京都大学での
クライアント証明書の利用サービスと
学内申請受付システムの紹介

京都大学 情報環境機構
古村 隆明

UPKIクライアント証明書
利用予定サービス

- 無線LAN接続
  - 802.1x EAP-TLS

- VPN接続
  - PPTP

  - SSTP
  - openVPN

- メール暗号化・電子署名 (S/MIME)

UPKIクライアント証明書利用時の注意

- 他組織のクライアント証明書も全て同じCA局から発行される
  →ルートCA・証明書チェインの確認だけでは他組織の利用者も認証OKとなってしまう

- クライアント証明書の subject を読み組織(O)や組織単位(OU)で判定を行う必要あり

PPTPサーバの設定

- mpd/FreeBSD + FreeRADIUSで構築

- mpd.conf

```
set radius server 127.0.0.1 secret 1812 1813
set auth enable radius-auth
set auth enable radius-acct
set radius enable message-authentic
```

PPTPサーバの設定

- eap.conf (FreeRADIUS)

```
tls {
  :
  verify {
    tmpdir = /etc/raddb/tmp
    client = "${confdir}/check_subject.sh %{TLS-Client-Cert-Filename}"
  }
}
```

- check_subject.sh

```
#!/bin/sh
openssl x509 -subject -noout -in ${1} |¥
 grep '^subject= /C=JP/L=Academe/O=Kyoto University/' |¥
 grep -q '/OU=Kyoto University Integrated Information Network System/' && exit 0
exit 1
```

  - openssl x509 コマンドの -subject オプションを利用して想定した属性値が subject に格納されているか確認

SSTPサーバの設定

- subject の値で判定を行うために
  radius 認証機能を利用して
  FreeRADIUS で判定する
- FreeRADIUS の設定は PPTP の場合と同様

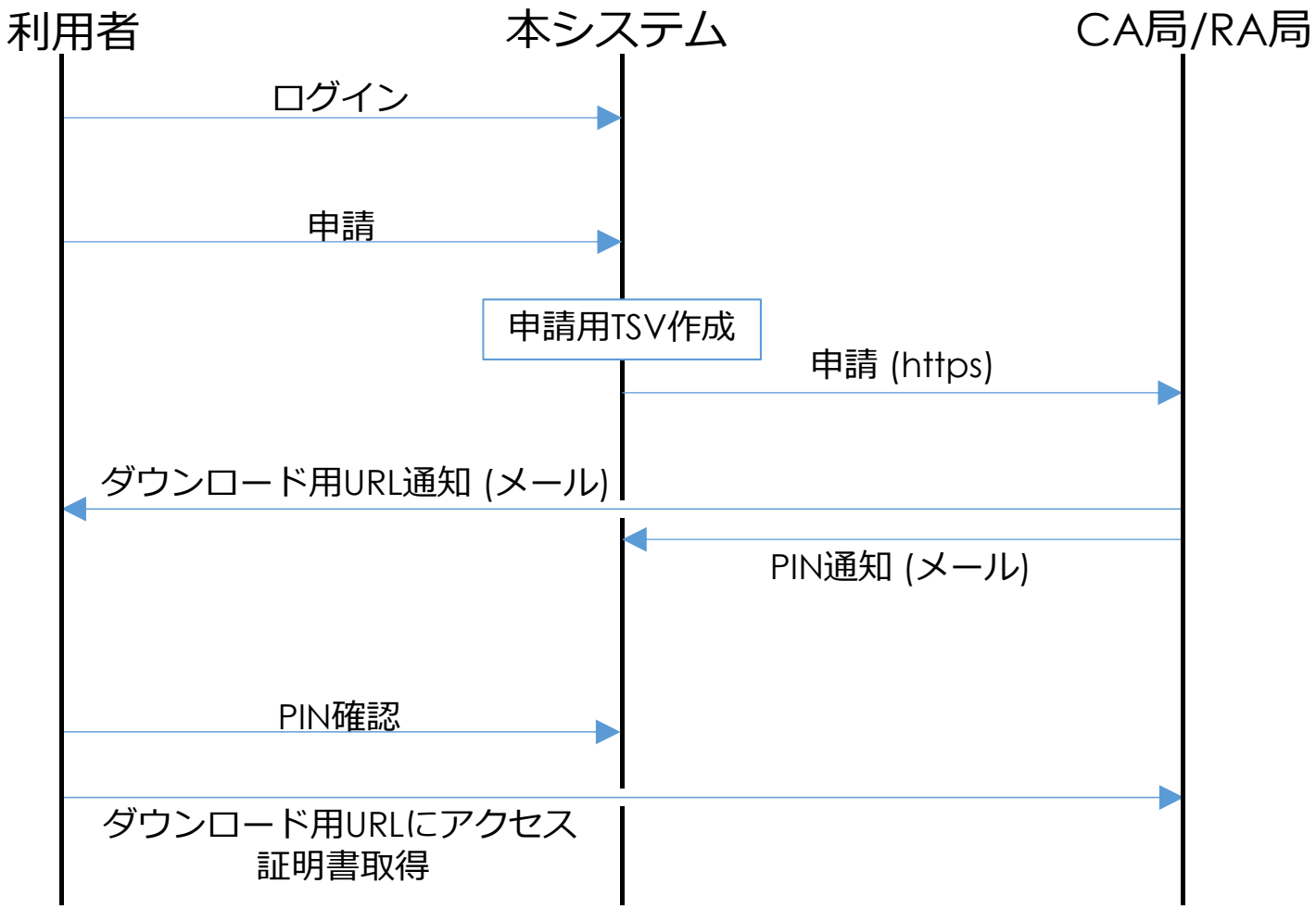
学内申請受付システム

- 利用者からクライアント証明書の申請を受付け NIIへ申請を行う

学内申請受付システム

- 証明書に記載する属性をShibbolethで取得
  - クライアント認証用証明書に uid
  - S/MIME用証明書に email

学内申請受付システムの処理の流れ



学内申請受付システム

- Ruby on Rails
- shibboleth認証

- オープンソース
  - github で公開中
    `https://github.com/y0s/shibcert-prototype`

- 有効期限10年のプライベート証明書
  - 2010年度から導入したIC職員証に導入

- S/MIME証明書を複数のデバイスで共有
  - 同一メールアドレスで複数のS/MIME証明を利用するとややこしくなる
  - 署名だけなら問題無い
  - 暗号化メールを送ってもらうときに問題が起きる

- クライアント認証用の証明書は必ずしもパブリック証明書である必要はない
- パブリックであることよりも(プライベートで構わないので)有効期限を延ばしたい