第 XIX 部

公開鍵証明書を用いた
利用者認証技術

第 19 部
公開鍵証明書を用いた利用者認証技術

第 1 章　moCA ワーキンググループ 2005 年度の活動

moCA ワーキンググループでは CA（Certification Authority）の振る舞いや証明書の扱いに注目し、オンライン CA である moCA（members oriented CA）の運用実験を行っている。この実験では WIDE メンバに対する証明書の発行・失効・更新を行い、利用環境や利用法に関する情報交換を行っている。

2005 年は、CA の運用体制の確立、鍵対の利用実験の継続、証明書の応用に関する情報交換という三つの目標を挙げて活動を行った。しかし 2005 年は、CA 証明書に関する既知の懸案事項が問題になったり、CA のサーバが故障して起動できなくなったりするなど、運用を維持するための対応作業が多く発生し、これらが活動の大部分を占めることになった。

下記に、おもな活動を三つの観点から報告する。

- 証明書利用上の改善
- 運用上の問題への対応
- MacOS&Safari ブラウザの対応

第 2 章　証明書利用上の改善

2.1 Web アクセスでの利用に関する改善

2005 年以降、moCA が発行している WIDE メンバ証明書は WIDE メンバ向けの Web ページにアクセスするための認証用として使われている。しかしこの Web ページへのアクセスログを調べたところ、1 月時点では 40%程度しか証明書が認証に使われていないことがわかった。つまり証明書の代替手段である共有パスワードが、高い割合で使われていることになる。

当初、これは WIDE メンバ証明書があまり使われていない状況を示していると考えられたが、Web ページの提供方法を工夫することで共有パスワードを利用できるようにしつつ証明書の利用を促すことができることがわかった。この工夫の結果、1 ヶ月間のアクセスのうち、約 80%で WIDE メンバ証明書を使った認証が行われることとなった（図 2.1）。この期間にユーザからクレーム等は寄せられず、パスワード形式の認証方式から電子証明書を使った認証方式に切り替える方法の成功例と言える。

ところで、moCA では発行した証明書のデータをすぐに消去するため、同一のユーザに証明書を再度配布するためには再発行を行う必要がある。2005 年は年間を通じて再発行の件数が少ないという傾向が

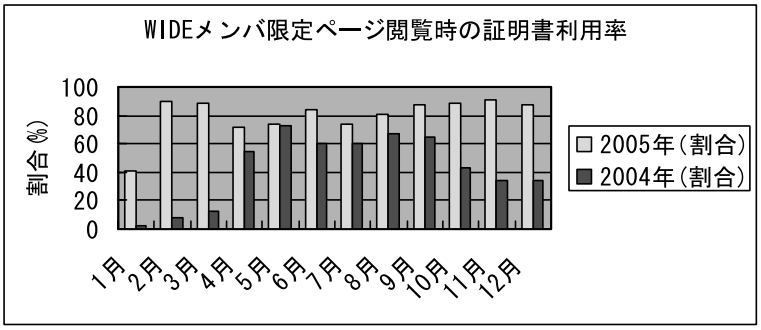

図 2.1.　WIDE メンバ限定ページ閲覧時の証明書利用率

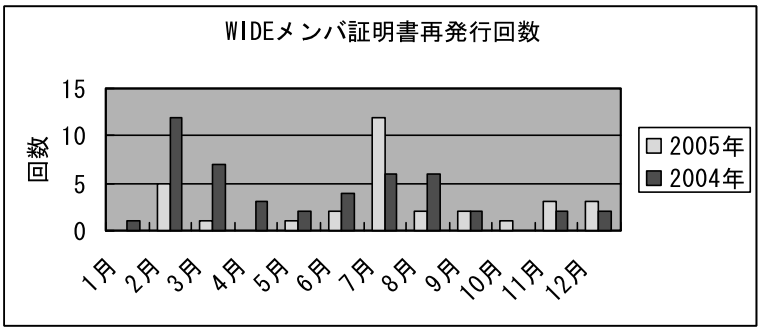

図 2.2.　WIDE メンバ証明書再発行回数

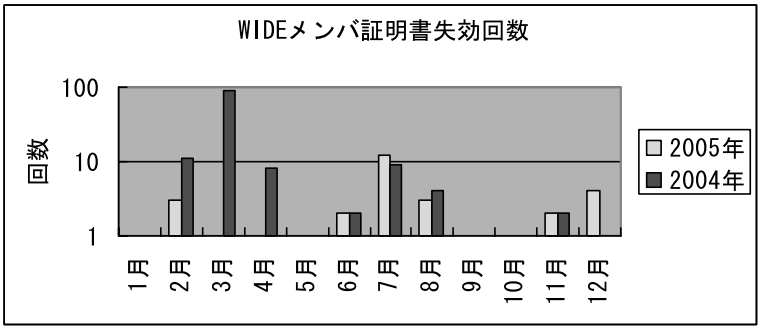

図 2.3.　WIDE メンバ証明書失効回数

見られた（図 2.2）。

2005 年の WIDE メンバ証明書の失効件数は、再発行と同様に年間の件数を合計すると 2004 年よりも少なかった（図 2.3）。2005 年は 1 年間の失効件数が、有効な証明書の数の 1 割に満たなかった。

2.2 S/MIME の利用に関する改善

S/MIME を利用するための、他の WIDE メンバの証明書の提供も行われるようになった。S/MIME を使って暗号化されたメッセージを作成するには、送信相手の証明書が必要となる。これまで WIDE メンバ向けにこのインタフェースは用意されていなかったが、CA ソフトウェアに付属している機能を使って Web インタフェースを使って提供できるようにした。今後、証明書が Web サーバによる認証用だけでなくユーザ間の認証に使われ、用途が広がることが考えられる。

第 3 章　運用上の問題への対応

(1) 秘書さん証明書の申し込み手順の問題（2 月）

秘書さん証明書の申し込みは、WIDE メンバからの発信を想定しており、moCA ワーキンググループの ML 宛てとしていたが、案内が不十分なため秘書さんが直接申し込みを行うケースがあった。この ML は投稿制限がかかっており、WIDE メンバでない秘書さんが申し込みを行っても、そのメールが moCA オペレータに届かず放置される件があった。そこで案内を改善し適切に申し込みが届くようにした。

(2) WIDE メンバ証明書配付時の MIME デリミタ問題（6 月）

WIDE メンバ証明書はメールの添付ファイルとして送られている。そのメールの本文には組み込み方法などの説明が書かれており、説明文の区切りに目印となる文字列が使われている。この文字列が、ユーザのメールソフトによって MIME の区切り文字列

と判断されてしまうことがあった。そのメールソフトではWIDEメンバ証明書が添付されていないように見えてしまった。

そこで、説明文の区切り文字列を変更し、区切り文字列と区別されるようにして、10件ほど再発行した（図2.2の7月分参照）。

(3) サーバ証明書再発行のWebサーバの証明書有効期限切れ問題（7月）

moCAではサーバ証明書の更新の際、そのサーバ証明書の管理者が自分自身でその申請を行い、再発行できるようになっている。そのインタフェースを提供しているWebサーバの証明書の有効期限が切れており、アクセスできないという不具合が起った。

このサーバのサーバ証明書を更新してアクセスできるようにすると共に、このインタフェースを提供しているプログラムを改造し、有効期限が切れた後でもサーバ証明書を再発行できるようにした。有効期限が切れた後でも1ヶ月前までさかのぼって再発行できる。

(4) CA証明書配付用のファイルの拡張子問題（7月）

CA証明書は、通常WIDEメンバ証明書に添付して配付しているが、WIDEメンバ以外のユーザに向けても提供しているつもりであったが不十分であった。

ユーザに公開しているCA証明書のファイルは、certという文字列が最後についたファイル名で提供されていた。しかしこのファイル名はWindows等で証明書ファイルとして識別されないため、CA証明書のインストールを行うためにファイル名を変更したり、Webブラウザ等で明示的にインストールしたりする必要があった。そこで証明書ファイルの拡張子を.cerに変更し、Windowsではダブルクリックだけで内容が表示されるようにした。WebサーバのMIME typeの設定も変更され、ダウンロードの際に証明書ファイルであることが認識されやすくした。

この修正によりWindowsでは証明書の内容を表示するダイアログボックスにインストールのボタンが表示されるため、比較的容易にインストールを行うことができるようになった。

この時にMLでよく使われている拡張子の種類と内容に関する情報交換が行われたが、CA証明書の組み込みを容易にするためのガイド等の作成は課題として残ることになった。

(5) moCAハードウェア故障による運用停止問題（7月）

moCAのハードウェアが故障し、起動できなくなった。HDDは無事でデータの移行ができたため、別のハードウェアに移して仮運用することとなった。

moCAのプログラムが動作する環境を構築にあたり、WIDEメンバの多大な協力があり、約3日間の運用停止で収拾を図ることができた。

(6) CA証明書の符号化問題（8〜9月）

WIDEルートCAとmoCAの証明書をMacOS付属のWebブラウザであるSafariで検証するとCA証明書でないように扱われてしまうことがわかった。調査の結果、X.509v3拡張フィールドのBasic Constraintsに含まれる値がDER方式ではなく、BER方式で符号化されていたことに起因することがわかった。

DER方式で符号化されていることを期待するプログラムで具体的に不具合が生じたため、DER方式で符号化した値に入れ替えたCA証明書を発行した。この対応にともないCA証明書のフィンガープリントをユーザに再周知した。

(7) Safariを使ってWIDE合宿申し込み用Webサーバにアクセスできない問題（9月）

WIDE合宿のWebサーバに不具合のあるCA証明書が設定してあり、Webサーバから送られるCA証明書を優先的に使用するWebブラウザを使うとアクセスできない問題が起ることがわかった。

CA証明書の不具合は有効期限が切れていたためで、このCA証明書がなぜWebサーバに設定されていたのかは不明。CA証明書を正しいものに置き換えることで対応した。

(6)と(7)は絡み合った問題であったため、MacOS&Safariブラウザの問題として次章で詳述する。

第4章　MacOS&Safariブラウザの対応

4.1 概要

PKI（Public Key Infrastructure）技術を用いたWIDEプロジェクト内部向け認証インフラの一つであるmoCA（members oriented Certification

Authority）では、WIDE メンバに対して毎年 WIDE メンバ証明書を発行して配付している。WIDE メンバが普段利用している端末の OS は、Windows、UNIX、MacOS などさまざまであることから、WIDE メンバ証明書はさまざまな OS とブラウザの組み合わせで利用されてきた。しかし、以前から MacOS 上でブラウザに Safari を用いて（以下、MacOS&Safari ブラウザと表記）、WIDE メンバ証明書を利用すると特定の Web サーバへのアクセスに失敗する場合があることが報告されていた。この問題を解決すべく 2005 年 8 月に調査を行った結果、WIDE で発行している CA 証明書の符号化方法が正しくないことが主因であることが判明した。

本問題への対処として CA 証明書の再発行が必要となったため、moCA WG では CA 証明書の再配付による混乱を避ける方法を検討した。その結果、WIDE ROOT CA 証明書、moCA 証明書についてそれぞれ正規の証明書を 2 種類利用するという形での運用を行うことで解決できることが判明したため、2005 年 9 月に上記手法を用いての対策を実施した。

4.2 MacOS&Safari ブラウザの組み合わせで Web アクセスに失敗する問題

WIDE メンバ証明書を使って Web アクセスが行えるよう設定されている Web サーバには、WIDE メンバ専用サーバ（`https://member.wide.ad.jp/wide-confidential/`）や WIDE 合宿申し込み用サーバなどがある。かねてより、MacOS&Safari ブラウザの組み合わせで操作した場合、WIDE 合宿申し込み用サーバへのアクセスが拒否されると報告されていた。そのため、WIDE 合宿申し込み用サーバの設定チェックを一度実施したが、原因はつかめなかった。具体的な不具合を下記に記載する：

(a) CA 証明書を MacOS の Keychain に登録していても、（証明書を利用しようとするたびに）無効な機関により署名されていますという警告表示が出る。

(b) WIDE 合宿申し込みサーバへのアクセスだけ拒否される（他のサーバを利用する場合には、アクセスできる）。

この問題のため、MacOS を使っている人が WIDE メンバ証明書を使って WIDE 合宿申し込みを行うには、例えば Mozilla などの別のブラウザをインストールする必要があり不便な状況となっていた。

4.3 調査および分析

4.3.1 証明書の符号化の調査

2005 年 8 月、WIDE の CA 証明書とそれ以外の機関の CA 証明書を比較していたメンバから、WIDE の CA 証明書のフィールドのうち、X.509 の basicConstraints 拡張フィールドの符号化が誤っているのではないかと指摘があった。

basicConstraints 拡張フィールドとは、当該証明書が CA 証明書か否かを示すためのフィールドで、RFC3280[109] では下記の ASN.1 表記で定義されている：

```
BasicConstraints ::= SEQUENCE {
    cA                  BOOLEAN
                        DEFAULT FALSE,
    pathLenConstraint   INTEGER (0..MAX)
                        OPTIONAL }
```

このフィールドではその証明書が CA 証明書である場合、BasicConstraints.cA の値を TRUE とする必要がある。

X.509 形式の証明書では、拡張フィールドを DER（Distinguished Encoding Rule）形式で符号化することとなっており、DER 符号化に関して定義している X.690[121] によれば、BOOLEAN TRUE は 0xff と符号化しなくてはならない：

“If the encoding represents the boolean value TRUE, its single contents octet shall have all eight bits set to one.”（[121] 11.1 Boolean values より）

ところが、WIDE の CA 証明書では、BOOLEAN TRUE を DER 形式ではなく BER（Basic Encoding Rule）形式による符号化にしたがって 0x01 と符号化していた。本件は Challenge PKI 2001[142] に参加した際にも、個別に指摘を受けたことがあったが、当時は具体的な不具合が見られなかったことから、WIDE ROOT CA や moCA の証明書について修正をしていなかった。今回の指摘により、MacOS における証明書ライブラリの BOOLEAN 解釈が厳格であり、WIDE ROOT CA 証明書や moCA 証明書が「CA 証明書ではない」と解釈されて Web アクセス自体を拒否されている可能性が高いことがわかってきた。

そこで、まずはワーキンググループ内での実験として、WIDE ROOT CA 証明書や moCA 証明書につ

いて basicConstraints 拡張フィールドの符号化を修正した版を作成し、MacOS&Safari ブラウザの環境にインストールして Web アクセスを試みた。RFC 3280 の 6.1.1 に記載されている内容から、ROOT CA 証明書の解釈については特別扱いをしている可能性があるため、moCA の証明書のみを修正すれば不具合が解消されると実験開始前は予想していたが、結果として WIDE ROOT CA および moCA の証明書の両方とも修正する必要があった。以上で、(a) の解決策が見つかり、(a) を解決すると、(b) も解決されることがわかった。

4.3.2 サーバ設定の調査

(b) の解決のため、WIDE 合宿申し込みサーバの設定を 2005 年 9 月に調査した。その結果、サーバ側に設定されている moCA 証明書が古く、有効期限切れとなっていることがわかった。そのためサーバに設定する CA 証明書を符号化修正版にし、期限切れの物を更新することで問題を解決した。なお、有効期限切れとなっていた moCA 証明書は、テスト用に臨時発行されたもので正式な証明書ではなかった。サーバの SSL 設定の際に誤って配付した可能性があるが原因を究明するには至らなかった。

4.3.3 問題の分析

(a) と (b) の問題は相互に絡みあっていた。たとえば、SSL のクライアント認証時には、サーバ側に moCA 証明書が必要になる。クライアント認証の際、moCA 証明書はクライアントから送信されるか、サーバの証明書データベースに登録されているものが使われる（どちらを優先するかは実装による）。(a) により、moCA 証明書は CA 証明書と認識されていないため、クライアントからは送信されない。この場合、サーバの証明書データベースの moCA 証明書が使われるはずだが、有効期限切れとなっており正しく利用できるものではなかった。そのため、クライアント認証の確認が失敗してしまう（図 4.1）。これを解決するには、(a) を解決するか、サーバ側に設定する CA 証明書を最新にするかが必要である。

また、moCA では、WIDE メンバ証明書だけでなく Web サーバの証明書も発行しており、サーバ認証でもクライアント認証でも moCA 証明書が必要になる。したがって、サーバ認証の場合もクライアント認証と同様に失敗する。サーバが Apache の場合、CA 証明書が有効期限切れになると、SSL 実行時にサーバからブラウザに対して CA 証明書を送信しないことを調査中に確認した。なお、サーバ認証に失敗した場合は、Apache サーバの設定によって

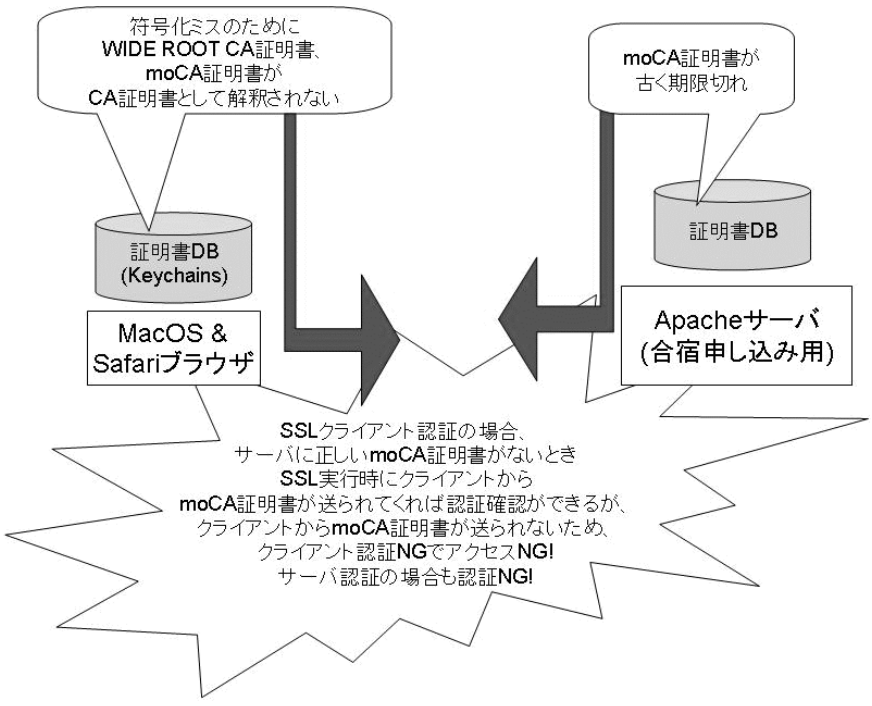

図 4.1. MacOS&Safari の問題分析結果

表 4.1. MacOS&Safari 問題の調査結果のまとめ

ブラウザに、符号化修正前の WIDE ROOT CA 証明書、moCA 証明書を登録したとき

	ブラウザによる CA 証明書解釈		サーバに設定されている moCA 証明書の状態（Apache サーバ）	
ブラウザの種類	WIDE ROOT CA	moCA	符号化修正前（通常）	期限切れ
Safari on MacOS X	CA でない（警告）	CA でない（警告）	サーバアクセス OK（警告は出る）	サーバアクセス NG
Internet Explorer 6.0 on Windows	CA	CA	サーバアクセス OK	サーバアクセス OK

ブラウザに、符号化修正後の WIDE ROOT CA 証明書、moCA 証明書を登録したとき

	ブラウザによる CA 証明書解釈		サーバに設定されている moCA 証明書の状態（Apache サーバ）		
ブラウザの種類	WIDE ROOT CA	moCA	符号化修正前（従来）	期限切れ	符号化修正版
Safari on MacOS X	CA	CA	サーバアクセス OK（警告は出る）	サーバアクセス OK	サーバアクセス OK
Internet Explorer 6.0 on Windows	CA	CA	サーバアクセス OK	サーバアクセス OK	サーバアクセス OK

ページを表示するかどうかを選択できるため即 Web アクセス NG となるとは限らない。

さらに、サーバから CA 証明書が送信される場合、MacOS&Safari ブラウザは、サーバから送信される moCA 証明書を優先して認証確認を行うことを確認した。そのため、ブラウザ側だけが moCA 証明書を符号化修正版にすればよい訳ではなく、サーバ側も同様に修正版にしないと警告表示は出ることがわかった。

これらの調査結果をまとめると表 4.1 のようになる。今回は、根本的に問題を解決するため、(a) の解決策から先に実施した。

4.4 対処方法

調査により、(a) の解決策として CA 証明書の再発行が必要となったが、この場合再配付の方法が大きな問題となる。CA 証明書は、WIDE メンバ証明書を毎年配付する際に一緒に配付しているため、同じ手段を使って配付することはできる。しかし、すべての環境で不具合が出る訳ではないため、WIDE メンバ全員に配付しなおすと混乱を招く可能性がある。

そこで、MacOS ユーザに向けて CA 証明書の修正版を提供する旨のアナウンスを行うが、それ以外のユーザは作業不要で従来どおりの CA 証明書を引き続き使えるようにした（この方法で問題が起きないことも実験期間に確認した）。つまり、WIDE ROOT CA 証明書と moCA 証明書に関してはそれぞれ 2 種類の証明書を正規に持つことになった。そのため、CA 証明書をブラウザにインストールする際に証明書の正当性を確認するためのフィンガープリント情報が 2 種類存在することになった旨のアナウンスも同時に行った（付録参照）。

4.5 まとめ

今回の検討と対策の実施により MacOS & Safari ブラウザの組み合わせでも警告が出ることなく WIDE メンバ証明書を利用できるようになった。また、付随する効果として MacOS X 標準で付いてくる Mail.app で S/MIME による暗号メールが WIDE/moCA 発行の証明書を用いて利用できるようになった。

不具合の主因と判明した X.509 形式の証明書の符号化問題は以前から指摘されていたが、具体的な不具合が出てもすぐには符号化問題との関係がわからず、具体的な調査に入る前に長時間が経過してしまった。CA 証明書の再発行と再配付は困難であるという気持ちが、余計に対処を遅らせることになった点は反省しなければならない。なお、今回は MacOS&Safari 固有の現象だけでなく、サーバ側の設定の問題も絡んでいたため問題が大きくなったが、Web アクセスに失敗するほどの問題は他ではほぼ起きないであろうと考えられる。

不具合の対処として必要となった CA 証明書の再配付については、運用方法との関わりで気づいた点があった。moCA WG の運用実験では、個人の証明書の有効期間を 1 年に設定した上で、ユーザに配付している。この更新した個人証明書を配付する際に、

必ずCA証明書（WIDE ROOT CAおよびmoCA）をも再配付している。これは、ユーザにとっては全く負担がかからず、最新のCA証明書をユーザに配付できるというメリットがある。このような運用方法を採用しているため、今回のようにCA証明書の再配付を必要とするような場合でも、最長でも1年でCA証明書が更新されることが期待できる。今後もさまざまな経験を積みながら、運用上合理的なCAを追求してゆきたい。

謝辞

PKIの普及という目標を理解してくださり、運用上の不具合にも関わらず調査へのご協力やフィードバックをしてくださっているWIDEプロジェクトの皆様に深く感謝いたします。

第5章　まとめ

運用上の問題が多発し、その対応に追われることとなってしまったが、問題の指摘のいくつかはワーキンググループメンバ以外からであったことが今までとの違いであった。トラブルシューティングにあたってもWIDEメンバからの数多くの支援を得ることができ、MLでの議論を通じて問題となる内容が明文化されたり、迅速な復旧が図られることとなった。2004年以降の“運用体制の確立”といった課題が2006年に持ち越しとなったものの、moCAがWIDEメンバの間で電子認証のインフラとして認知され、取り入れられつつあることが感じられた一年であった。

2006年6月は、ルートCA証明書の期限切れにともない、CAの鍵変更を予定している。スムーズな鍵変更のためにも、運用の信頼度を上げるためにも、効率的な運用を行うためにも、さまざまな改善と工夫を凝らしていきたい。

付録A　アナウンス内容（2005年9月5日）

(Japanese message is followed by English one.)

moCA WGより重要なお知らせです。

MacOS X & Safariをお使いの皆様へ

これまでWIDEメンバ証明書を使ったWIDE合宿申し込みページへのアクセスができないと報告を受けておりました。

原因がわからず対策を打てないままご不便をおかけしていましたが、このたび対処方法が見つかりましたので、対処内容をリリースさせていただきます。

他の皆様へ

下記不具合のない方につきましては、対処の必要はありません。ただし、今回の対処により、CA証明書、具体的には、WIDE ROOT CA証明書、moCA証明書についてそれぞれ正規の証明書が2種類という形での運用となります。従って、フィンガープリントを確認される場合には注意が必要となります。現在の段階では、正規の証明書が2種類有っても利用上の問題は発生しないことをmoCA WG内の実験で確認しております。

この2種類の正規証明書という形態は、あくまでも現在判明した問題点を解決するための経過措置であり、次回の証明書更新の際には、問題対処済みの証明書に置き換えることになります。

今後のWIDEメンバ証明書配付や、Webサーバ証明書配付では今回の対処で発行した「符号化問題対応版」のCA証明書を配付するように移行していきます。なお、Webサーバに設定しているCA証明書も、「符号化問題対応版」に入れ替えていただくように別途お勧めしていきます。

以上、よろしくお願い致します。

問題解決のためのガイド（日本語版）

［対処により不具合が解消される端末環境］

MacOS X & Safari

現時点ではMacOS X 10.3（Panther）及びMacOS X 10.4（Tiger）のみで確認しております。

［解消される不具合］

WIDEメンバ証明書を使ったWIDE合宿申し込みサーバ“`https://widecamp.e-side.co.jp/`”へ

のアクセスを拒否される

［今回の対処を実施するメリット］

MacOS X 標準で付いてくる Mail.app で、SMIME が利用できるようになります。

また、今後、（WIDE 関連の）SSL 対応 Web サーバで「符号化問題対応版」の CA 証明書設定が浸透すると、WIDE メンバ専用ページへの最初のアクセス時の警告「無効な機関により署名されています」が表示されなくなります。

［原因］

直接はサーバ側の設定が影響していると思われ、正確な原因究明がまだ必要です。しかし、端末側でも下記の問題があり、下記を解決すると、不具合を回避できることがわかりました。

（Internet Explorer, Mozilla 等では不具合が出ていないものの）WIDE ROOT CA 証明書、moCA 証明書の符号化に問題があり、MacOS X & Safari では、CA の証明書と認識できていなかった。

［対処方法］

WIDE ROOT CA 証明書と moCA 証明書の符号化問題対応版をリリースします。これらの CA 証明書の入れ替えを行ってください。WIDE ROOT CA 証明書を X509 Anchors に入れるのがコツです。

新しい WIDE ROOT CA 証明書のフィンガープリント
SHA1 フィンガープリント
be 97 ae 7f c0 37 d2 cb c5 f2 3b eb d3 2c f5 07 74 c3 ef fe

新しい moCA 証明書のフィンガープリント
SHA1 フィンガープリント：
27 fa 6b c3 25 6d 4f 0f 6b 3d f2 a5 b6 8a 83 0a 53 33 7f 45

(a) Tiger の場合

1. キーチェーンアクセスを使って、キーチェーン「ログイン」やキーチェーン「X509Anchors」から以前にインストールした moCA 証明書、WIDE ROOT CA 証明書を消す。

   キーチェーンアクセスは「アプリケーションフォルダー下のユーティリティーフォルダ」にあります。X509Anchors を変更する際には特権ユーザーのパスワードが必要となります。

2. 新しい moCA 証明書をキーチェーン「ログイン」に追加する。
   `http://moca.wide.ad.jp/moca-for-macos050818.cer`

3. 新しい WIDE ROOT CA 証明書をキーチェーン「X509Anchors」に追加する。
   `http://member.wide.ad.jp/wg/moca/wideroot-for-macos050822.cer`

(b) Panther やそれより前の場合

1. キーチェーンアクセスを使って、キーチェーン「ログイン」やキーチェーン「X509Anchors」から以前にインストールした moCA 証明書、WIDE ROOT CA 証明書を消す。

2. 証明書キーチェーンがない場合、「キーチェーンを追加」で、
   /System/Library/Keychains/X509Anchors（ルート CA）
   /System/Library/Keychains/X509Certificates（中間 CA）
   を追加。

3. 新しい moCA 証明書をキーチェーン「X509Certificates」に追加する。
   `http://moca.wide.ad.jp/moca-for-macos050818.cer`

4. 新しい WIDE ROOT CA 証明書をキーチェーン「X509Anchors」に追加する。
   `http://member.wide.ad.jp/wg/moca/wideroot-for-macos050822.cer`

［さらなる調査について］

今後、より正確な原因究明の調査を行い、別途レポートにまとめる予定です。

［関連 URL］

http://member.wide.ad.jp/wg/moca/
wide_root_ca.html
http://moca.wide.ad.jp/

日本語版は以上

——+——+——+——+——

This is an IMPORTANT notice from moCA Working Group.

MacOS X & Safari users,

Some people have reported to the moCA WG that they couldn't access to the WIDE camp application page. Through our investigation, one measure has been found and is released now.
So, please follow the guide described below.

All other WIDE members,

Any action is not required if you don't experience the same problem, but please let us announce that the WIDE ROOT CA and moCA each have two certificates, *** both of which are correct ***.
Just be careful when you confirm the CA fingerprints, because two kinds of fingerprints are displayed on the moCA WG web pages.
Any problem due to the existence of two kinds of CA certificates wasn't revealed through several experiments within moCA WG.

This status is temporal. By the next key update, the changeover will be gradually progressed and each CA will have only one certificate again.

Apology for your inconvenience,

moCA Working Group

——o——o——o——o——

The guide to fix the WIDE camp web page (https://widecamp.e-side.co.jp/) access problem

[the target PC environment]

MacOS X & Safari

the moCA WG has examined the versions of MacOS X 10.3 (Panther) and 10.4 (Tiger).

[the additional merit of this fix]

1. S/MIME with Mail.app can be used.
2. The warning message "This certificate is signed by invalid authority." won't be seen after a while when the WIDE members only page is accessed with Safari.

[the cause of the problem]

further investigation is needed, though it's anticipated something wrong is server side's configuration.

But encoding problem is found in the WIDE ROOT CA and the moCA certificates and these certificates aren't accepted as CA certificates in MacOS X and Safari environment. This fix of encoding problem is related to the fix of the access problem. So, the fix of encoding problem is released at first.

[the measure]

Replace the WIDE ROOT CA certificate and the moCA certificate to the new ones, which are called "encoding problem fixed version" certificates. The key point is to install the WIDE ROOT CA certificate to the "X509 Anchors".

The fingerprint of new WIDE ROOT CA certificate
SHA1 fingerprint:
be 97 ae 7f c0 37 d2 cb c5 f2 3b eb d3 2c f5 07 74 c3 ef fe

The fingerprint of new moCA certificate
SHA1 fingerprint:
27 fa 6b c3 25 6d 4f 0f 6b 3d f2 a5 b6 8a 83 0a 53 33 7f 45

(a) in case of Tiger

1. remove the old WIDE ROOT CA and moCA certificates using KeyChain access.

These certificates might be installed in “Login” or “X509 Anchors”.

KeyChain access is found in “utility folder” under “application folder”. The root privilege is required when “X.509 Anchors” is changed.

2. Please install the new moCA certificate to KeyChain “Login”.
   The URL is `http://moca.wide.ad.jp/moca-for-macos050818.cer`

3. Please install the new WIDE ROOT CA certificate to KeyChain “X509 Anchors”.
   The URL is `http://member.wide.ad.jp/wg/moca/wideroot-for-macos050822.cer`

(b) in case of Panther or older version

1. remove the old WIDE ROOT CA and moCA certificates using KeyChain access.
   These certificates might be installed in “Login” or “X509 Anchors”.

2. “Add KeyChain” if the certificate KeyChain doesn’t exist. add
   /System/Library/Keychains/X509Anchors
   (for root CAs)
   /System/Library/Keychains/X509Certificates
   (for subordinated CA)

3. Please install the new moCA certificate to KeyChain “X509Certificates”.
   The URL is `http://moca.wide.ad.jp/moca-for-macos050818.cer`

4. Please install the new WIDE ROOT CA certificate to KeyChain “X509 Anchors”.
   The URL is `http://member.wide.ad.jp/wg/moca/wideroot-for-macos050822.cer`

[related URL (japanese only)]
`http://member.wide.ad.jp/wg/moca/wide_root_ca.html`
`http://moca.wide.ad.jp/`

付録 B　フィンガープリントの一覧

概要

PKI は公開鍵を使った認証技術である。PKI を利用した認証は、公開鍵を使って作成された電子署名を検証することで行われるため、その公開鍵が正しく検証者に渡っていなければならない。WIDE の各認証局が発行した証明書を検証するには、検証を行うもの（検証者）が各認証局の証明書を原本と違わないように保持している必要がある。

moCA では、moCA 対応のサーバ（クライアント認証を有効にした SSL 等のサーバ）を認証したり、WIDE メンバ同士が電子メールのやり取りを行って相手を認証したりすることを証明書の利用場面として想定している。従って WIDE メンバは検証者であり、また moCA 対応のサーバも検証者である。

ここでは、WIDE における証明書の検証者が予め入手した認証局の証明書を検証することができるように fingerprint をまとめる。Web ブラウザの CA 証明書表示機能、または認証局の証明書を PEM 形式で保存し、OpenSSL を利用して表示する機能を使って、保持している証明書が原本と相違のないかどうかを確認することができる。

フィンガープリントの計算方法には SHA5 と MD5 の二種類がある。そのため後述する一覧では二種類の値を記述しておく。しかし、どちらか一方を使って確認するだけでよい。なお、Windows の証明書の表示では SHA1 の値が表示され、Netscape では MD5 が、Mozilla の場合は SHA1 と MD5 の両方が表示される。

フィンガープリント一覧

以下に、2006 年 1 月現在の各 CA 鍵のフィンガープリントを示す。WIDE ROOT CA および moCA は、正規の証明書が 2 種類あることに注意していただきたい。

WIDE ROOT CA（符号化問題対応版）

SHA1 フィンガープリント
be97 ae7f c037 d2cb c5f2 3beb d32c f507 74c3 effe

MD5 フィンガープリント
A0:17:BA:50:A4:FF:1D:32:3B:52:40:94:C5:
4F:C0:B0

WIDE ROOT CA
SHA1 フィンガープリント
3560 185D 83DC CBB7 0EBB 45AD 1E9B
F529 A816 0562
MD5 フィンガープリント
2B:68:BD:1B:26:28:2A:AC:CF:F3:45:90:1D:
6C:2A:9C

moCA （符号化問題対応版）
SHA1 フィンガープリント
27fa 6bc3 256d 4f0f 6b3d f2a5 b68a 830a 5333
7f45
MD5 フィンガープリント
1F:37:AA:3A:FE:5B:CF:C5:FC:37:46:7C:D6:
74:73:DB

moCA
SHA1 フィンガープリント
487E 16E1 746E 5C16 8A7D C55D DE80
37E8 9241 7FA3
MD5 フィンガープリント
17:FD:D2:8A:C2:36:5D:0E:0B:A7:69:BC:9D:
7F:E6:97

SOI CA
SHA1 フィンガープリント
0A92 34A8 B589 C835 6101 3151 CBC6 4F18
1ACE 6D4D
MD5 フィンガープリント
7B:23:02:D1:76:37:44:81:76:35:DA:8A:51:
BF:B5:48

AI3 CA（休止中）
SHA1 フィンガープリント
0AE8 76F5 7240 BA67 99B9 A200 C94C 1650
FBB5 29F0
MD5 フィンガープリント
19:89:C9:CF:D5:E1:8F:E1:65:51:92:72:A2:
49:96:0F