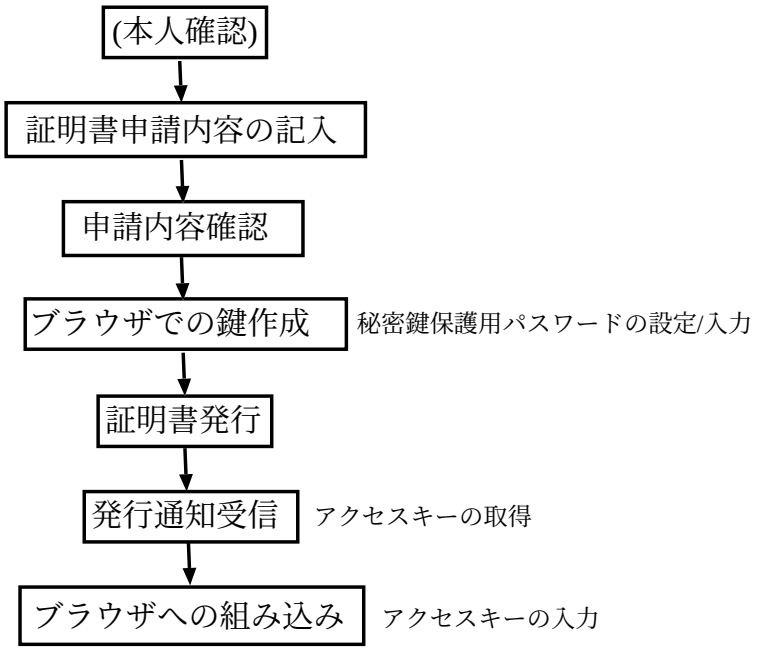


図 3.2: 追加実験におけるステップ 1 の流れ (その 2)

ちるが、証明書発行までの手続きをできるだけ簡単にし、証明書をブラウザに組み込む段階で、鍵の作成者と電子メールの受信者が一致しなければ手続きが完了しないようにした。もしも電子メールがエラーで返ってくるなど問題が発生した場合には CA オペレータが証明書を廃棄する運用を行なう。この手続きでは、電子メールで送られてくるパスワードの配布と利用のサイクルが短時間であることから、図 3.1で配布する証明書発行用パスワードよりも管理が楽になり、ユーザが受け入れやすいのではないかと予想した。

これら 2 通りの手続きを実験するために、被験者であるメンバを 2 つに分類し、1 メンバあたり 1 つの手続きを試してもらうことにした。

1. はじめて実験に参加するメンバ

   図 3.1 に沿って証明書を発行する。

2. 7 月実験に参加したメンバ

   図 3.2 に沿って証明書を発行する。

3.3 本人確認手続き

時期的に本人確認手続きをオフラインで行なう機会がなかったことから、メンバの分類ごとに以下の方法で確認を行なった。

1. はじめて実験に参加するメンバ

   節 2.4.4の (2) とした。

2. 7 月実験に参加したメンバ

   7 月実験時に行なった本人確認手続きをもって本人確認済みとみなし、省略した。

3.4　証明書発行手続き

はじめて実験に参加するメンバについては、節 2.4.5に示す手順で各自が証明書を発行した。

7 月実験に参加したメンバについては、以下に示す手順で各自が証明書を発行した。

1. SSL 対応 Web ブラウザを利用して CA サーバにアクセスする
2. ガイドにしたがって CA サーバの証明書発行メニューを実行し、必要事項の記入や鍵作成を行なう
3. 証明書を発行し、発行通知を電子メールで受け取る
4. 再度 CA サーバにアクセスし、発行通知に書かれたアクセスキーを用いてブラウザに証明書を格納する

一人で証明書を 2 個以上発行する際には、必要事項として証明書の発行理由を選択式で入力するようにした。

3.5　実験結果

1998 年 4 月現在、証明書の発行総数は約 60 であり、ほとんどが 7 月実験に参加したメンバによって発行されている。7 月実験時と比較すると証明書発行数は 1/3 である。

証明書を 2 個以上発行したケースは 14 件あった。その発行理由の内訳を、表 3.3 に示す。

図 3.3: 証明書を 2 個以上発行した理由

複数のプラットフォームで使いたいから	5
異なるバージョンの Netscape で使いたいから	3
環境をなくしたから	2
その他	4

3.6 考察

3.6.1 手続きに関する考察

はじめて実験に参加するメンバと、7 月実験に参加したメンバによって、ステップ 1 の手続きを変えて比較しようとしたが、はじめて実験に参加するメンバがほとんどなく、比較できる段階ではない。実験途中とはいえ、1998 年 3 月中旬には合宿が既に終了していることから、この点に関しては失敗した。

全体的に証明書発行総数が少ない理由としては、7 月実験と内容がほとんど変わらないため新たな関心が得られにくかったこと、実験に参加できない環境を考慮して証明書なしでも同じ情報が見えるようにせざるをえなかったことが影響したと考えられる。実際、証明書なしである情報へアクセスした数は約 7,600 件であるのに対し、証明書を用いて同じ情報へアクセスした数は 170 件程度であった。既存のシステムを利用した実験を進める上での困難さが表れたといえる。

はじめて実験に参加するメンバがほとんどない理由としては、本人確認手続きの手順 (節 2.4.4の (2)) に問題があるからと考えられる。7 月実験時も、節 2.4.4の (2) にしたがって本人確認手続きをしたメンバは非常に少なかった。今後は、担当ボードにいくつかの本人確認手続きを説明し、あらかじめどの方法をとるか調査し具体的にガイドする必要がある。

実験ガイドに関しては、付録 6.3に示す通り画面イメージを利用したガイドを用意した。実験ガイドのトップページへのアクセス数は約 200 件であり、証明書申請画面へのアクセス数は約 100 件であった。実際の証明書発行数が 60 程度であることから、実験ガイドは見たがその先に進まなかったか、途中でやめた可能性はある。手続きに関する個別の問い合わせは 7 月実験時に比べほとんどなかったが、このような数値や数人から得た感想をみる限りスムーズに証明書が発行できたとは言い切れない。実験ガイドを作成する側にとっては、最後まで読んでもらうためにはできるだけコンパクトにまとめようと考える傾向があるが、手順の説明だけでなくしくみに関係する補足的な説明を適度に加える必要があり、課題が残る。

3.6.2 証明書を 2 個以上発行する理由に関する考察

証明書発行総数に比べて、一人で証明書を 2 個以上発行している割合は 2 割程度となっている。その理由としては、Netscape ブラウザを複数のプラットフォーム上で利用したい、あるいは異なるバージョンを含めて利用したいという理由によるものが多い。これらの状況では今まで使用してきた証明書を廃棄したいわけではない。[1]

また、設定ファイルが消えるなど環境を失ったという理由による再発行は、発行総数の 3 % となっている。この場合の事態の深刻さはプロトコルによって異なる。S/MIME とし

[1] Netscape ブラウザ自身、7 月実験時よりもバージョンアップが進み、最新バージョン (実験時 4.04) では異なるプラットフォーム間で秘密鍵と証明書をコピーできる。

て利用していれば証明書を廃棄し CRL を発行して通信相手に知らせないと、暗号メールの受信に支障が出る。しかし、SSL のクライアント認証ではサーバに対して毎回証明書を提示しているため、新しい証明書が発行されれば実用上の問題はあまりない。

今までのところ、秘密鍵を盗まれたといった特に深刻な理由による再発行はない。深刻でない理由による再発行手続きを簡略化する意味では証明書を 2 個以上発行できる機能は有効である。しかし、もしも深刻な事態が発生した場合には、オフラインによる連絡手段を使って個別に対処し CRL を発行する必要がある。その際の本人確認手続きについては今後の課題である。

3.6.3　既存のツールの機能に関する考察

Netscape ブラウザ 4.03 以上については SSL クライアント機能に加え、S/MIME 機能を使えるように証明書フォーマットを対応させた。追加実験に参加したメンバは、WIDE メンバ同士で S/MIME による暗号メールをやりとりできた。また、WIDE メンバ以外の人とは、以下の準備をした上で暗号メールをやりとりできることが確認された。

- 相手の証明書を発行した CA の証明書をあらかじめ Netscape ブラウザに組み込むこと
- 一度署名付きメールを交換しあうこと

また、分科会内で行なった実験により、Netscape ブラウザ 4.03 以上では、現在使っている証明書とは有効期限とシリアル番号のみが異なる証明書を組み込むと証明書を更新できることも確認された。これにより、実験時に発行した証明書を継続して定常サービスに移行できることがわかった。

3.6.4　Web サーバの証明書発行について

moCA はユーザに対してのみ証明書を発行したのではなく、1998 年 3 月合宿のメンバ限定情報を提供する Web サーバを SSL 対応にする過程として、Web サーバに対する証明書を発行した。

また、合宿申し込みシステムを提供する別の Web サーバについては、SSL の暗号化機能 (とサーバ認証機能) のみを提供したいという要求があり、この Web サーバに対する証明書も moCA が 発行し、合宿システムとの連携を図った。

実験では、ユーザに対する証明書の発行が主体であったが、SSL の暗号化機能を利用するために Web サーバの証明書を必要とするケースは今後も考えられるため、Web サーバの証明書発行サービスの定常化に関しても検討したい。

第 4 章

電子メールの到達性を利用した本人認証

CA は電子的な身元を保証するサービスで、CA に身元の保証を依頼する際に何らかの形で本人認証をする必要がある。認証は最終的な電子的な識別子 (たとえば電子メールアドレス) と、身元保証をしたいもの (電子メールの受取人) を関係付けるものである。その際に「どこまで確実」に関係付けを保証するかという問題がある。

WIDE 内での第一回の実験は「確実」性を高く維持するために、ボードによる「首実検」による認証を行った。この方式は、「WIDE のメンバである」事は「ボードによる承認を経てメンバになる」という処理を行うモデルをそのまま CA の認証プロセスに組み込んだものである。「WIDE メンバであること」を確実に確認する面では「高い確度」を提供できる反面、認証のプロセスが「人」を介して「確認」することを要求し、認証のプロセスの自動化が難しい。また「人」が介在するため処理速度が遅い。特に、CA のオペレータに負荷がかかる傾向があり、認証すべき母集団が大きいと破綻する (WIDE は約 400 名の構成員を持ち、短期間で登録を行える限界の大きさではないかと思う)。

第二回の実験では、第一回の実験の反省に基づき、電子メールの到達性により認証を行った。WIDE メンバの ID は電子メールアドレスであり、証明書には電子メールアドレスを入れて発行している。証明書の一部として、電子メールアドレスが入っているために、「実際に使う電子メールアドレス」と「電子メールの受取人」が同一であることを保証すれば、WIDE 内での第二回目の実験/VeriSign Class 1 Digital ID/富士ゼロックスでの実験において使っている「アクセスキー」/「デジタル PIN」/「チケット」を申請したユーザの申請した「電子メールアドレス」に送ることにより、本人認証を自動的/機械的に行うことが可能になる。

「実際に使う電子メールアドレス」と「電子メールの受取人」が同一であることの保証は、第一回の実験で「首実検」をしたユーザに関しては「WIDE のメンバ」と「電子メールアドレス」が同一であることを保証している。保証されている電子メールアドレスにたいして、moCA は「アクセスキー」を電子メールで送る。証明書は申し込みを行った時点で発行されるのではなく、申込人が電子メールを使って「アクセスキー」を得て、「アクセスキー」を指定された URL で入力することによりはじめて発行される。このモデルは、日本におけるパスポート発行のメカニズムに良く似ている。個人証明書の発行/パスポートの発行ともに、「個人」と「住所 (個人証明書の場合は、電子メールアドレス)」を関連づけて

いる。

表 4.1: 本人確認の比較

	認証の根拠	宛先の根拠
パスポート	戸籍抄本	住民票
moCA 個人証明書	ボードの承認	ボード承認済電子メールアドレス

証明書の中に電子メールアドレスが組み込まれているため、不正な電子メールアドレスを指定しても使うことができない。

また他人の電子メールアドレスを指定した場合、「アクセスキー」の入手ができず証明書を得ることができない。

詐称された電子メールアドレスを持った WIDE メンバに「アクセスキー」が送られるが、このメンバが証明書を取得することは可能ではあるが、申請を行ったブラウザを使わない限り、申請時に計算した秘密鍵を得る手段がないため (Internet Explorer/Netscape Communicator ともに「秘密鍵のみ」を取り出す手段は Microsoft/Netscape は提供していない)、公開鍵と秘密鍵のペアをそろえることができず個人証明書を使うことができない。

申請者が一時的に他人の電子メールアドレスを使うことができ、「アクセスキー」を電子メールで得ることができた場合、S/MIME など電子メールを使わずに Web サーバに対してのアクセスに個人証明書を使う場合は、証明書を使い続けることができる。しかしながら、証明書の有効期限近くに「証明書の期限切れ/延長処理のお知らせ」を行う場合が多く、その時点で「電子メールアドレスが一時的に他人に使われた」ことが発覚する。この場合、悪用された期間を「個人証明書の有効期間」に容易に限定でき、どの情報が「不正アクセス」されたかを特定できる。

VeriSign の Class 1 Digital ID サービスは、単に「VeriSign の Class 1 Digital ID サービスの名前空間内でのメールアドレスの一意性」のみを保証している。VeriSign の Class 1 Digital ID サービスを利用した場合、ユーザは自分の電子メールアドレスを指定する。また、申請画面にて申請者が指定する「チャレンジ」を入力させる。この「チャレンジ」は、申請者が証明書を「Revoke」/「Replace」など発行された証明書に対して何らかの変更を加えるときに必要となる。VeriSign は、申請者が指定した電子メールアドレスに対して WIDE と同様の「デジタル PIN」を送る。WIDE の「アクセスキー」と同様に、VeriSign の場合は「デジタル PIN」を使って個人証明書のダウンロードを行う。

第 5 章

おわりに

われわれは、既存の CA 運用パッケージやアプリケーションを利用して現実にどの程度の運用やサービスができるかを試した。特に一般的なパッケージソフトでは実現困難な証明書発行の手続き面の実験を重視した。

2 回の実験を通じて、最適な証明書発行手続きが得られたかどうかについては、明確な結論が出なかった。但し、組織の柔軟性と本人確認手続きの困難さとの相関など有意な知見が得られている。

実際、「WIDE メンバである」ことを確認するだけの、それほど厳密とはいえないレベルの本人確認であっても、当初に決めた通りには実行しきれなかった。インターネット上のサービスを利用するためのユーザ登録において、どの程度の本人確認手続きが必要か、といったことは今まであまり追求されたことがなく、あらたまって本人確認手続きを行なうことに対して、ユーザもサービス提供側も慣れていないのが実状である。今回このような手続きに拘ったのは、従来より認証が強化されると期待されている技術であっても最初の登録時の本人確認レベルがアプリケーションの要求に見合わなければ意味がないと考えたからである。これは、公開鍵暗号を利用した認証技術だけの問題ではなく、使い捨てパスワード、バイオメトリクスを利用した一般の認証技術にいえることである。

本人確認後の証明書発行ではメンバに混乱が見られた。原因はおもに 2 種類のパスワードが一連の証明書発行手続きに出てくることによると考えられる。2 種類とは秘密鍵を保護するためのパスワードと、証明書発行用パスワード (あるいは証明書組み込み用パスワード) である。サービス提供側としては、パスワードの意味を説明したり、パスワードを減らすように努力するとともに、今後ユーザがいくつも証明書を持つことになっても、秘密鍵を保護するためのパスワードは 1 つだけ覚えればすむような鍵管理の統合機構を検討する必要があるだろう。

CA および アプリケーションを実現する既存のツールの実用性については、動作不安定な点、機能的に不十分な点およびユーザインターフェイスの問題点がいくつかあり、一般ユーザ向けのサービスとして提供できるレベルではない。しかし、7 月実験ではブラウザを Netscape 社製に限定したにもかかわらず、100 名以上 (1/4 強) の実験参加を得ることができた。また、実験開始当初よりツールの機能やユーザインターフェイスは改善されつつある。したがって、多数の参加者を見込んだ実験を行なうことが可能な環境になってきた

といえる。

CA を基盤としたシステムは大規模なシステムに有効といわれるが、それを実感できるまでにはクリアしなければならない種々の課題がある。特に、既存のシステムに CA を導入する際は、手続きが従来より複雑になるとユーザの抵抗感が大きくなることがあらためてわかった。今後は、CA を基盤とした新しいアプリケーションを開発するか、3 つ 4 つのアプリケーションに共通に適用できる環境を用意するか、といったしかけを考えて再度挑戦するべく、検討を進める予定である。

第 6 章

付録

6.1 moCA 仕様

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

6.1.1 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	ICAP1.0 + カスタマイズ (7 月実験時) ICAP2.2 + カスタマイズ (追加実験時)
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-0.8.0

SSL 化しているのは、SSL 暗号化機能を利用し証明書発行用パスワード入力時の盗聴を防止するためである。

6.1.2 CA の位置付け

図 6.1に示すように、階層下に位置づけた。理由は、共通の上位 CA を持つ場合でも、他の CA が発行した証明書では実験時のターゲット情報にアクセスできないことを確認したかったためである。

1997 年 9 月 15 日に IPRA の有効期限が切れた際には証明書の有効性確認時に影響が出て、証明書の更新が行なわれるまでの 10 日間、ターゲット情報にアクセスできなくなった。

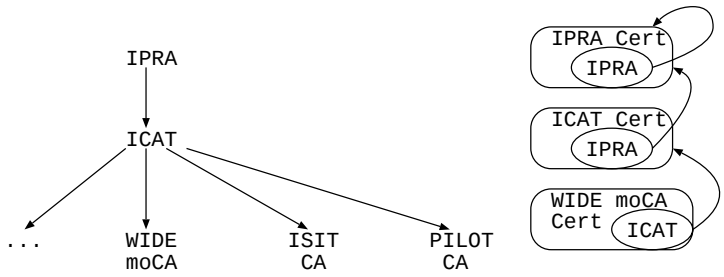


図 6.1: moCA の階層上の位置づけ

6.1.3　証明書フォーマット

	7 月実験時	追加実験時
X.509 のバージョン	1	3
ユーザ識別子 (DN)		
Country	JP(固定値)	JP(固定値)
Organization	WIDE Project(固定値)	WIDE Project(デフォルト値)
Organizational Unit	(自由に記入)	(自由に記入)
Common Name	WIDE ML に登録されている電子メールアドレス	氏名
Email(PKCS#9 定義)	(使用せず)	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	(バージョン 1 のため使用せず)	
basic Constraints		not CA (0)
certificatePolicies		ICAP 固定値
authorityInfoAccess(*)		ICAP 固定値
cRLDistributionPoints		ICAP 固定値
netscape-cert-type(**)		SSLclient および S/MIME

(*) ICAT 独自

(**) Netscape 独自

追加実験時の証明書内容例:

```
Version No = 2
Serial No = 20
Validity        from 980227103939Z
                to   980611000000Z
issuer:
        C=JP
        O=WIDE Project
        OU=members only CA
subject:
```

```
        C=JP
        O=WIDE Project
        OU=NEC Corporation
        CN=Mine Sakurai
        emailAddress=m-sakura@ccs.mt.nec.co.jp
signature:
        md5WithRSAEncryption
publickey:
        alg = rsaEncryption
basicConstraints:
        not critical
        not CA
certificatePolicies:
        not critical
        CertPolicyID: ICAT CPS
        PolicyQualifierInfo:
                policyIdentifier: id-pkix-cps
                qualifier: http://www.icat.or.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
```

6.1.4 1 メンバが発行できる証明書の個数

7 月実験時は 5 個まで、追加実験時は 10 個まで自由に証明書を発行できるようにした。ただし、追加実験時は、2 個目から証明書発行理由の選択を必須とした。

6.2 証明書発行用アカウント申請用紙 (例)

```
------------------------------------------------------------------------
WIDE moCA 登録申請用紙
                                                    1997 年    月    日
*印のついた項目のみ記入し、担当ボードと一緒に CA オペレータのところま
で来て下さい。
*       氏名:
*       所属:
*       e-mail:
        (wide@wide に登録されている e-mail アドレスを記入して下さい。)
        パスワード:     7-=N@XLq
        身元確認方法 (確認時 CA オペレータが記入):
                        □ 学生証、社員証、身分証明書
                        □ (各種) 免許証 + 名刺
                        □ パスポート + 名刺
                        □ クレジットカード + 名刺 + 写真
                        □ 健康保険証 + 名刺 + 写真
```

□ ボードが保証
□ その他 ----------

担当ボード署名: ----------

---------- 切 り 取 り 線 ----------

1997 年　月　日

WIDE moCA 登録申請 (控)

証明書発行時に利用するアカウント名 (e-mail アドレス) とパスワードは以下になります。大切に保管して下さい。

*　e-mail:
(wide@wide に登録されている e-mail アドレスを記入して下さい。)

パスワード:　7-=N@XLq

※登録作業終了の通知後、以下の URL へアクセスして下さい。
<URL:http://moca.wide.ad.jp/index.html>

※証明書は、1 人あたり 5 個まで発行できます。

6.3　実験ガイド

6.3.1　実験概要

図 6.2 〜 図 6.3 参照。

6.3.2　個人証明書の取得

追加実験用に作成したガイドのうち、初めて実験に参加する場合の個人証明書の取得ガイドを示す。


WIDE
moCA
Welcome to WIDE moCA!!
Web アクセスコントロールの実験 (~ 6/12)
moCA の証明書を使うのに必要なもの：
個人証明書の取得に必要なもの：

図 6.2: 実験準備

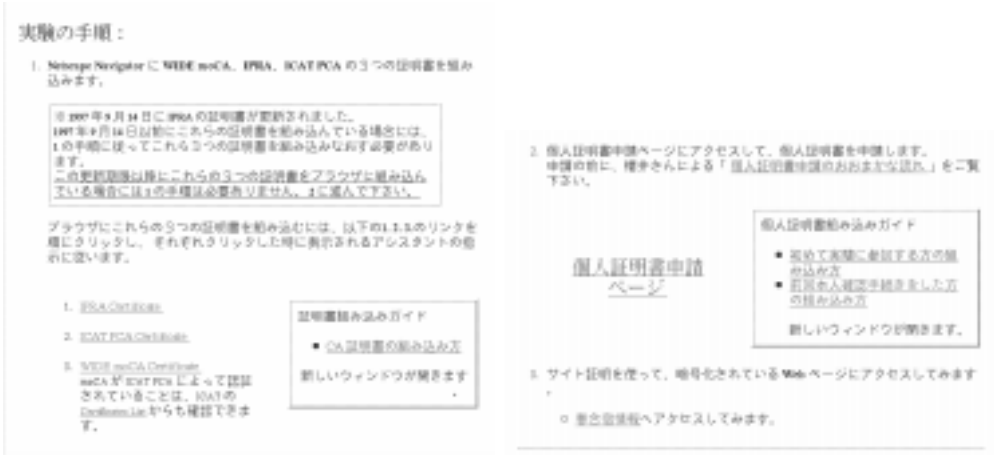


図 6.3: 実験手順

個人証明書の取得の仕方と組み込み方
～ 初めて実験に参加する場合 ～

1. 個人証明書として申請する内容を入力する (図 6.4)。

図 6.4: 証明書を申請する

アルファベットを使って入力します。 入力し終わったら「Send Request」をクリックします。

2. 申請内容の確認をする (図 6.5)。

図 6.5: 証明書申請内容の確認

先程入力した申請内容を確認します。 登録されているパスワードを入力してから「Issue Certificate」ボタンをクリックして下さい。

> 登録されているパスワードとは、 CA オペレータによって登録された、あなた用のパスワードのことです。 このパスワードは、あなたが担当ボードを介して CA オペレータに知らせておくものです。
>
> 詳しくは、「パスワードに関する説明」の (A) 証明書発行用パスワード をご覧下さい。
>
> ※ CA オペレータにあなたのパスワードを渡す方法については、オフラインを一部利用する方法 などをご覧下さい。

3. グレードを選択する (図 6.6)。
   今は 512 しか選べません。

「Issue Certificate」をクリックすると、ブラウザのダイヤログボックスが 現われ、秘密鍵を生成する手順に入ります。

4. 秘密鍵を生成する (図 6.7)。
   ブラウザが秘密鍵を生成します。

   秘密鍵についての説明を読む (図 6.8)。
   「詳細...」をクリックするとブラウザが生成する秘密鍵についての説明が表示されます。説明を読んだら「ＯＫ」をクリックし、4 の画面に戻って「ＯＫ」をクリックします。

5. ネットスケープパスワードを決める (図 6.9)。

図 6.6: 鍵長の選択

図 6.7: 鍵作成

秘密鍵を使う際に入力するネットスケープパスワードを入力します。

ネットスケープパスワードに関する説明を読む (図 6.10)。

図 6.8: 鍵作成 (続き)

図 6.9: 秘密鍵保護用パスワードの決定

5 の画面で「詳細...」をクリックするとネットスケープパスワードに関する説明が表示されます。
説明を読んだら「ＯＫ」をクリックし、5 の画面に戻って「ＯＫ」をクリックします。

> ネットスケープパスワードとは、ブラウザに組み込んだ秘密鍵を保護するためにつけるパスワードです。 このパスワードは個人認証を行っている Web サーバにアクセスするときなどに 聞かれます。
>
> 詳しくは、「パスワードに関する説明」の Netscape パスワード をご覧下さい。

6. 個人証明書を確認する (図 6.11)。
表示される証明書を確認します。
確認したら「Netscape ブラウザに証明書を組み込む」をクリックします。するとブラウザのダイヤログボックスが開きます。
7. 証明書名を入力する (図 6.12)。

図 6.10: 秘密鍵保護用パスワードを決める (続き)

図 6.11: 証明書の確認

ブラウザの個人証明書の一覧のところで表示される、この証明書の表示名を入力します。「ＯＫ」をクリックすると、個人証明書の組み込みは終わります。

(··· 中略 ···)

12. ブラウザのメニューから証明書が組み込まれていることを確認する (図 6.13)。

Communicator の場合には、メニューバーの 「Communicator(C)」→「セキュリティ情報 (S)...」を選択し「証明書」 の「本人」をクリックします。「あなたの証明書：」のところに 7 で入力した名前の証明書があることを確認します。

以上

図 6.12: 証明書に名前をつける

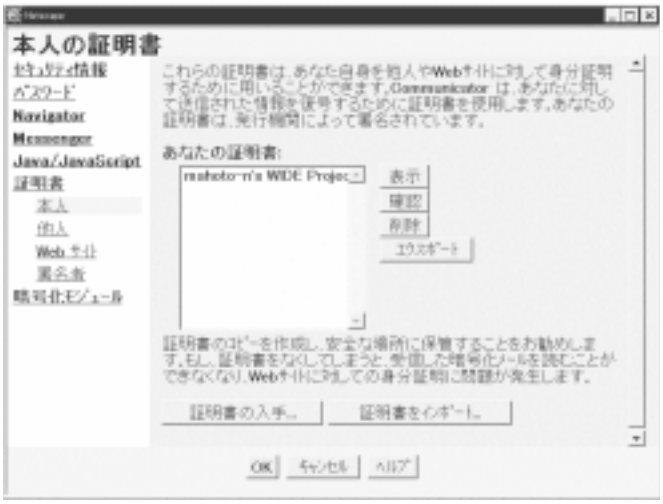


図 6.13: 証明書の組み込み確認