

# 個人証明書発行手続き

← 制御の流れ

← データの流れ

権限 状態、SID

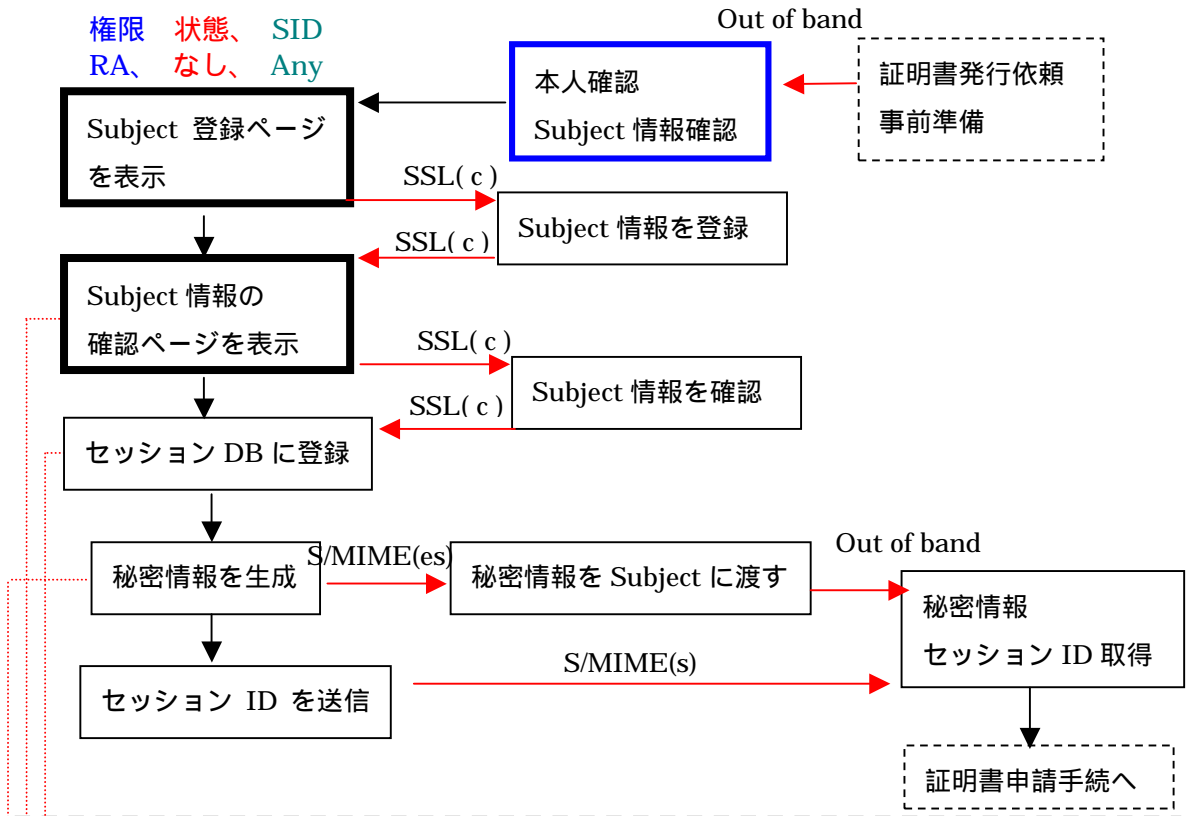
RA server

RA

Subject

## Scene1

subject さんが、窓口の RA さんに証明書発行依頼を行い、これを契機に RA さんが証明書発行手続を開始します。subject の事前準備には、CACAnetCA 証明書の組込などが含まれます。



subject メールアドレス	subject_name_j=>文字列,	Subjectの名前(漢字氏名)
subject 氏名	subject_mail=>文字列,	Subjectの電子メールアドレス(証明書用)
RA メールアドレス	subject_name=>文字列,	Subjectのローマ字氏名(DN生成用)
RA 氏名	ra_mail=>文字列,	RAのメールアドレス
環境変数から	ra_name_j=>文字列,	RAの名前(漢字氏名)
RA の DN	ra_dn=>文字列,	RA のDN
RA 証明書のシリアル番号	ra_serial_no=>整数,	RA 証明書のシリアル番号
	p12_password	P12用パスワード

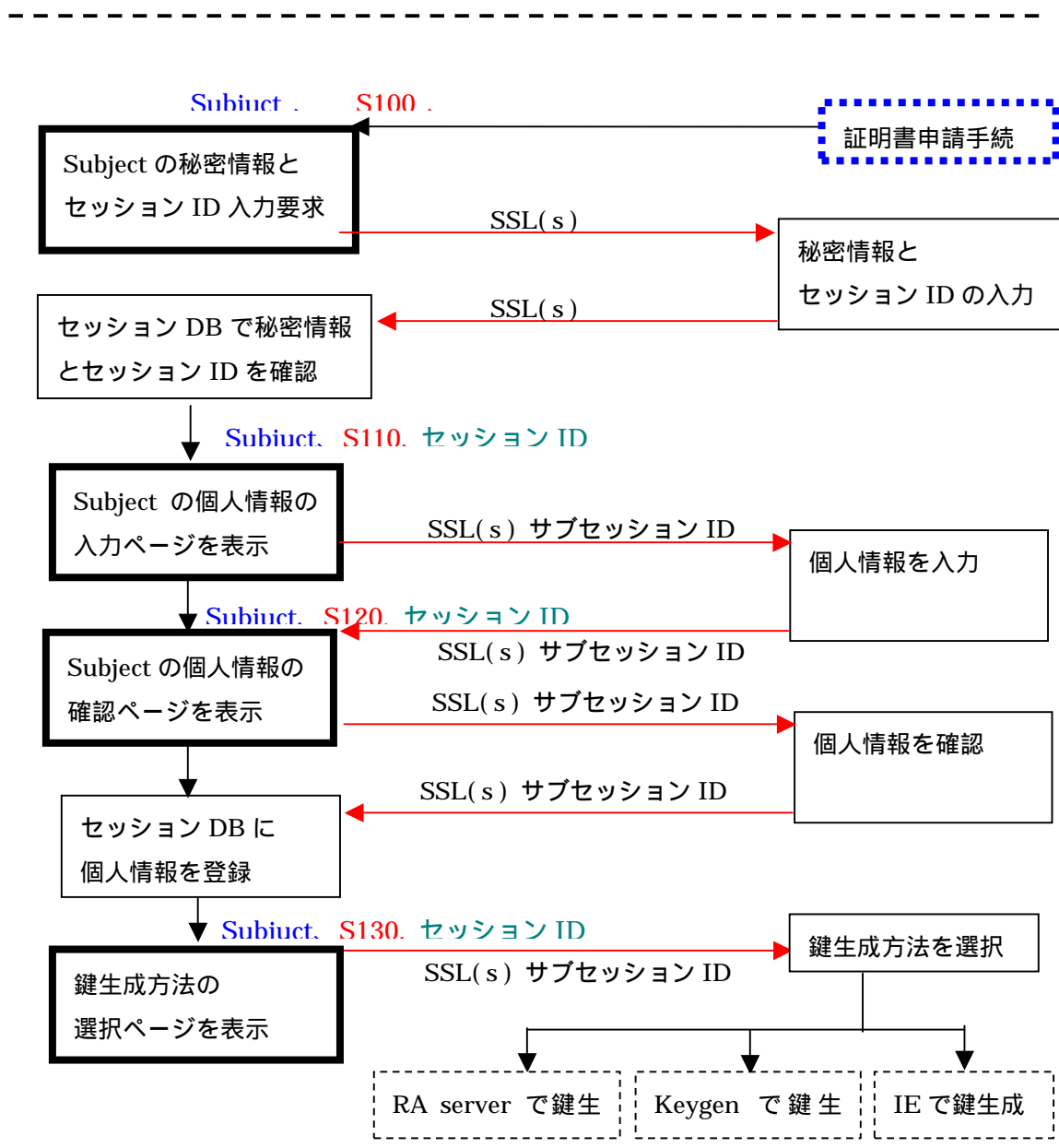
## Scene2 証明書申請手続き

subject さんは、自分が受け取った秘密情報とセッション ID を使って証明書発行システムにアクセスし、証明書の申請を行う。

入力する情報の内、証明書に関係するのは、自分の証明書に入れたいメールアドレスとローマ字氏名である。

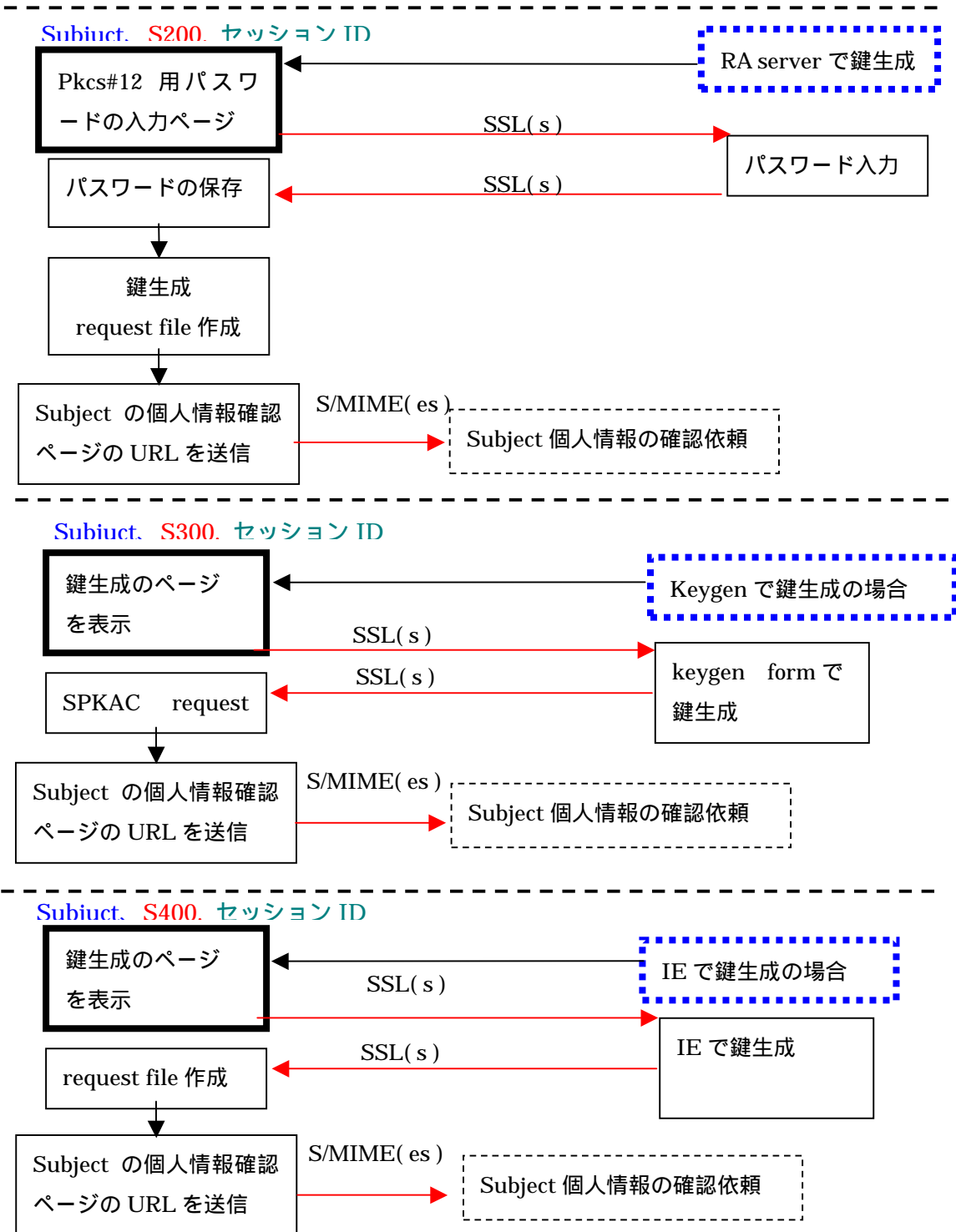
他にも、RA のポリシーに応じて、住所、電話番号、性別、生年月日などの個人情報を入れる。

秘密情報の入力を 1 度だけで後のセッションを連続的に行う。ただし、途中でセッションを横取りされないようにするためにこのシーンのためのサブセッション情報が必要。



### Scene3 鍵生成

このセッションは、シーン 2 から継続している。したがって、サブセッション ID などはシーン 2 のものと同じものを使用する。鍵生成の方式は、3 種類あるがいずれの方式でも、鍵生成が終わると RA に個人情報の確認を依頼する。



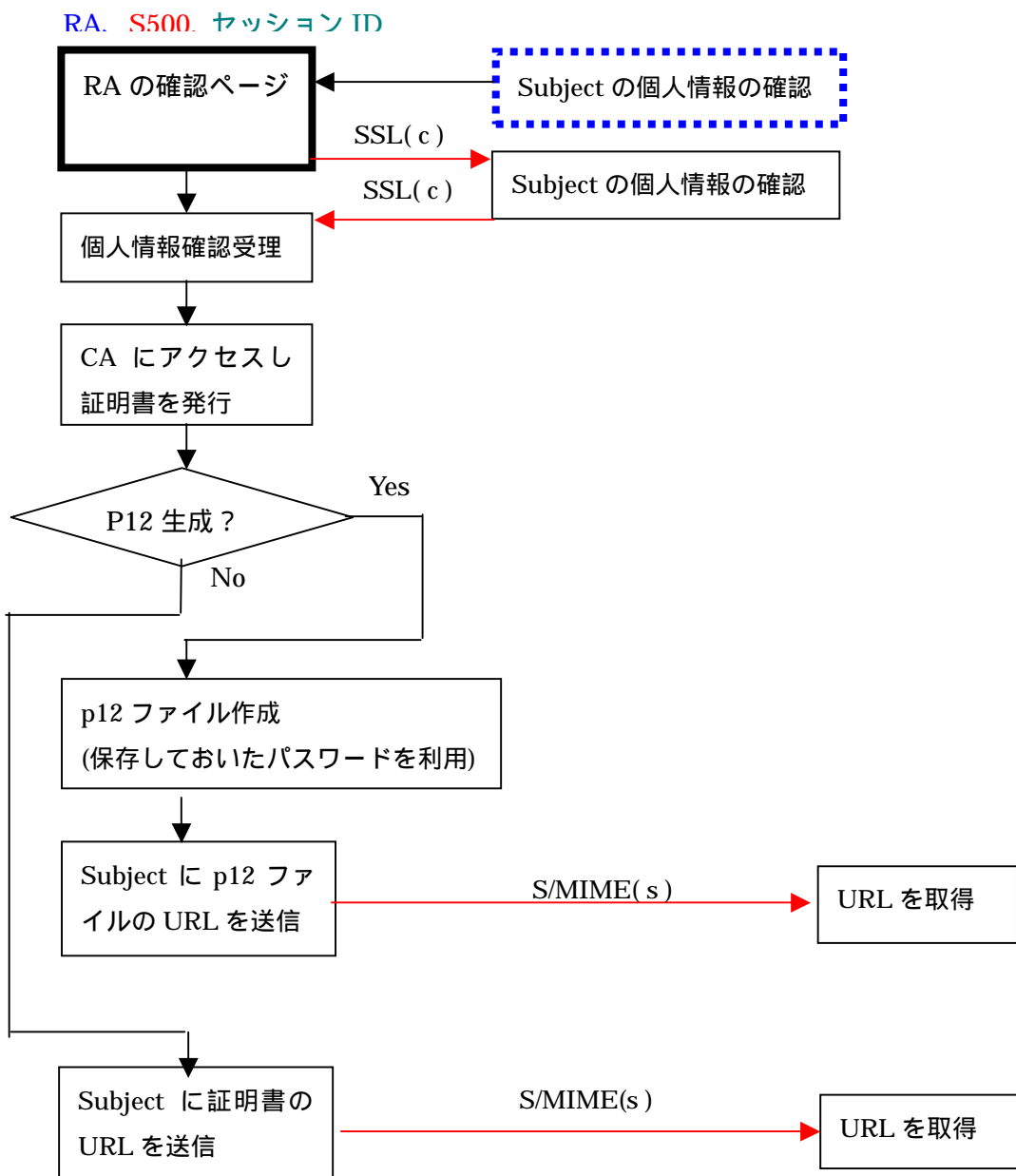
## Scene4 RA による Subject の個人情報確認と証明書発行

このセッションは、RA によるセッションである。

個人情報の確認が成功したときに、CA と RAserver が通信を行い、証明書を発行する。

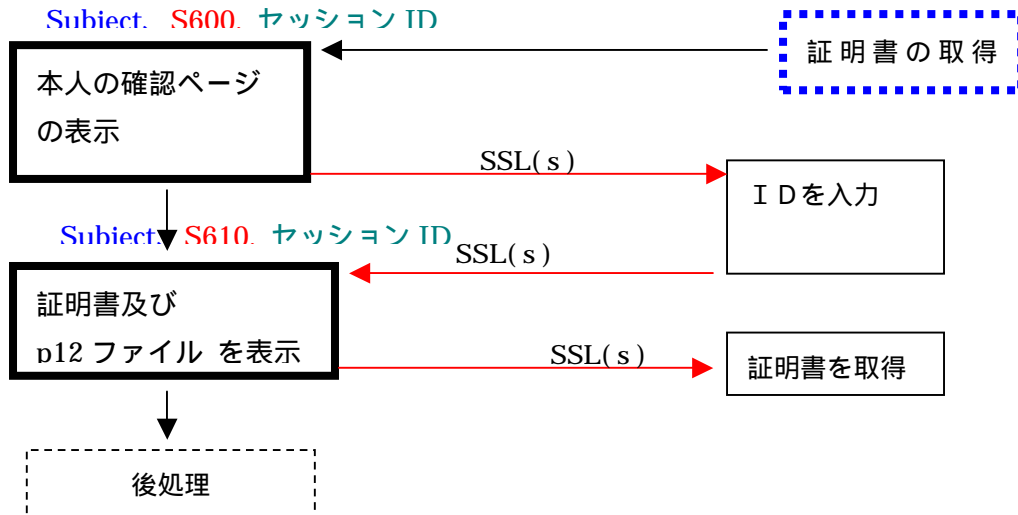
シーン 3 のコンテキストが RAserver での鍵生成の場合、ここで P12 ファイルを生成する。

それ以外の場合は、単に証明書を subject に送信するだけである。



## Scene5 証明書の取得

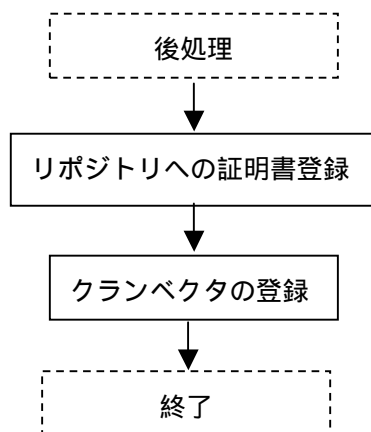
ここでは Subject が証明書を取得する。指定のURLへのアクセスにはIDが必要である。



## Scene6 後処理

このセッションは、シーン5からの継続である。

ここで、発行された証明書のリポジトリへの登録やクランベクタの登録処理などを行う。



**フリースクール(久留米)：2002年4月30日**

証明書のシリアル番号管理について

Raserver のデータベースでシリアル番号を管理する .

理由：RAA 分割，区分 CRL に対応するため，RAA 毎に index.txt が存在する .

シリアル番号の衝突を避けるため，シリアル番号の集約をするデータベースが必要 .

CRL 配布点について

LDAP で配布しようとする時，対応しないアプリケーションがある .

LDAP と HTTP 間の変換プロキシが必要では？

OpenSSL の設定ファイル

CA を構築するディレクトリのパスを，各設定ファイル(ca.conf，person.conf，server.conf)の同期をどうするか？

CA の秘密鍵の置き場所

RAA 毎のディレクトリに置くのはまずいかな？

どこかのディレクトリに固定して，フルパスで記述 .

ユーザー証明書の有効期間

1年と少しにする？オーバーラップの期間をもうける .

実際の期間は RAA のポリシーに依存する .

最長期限の縛りは必要でしょう .

CRL の発行頻度

RAA のポリシーに依存する .

最低頻度の縛りは必要でしょう

余談：OCSP レスポンドのサービスとか出来たらいいな .

使用するハッシュについて

やっぱり SHA1 の方がいいかなあ？

RAA ページについて

RAA の証明書を発行するページが必要では？

そのページにアクセスする権限の管理が必要では？

文字列の最大値は？

commonName\_max = 64

emailAddress\_max = 64

extendedKeyUsage

RAA のポリシーに依存する .

policyConstraints

CACAnet Class A CA では無しでいこう .

PASS 検証に関係する項目だけど , CACAnet は PASS 検証しない .

S/MIME 処理の不具合

From 行が認識されない . 重大な問題である . 改行コードの問題かな？

桑山さんが言っていた , バイナリモードを試してみる必要有り .

設定ファイルの整理

CA 用 , person 用 , server 用と整理しないとね .

2002 年 5 月 13 日

policy constraints 問題

Windows XP Home Edition での IE6 では , フィールドの存在は理解している .

ただし , 値がおかしいと判断している？

Windows2000 上では , フィールドの存在を理解できない .

フィールドを理解できるのは , Windows XP じゃないとダメなのでは？

XP ではフィールドの値がおかしいので , 「無効なポリシーがある」

ということで , CA の証明書を信頼しないという問題がある .

Policy Constraint を外して CA の証明書を作成 .

その CA からユーザー証明書を作成して実験をした .

新しい証明書での実験結果

・ WindowXP な方

CA とユーザーの証明書の組込に成功 .

IE6.0 と NS6.2 では問題なかった .

・ Windows2000 な方

IE5.5 と Winbiff 2.34PL1 + S/Goma2.12 では問題なかった。

Mozilla では、本文に日本語を使わないと送ることが可能。検証中。

文字コードを変えたら、うまくいった模様。

・ MAC OS X

mozilla 1.0RC2 日本語リソース入りでは、Winbiff 2.34PL1 + S/Goma2.12 から送った署名・暗号化メールは、「メッセージに署名が含まれていない」というふうにててしまう。

日本語化を行わなければ、PKCS#12 ファイルの読み込みが出来る。

ただし、CA の証明書を消してから入れなおさないといけない。

CRL 配布点について

区分 CRL を置く場所の問題があるのでは？

現状は以下のとおり。www.cacanet.org に決め打ちしてしまっている。

X509v3 CRL Distribution Points:

URI:<http://www.cacanet.org/CACAnetClassACA/CACAnetClassACA.crl>

Friendly Name

は、CA 証明書もユーザー証明書は入れたほうがいい？入れないほうがいい？

ユーザー側で入れるようにしておいたほうがいい？

**フリースクール(久留米)：2002年7月8日**

証明書の廃棄

「CACAnet の証明書廃棄システム説明会」の見直し

<http://cvs.cacanet.org/fsc/revoked/0827.html>

証明書廃棄ページのインターフェース

- ・ 廃棄する証明書の電子メールアドレスやシリアル番号を入力。
- ・ 対応する証明書の内容を表示して確認。  
(リポジトリから検索するようにする?)
- ・ その証明書を廃棄要請。

証明書廃棄申請の方法は、RA に任せる。

廃棄ページというよりは、リポジトリと連携を取った

証明書検索のページにするほうがいいだろう。

検索項目は、電子メールアドレス、シリアル番号、名前など。

RA 権限をもつ人がアクセスした場合は、

「廃棄」が出来るというイメージになるのではないか？

だれが廃棄可能にするか？（RA の CPS による？）  
他の機関の RA が廃棄できないようにする必要はある．  
証明書を発行した RA のみが廃棄可能にする？  
それとも，同じ機関の RA なら，誰でも廃棄可能にする？  
また，それを制御するインターフェースは必要か？

制御のインターフェースを作る．  
発行した RA 以外が廃棄可能か不可能かの二択のみをサポート．

#### 廃棄の履歴

どの RA が，どの Subject の証明書を廃棄したかの履歴を残す必要があるか？  
必要ならば，どのように残して，どのようなアクセス制御が必要か？

RA が発行した証明書と廃棄した証明書のシリアルリストとかが要るだろう．  
配列とかリストとかのデータ構造に入れることになる．Pstore が便利かな？

RA が不正を行った場合，  
その RA から発行された証明書をまとめて廃棄する仕組みが必要か？

バッチ処理する仕組みは必要だろう．発行や廃棄も含めて．  
でも，これは将来的な話ではないだろうか？

#### 廃棄する時間の制約

CRL を発行する時間に重ならないように制約を設けるか？

証明書の発行と廃棄が重なるほうが危険だろう．  
CRL についてはインデックスファイルの読み込みだけなので，  
特に気にしないでいいのではないかな？

#### CRL の発行

「CACAnet の証明書廃棄システム説明会」の見直し

<http://cvs.cacanet.org/fsc/revoke/0827.html>

CRL のバージョンは 1 でいくか？

CRL のバージョンが 2 なら，どんな拡張を入れるか？

・crlEntry extension は，OpenSSL の仕様上ダメ

- ・ cRLNumber はサポートしていない

現 C P S 7 章ではバージョン 1 になっているが、

バージョンは 1 と 2 の両方を発行する。

プロファイルは、<http://cvs.cacanet.org/fsc/revoked/0827.html> を参照。

\*\*\*\*\*

OpenSSL 0.9.7 b2 では、crlEntry extension をサポートしているらしい。

これは、将来的な構想になるだろう。

\*\*\*\*\*

CRL の中に、Revoke をした RA 名を残す必要があるか？

CRL の仕様の無理。

区分 CRL への対応

index.txt を分割するかしないか？

- ・ index.txt を分割する場合  
分割するのなら、Serial をどこで管理するか？  
RAServer 側か？ CA 側か？

分割する方向でいく。

証明書の発行や廃棄時に、処理が複雑になる。

シリアル番号の管理は、やっぱり CA でやるのが合理的だと思う。

- ・ index.txt を分割しない場合  
CRL 発行時に RA 名をキーに抜き出して、RA 毎にインデックスデータを作成。  
そのインデックス使って区分 CRL を発行する。

この場合は、証明書の発行や廃棄時は特に問題は生じないが、

CRL を発行する時に処理が複雑になる。

サーバー群 ( CA , RAServer , LDAP ) の時刻校正をどうするか？

NTP サーバーを動かすか？動かすならどこで動かすか？

dns.cacanet.org で、NTP サーバーを動かす。

あと、ズレの許容範囲をどうするか？

以下のリポジトリへの設置方法では、1分ずつずらして設置になっているので、それに見合う許容範囲にするべきであろう。

でも、将来的には GPS 装備が理想ですね。

CRL リポジトリへの設置方法

HTTPS や LDAPS を使って、定期的に設置するしかないか

午前3時～4時頃は、システムがデータベースを構築している時間帯なので避ける。  
午前5時頃に、CA -> RAserver -> LDAP と1分ずつずらして設置する。

**フリースクール(久留米)：2002年7月17日**

東様の質問に答える形式で行った。

#### 1. 証明書の発行

各資料の確認(基本仕様書)

- ・「個人証明書発行手続き」の資料は、以下の資料を示すのですか？

<http://cvs.cacanet.org/doc/index.html>

CACAnet の証明書発行システム 2001年6月15日

「個人証明書発行手続き」の資料は、次の証明書発行システム仕様書のことを示す。

[http://cvs.cacanet.org/spec/b\\_spec.pdf](http://cvs.cacanet.org/spec/b_spec.pdf)

- ・システムの「ディレクトリ構造」の資料？ 見当たらないと思います。

(システムは、RA・RAserver が該当しますか？)

システムの「ディレクトリ構造」の資料は、大体作成してあるので公開する。

システム正常系

#### 1.1 発行依頼申請(詳細設計)

(1) subject の登録項目は、どのような情報ですか？

- ・subject メールアドレス
- ・subject 氏名

( 個人情報の内容はどこまで必要ですか？ R A の C P S で確定した項目になりますか？ )

基本的には Subject のメールアドレスとローマ字氏名のみ .  
RAA の CPS によって増やすことは可能 .

( 2 ) 登録ページ・確認ページ・承諾ページ ( O K ・ N G ) の画面レイアウトはどのようになっていますか？

実際の画面を確認してもらおう .

( 3 ) 各画面の入力チェックはどのようになっていますか？

基本的に入力チェックは行わず確認画面によってチェックする  
ただし、Subject の入力情報として CACAnet としての必須入力情報の部分に関しては空白もしくは不当なアドレスチェックを行う。

- ・ローマ字氏名
- ・電子メールアドレス

( 4 ) 画面の入力順番はどうなっていますか？

「Subject 向け手引書」を参照 ( まだ Web 上には公開されていない部分 )

異常 ( 5 ) 入力ミス時のエラー内容及び処置はどうなりますか？

( エラー情報をメッセージファイルで管理しますか？ ) エラー仕様はどのようなものですか？

エラーの内容に対して逐一对応する

( 6 ) D B 構造はどのようになっていますか？ ( ファイルレイアウトはどのようになっていますか？ )

セッション T B L 構造等

(セッション I D とセッション状態コード (コードの意味は? コード仕様が必要))

DB 構造はどうなっているかについて討論中

DB 構造のセッション管理のデータベース部分としては、現在の scdb.rb の SCDB プログラム部分にある rd 形式で書かれたものが大枠であり、それを元に DB 構造の仕様書を作成。

RADB も同様。

( 7 ) 秘密情報の生成する仕様はありますか?

(関数になっていますので無いですね関数仕様はありますか)

10 桁の 10 進乱数を使用。

( 8 ) 秘密情報及びセッション I D を s u b j e c t に渡す情報はどのような項目ですか?

異常 ( 9 ) 秘密情報を生成中障害が発生した時、再生成することが可能ですか?

(どのような処置になるのですか?)

この場合は異常システムが発動しますので、エラーページが出力し「ふりだしにもどる」となっている。次のバージョンで厳密に対処。

異常 ( 1 0 ) 秘密情報とセッション I D の情報は食い違いが発生しないのですか?

(バージョン等)

秘密情報とセッション I D は同一のレコードで管理しているので系統的に食い違いはおきず、Subject 側で食い違いをおこしていたらエラーとなる。

( 1 1 ) 秘密情報を s u b j e c t へ渡す場合は、どのような受け渡しをするのですか?

(媒体であれば、ファイル名及び内容が必要と思います)  
ファイルのネーミングルールがありますか？

アウトオブバンドです。

## 1.2 証明書申請手続き(詳細設計)

(1) Subjectの秘密情報とセッションIDの入力画面の画面レイアウトはどのようなになっていますか？

秘密情報は、アウトオブバンドにてい得た情報を書き込む。  
セッションIDは、メールで通達が来るURLの中に情報が入っている。

(2) 入力項目は秘密情報(入力内容及び桁)とセッションIDのみですか？

秘密情報だけ。

(3) 入力値が不整合の時、どのような処置になりますか？

(subjectに対し、エラーを表示し再入力をしてもらうか、システム管理者へ依頼となりますか)

セッションIDが異なる場合は、異常処理によりセッションが存在しない旨を告げる。  
秘密情報が異なる場合は、異常処理により秘密情報が異なる旨を告げる。  
SCDBの仕様を変更して、5回失敗した場合は「ふりだしにもどる」とする

異常(4) 秘密情報及びセッションIDの有効期間はありますか？

(subjectが申請をしなかった場合)

有効期間はある。プログラムのにはstate.rbをまとめる。  
その期間が妥当かどうかは考えていない。  
現状はそのまま。

( 5 ) s u b j e c t の個人情報の入力ページでの個人情報はどのような項目になりますか？

画面レイアウトはどのようにになりますか？ ( R A の C P S 検討 )

運用しながらきめる . R A の C P S による .

( 6 ) 各画面の入力チェックはどのようにになりますか？

運用しながらきめる .

( 7 ) 画面の入力順番はどうなっていますか？

( 8 ) 入力ミス時のエラー内容及び処置はどうなりますか？

( 9 ) D B 構造はどのようになっていますか？ ( ファイルレイアウトはどのようになっていますか？ )

セッション D B 構造等

まとめて、先に話した内容と同じ .

( 1 0 ) 鍵生成方法の選択ページのレイアウトはどのようにになりますか？

PKCS12 か鍵ペアをブラウザで生成の 2 択

### 1 . 3 鍵生成 ( 詳細設計 )

R A s e r v e r での鍵生成

( 1 ) P k c s # 1 2 のパスワード入力ページレイアウトはどのようになっていますか？

レイアウトは見てもらう .

パスワード入力は , HTML フォームのパスワード入力にて任意の文字を入力 .

パスワードは 2 回入力して , そのチェックを JavaScript にて確認 .

長さの制限はしていない . 運用実験で調整 .

( 2 ) 入力チェックはどのようになりますか？

JavaScript を使って確認 .

( 3 ) パスワードの保存を保存する場所 ( フォルダ ) 及びレイアウトはどのようになりますか？

セッションが切れるのでパスワードを再利用しないといけないため , SCDB に生で保存 . 生でなくハッシュなどにするとしたら , 証明書発行手続きのながれを変えなければならないので , 次期バージョンへの検討事項とする .

( 4 ) このパスワードはいつまで有効ですか？

SCDB にパスワードを入れているが , システム的には 1 . 2 ( 4 ) で話したセッション保存時間がシステム的なパスワード保持期間になる .

( 5 ) 鍵の生成後、保存場所 ( フォルダ場所 ) と保存期間？

PKCS12 ファイル中の秘密鍵の扱いは , 運用しながら考える . .

秘密鍵ファイルを消すかどうかは運用中に考える .

秘密鍵のモードが不適切なので , モード変更する必要有り .

デフォルトの umask 設定と , パーミッションの変更をするという二重チェックを行う .

異常 ( 6 ) 鍵生成時、なんらかの障害で生成出来なかった場合は、どのような処置になりますか？

( 生成時、S / M I M E 時 )

現在は鍵生成時にエラーが出た場合の処理は現状ではできていない .

もし起きた場合は再チャレンジ可能にする .

( 7 ) 個人情報確認のページはどのような構成になりますか？(ディレクトリ構造を参照  
ですか？)

Subject の個人情報なども含む入力情報は，A の CPS による．  
確認ページは入力された情報をそのまま出力し，本人による確認．

( 8 ) 個人情報確認ページの管理はどのようにになりますか？

個人情報の管理確認という面では，RA の CPS に依存する．

( 9 ) 秘密情報とセッションIDをもっている人は、何回もパスワード更新が可能です  
か？(勉強不足)

P12 作成前では可能だが，P12 作成後は不可能．  
表向きは出来ないが、裏技的な方法を用いれば可能な状況になる．

keygenでの鍵生成(よく理解していませんすみません)

( 1 ) 鍵生成ページのレイアウトはどのようになっていますか？

レイアウトは本物を見てもらうとして，  
発行システムの画面ではなくてIEのレイアウトになる．

( 2 ) 鍵の生成情報は、何回もrequest可能ですか？

これも表向きは出来ないが、裏技的な方法を用いれば、可能な状況にもなる．

IEで鍵生成(他のブラウザの評価をどこまでの範囲としますか？)

1.3と同じ．

( 1 ) 鍵生成ページのレイアウトはどのようになっていますか？

本物を見てもらって確認してもらおう。

( 2 ) 鍵の生成情報は、何回も request 可能ですか？

これも表向きは出来ないが、裏技的な方法を用いれば、可能な状況にもなる。

1. 4 個人情報確認と証明書発行 ( 詳細設計 )

( 1 ) RA の確認ページは誰でも参照することが可能ですか？

RA しか参照できない。他の団体の RA も参照できない。

( 2 ) 個人情報確認のレイアウトはどのようになっていますか？

レイアウトは本物を見てもらう。

RA の CPS によった、Subject の入力した個人情報が出力され確認。

( 3 ) 個人情報確認受理の判断は、なんですか？

RA の CPS による。

( 4 ) P 1 2 での生成とその他での生成判断条件情報はどこで管理されていますか？

SCDB で管理。

( 5 ) 各個人用証明書はどのように管理されるのですか？

( 証明書の有効期間は、どのときから 1 年間有効になりますか )

リポジトリによって管理。管理はリポジトリ運用方針による。

リポジトリの運用方針をつくる必要有り .

#### 1 . 5 証明書の取得 ( 詳細設計 )

( 1 ) 本人確認ページのレイアウトはどのようになっていますか ?

証明書の取得のフェーズでは 本人確認はしない .  
本人確認をしなくても良いかと議論が進んだが ,  
PKCS12 の証明書取得画面は PKCS12 のパスワードレベルでの  
セキュリティレベルということで ,  
現状の証明書発行手続きのままですめる .

( 2 ) 入力チェックはどのようになりますか ?

( セッション ID 入力 )

( 3 ) 入力エラーは、どのようになりますか ?

これらは 1 . 5 ( 1 ) があつた場合は必要だが , このシステムが無いので不要 .

**フリースクール ( 久留米 ) : 2002 年 7 月 19 日**

証明書の入手

LDAP リポジトリの仕様

・使用する文字コード

ASCII のみにする .

RFC2459 と RFC3280 では , 2003 年 12 月 31 日以降は DN は UTF8 が MUST な  
ので ,

UTF8 への対応は考えておく必要がある .

・DN の仕様

以下のとおりとする . OU の階層は , これ以上増やさない .

countryName = Country name

countryName\_default = JP

0.organizationName = Organization Name  
 0.organizationName\_default = CACAnet Fukuoka  
  
 1.organizationName = RAA Name (eg. company)  
 1.organizationName\_default = CACAnet Fukuoka Members RAA  
  
 0.organizationalUnitName = Organizational Unit Name (eg. section)  
 0.organizationalUnitName\_default = member (または明示的に部署を指定)  
  
 1.organizationalUnitName = Entity Type  
 1.organizationalUnitName\_default = person (サーバー証明書なら server)  
  
 2.organizationalUnitName = RA serial  
 2.organizationalUnitName\_default = RA 証明書のシリアル番号  
  
 commonName = User Name  
 commonName\_max = 64  
  
 emailAddress = Email Address  
 emailAddress\_max = 64

- ・ 証明書を登録するスキーマ  
inetOrgPerson クラスの userCertificate 属性に , Binary データとして登録 .
- ・ CRL を登録するスキーマ  
cRLDistributionPoint の certificateRevocationList に , Binary データとして登録 ?  
CN を入れないといけないのだが , CN が何者かわからないので , 調べる必要あり .

```

objectclass ( 2.5.6.19 NAME 'cRLDistributionPoint' SUP top STRUCTURAL
    MUST ( cn )
    MAY ( certificateRevocationList $ authorityRevocationList $
        deltaRevocationList ) )
  
```

リポジトリからの削除  
 リポジトリには最新の証明書のみを上書きして登録し ,

過去の証明書は他の場所に保管しておく。

#### CA のエラーコード

CA マシンが返す HTTP Response の Body 部分に以下のデータを入れておき，  
RAserver がそのデータを見て判断するようにする。

##### ・肯定応答

+OK

-----BEGIN CERTIFICATE-----

.....

-----END CERTIFICATE-----

##### ・否定応答

-ERR Error\_Code

##### ・Error\_Code について

- 1 : CA の秘密鍵のパスワードファイルオープンエラー
- 2 : 新しいシリアル番号の生成エラー
- 3 : Request データのエラー  
空データ，フォーマットエラーなど
- 4 : OpenSSL 実行時のエラー  
起動エラー  
実行時の何らかのエラー  
署名失敗  
インデックスファイルのアップデートミスなど

**久留米で議論2：002年7月22日**

山村，大岡，永井，遠藤，田代

現在の発行システムの仕様では，

OU（部署名）については RA の証明書のを引き継ぐようになっています。

しかし，RAA から RA になる人に証明書を発行する場合，RAA と RA は部署名が異なっ  
てきます。

よって，RAA から RA の証明書を発行する場合を考えなくてはならないと思います。

昨日話した内容では、RAA がアクセスしてきた時には、S000 のところで、部署名を入力するフォームを出すようにすればいいのではないかとのことです。デフォルトでは、RAA の部署名を入れておくほうがいいですかね？ 部署データは、セッションデータベースに入れておきます。

また、RA の証明書をリポジトリに登録する時は、OU (部署名) のディレクトリエントリが作成されていないので、OU のディレクトリを作ってから RA の証明書を登録するということになると思います。

それから、秘密情報を入れるところで、秘密情報を間違った時の処理です。5 回失敗したら振り出しに戻るようにしたほうがいいという意見になっていますが、これを実現するには、セッション管理データベースで扱う項目を増やすという方向に行きたいと思います。

#### **久留米で議論：2002年7月26日**

メンバー：田代，山村，永井，遠藤

##### リポジトリから証明書の検索

- ・ 証明書のシリアル番号での検索は出来ない。  
(それでいいのかな?)
- ・ 検索ページのインターフェースは、  
検索の項目 (CN や emailAddress, OU など) を入力するフォームを並べる。  
(あいまい検索のようなものは必要か?)
- ・ DN で使うローマ字はヘボン式とする。

##### 証明書の廃棄

- ・ 失効理由の入力  
失効理由については、RFC2459 の"5.3.1 Reason Code"で規定されているものを使う。
- ・ どの RA が廃棄したかについての管理

##### 証明書廃棄リスト

- ・ RFC2459 では cRLNumber は必須だが、OpenSSL0.9.6d では未対応。  
今回の仕様では cRLNumber は無しにする。

OpenSSL の機能制限を考えると、証明書廃棄管理データベースが必要な気がします。

- ・ 廃棄された証明書のシリアル番号
- ・ 廃棄された理由(Reason Code)
- ・ 廃棄した RA の情報 (シリアル番号など)

が入っているようなものです。

**久留米で議論：2002年7月29日**

山村，大岡，永井，遠藤，田代

#### リポジトリの構造

\*\*\*\*\* LDIF 形式データ \*\*\*\*\*

dn: o=CACAnet Fukuoka,c=JP

objectClass: top

objectClass: organization

o: CACAnet Fukuoka

dn: o=CACAnet Members RAA,o=CACAnet Fukuoka,c=JP

objectClass: top

objectClass: organization

o: CACAnet Members RAA

dn: ou=CACAnet Class A Members RA,o=CACAnet Members RAA,o=CACAnet Fukuoka,c=JP

objectClass: top

objectClass: organizationalUnit

ou: CACAnet Class A Members RA

dn: ou=0BC7D08ECC4CE340676D10071BE4C80B,ou=CACAnet Class A Members RA,o=CACAnet Members RAA,o=CACAnet Fukuoka,c=JP

objectClass: top

objectClass: organizationalUnit

ou: 0BC7D08ECC4CE340676D10071BE4C80B

dn: ou=person,ou=0BC7D08ECC4CE340676D10071BE4C80B,ou=CACAnet Class A Members RA,o=CACAnet Members RAA,o=CACAnet Fukuoka,c=JP

objectClass: top

objectClass: organizationalUnit

ou: person

dn: CN=Shoji  
endo,OU=person,OU=0BC7D08ECC4CE340676D10071BE4C80B,OU=CACAnet Class  
A Members RA,O=CACAnet Members RAA,O=CACAnet Fukuoka,C=JP  
objectClass: top  
objectClass: inetOrgPerson  
sn: Shoji  
cn: Shoji endo

dn: Email=shoji@sylabo.co.jp,CN=Shoji  
endo,OU=person,OU=0BC7D08ECC4CE340676D10071BE4C80B,OU=CACAnet Class  
A Members RA,O=CACAnet Members RAA,O=CACAnet Fukuoka,C=JP  
objectClass: top  
objectClass: inetOrgPerson  
sn: Shoji  
cn: Shoji endo  
mail: shoji@sylabo.co.jp  
userCertificate::  
MIIG3TCCBcWgAwIBAgIQD46bUYfy19lWtywnCdpMHDANBgkqhkiG9w0BAQUFAD  
BXMQswCQYDVQQG  
.....

リポジトリを登録する場所の DN は , 証明書を解析して作成する .

DN の電子メールは "Email" を使う . "emailAddress" は , OpenSSL を 0.9.7 にバージョン  
アップした時に考える .