FriBID
and
Browser Security Software

FOSDEM 2011
Samuel Lidén Borell

eID in Sweden

- Common uses:
  - Signing electronic forms from the government.
  - Authentication on Bank sites.
- **Not** physical ID
  - Certificate with Personal Number + Name
  - Used to create **signatures** on the **web**
    (Compliant with 1999/93/EC)
- Issued by private companies

eID in Sweden

- Several systems in use:
  - Nordea eID (will merge with BankID in 2012)
  - Telia/SEB eID
  - BankID
- All **proprietary** and **incompatible**
  - And might **not work** with your favourite browser/architecture/OS version, etc.
- Lots of users: > **3 million** (there are 9 million citizens)

eID in Sweden

- Several systems in use:
  - Nordea eID (will merge with BankID in 2012)
  - Telia/SEB eID
  - **BankID**
- All ~~**proprietary**~~ and **incompatible**
  - And might ~~**not work**~~ with your favourite browser/architecture/OS version, etc.
- Lots of us

ere are 9 million citizens)

FriBID works with this one...

...and solves these problems

eID in Europe

Bank-id
(Norway)

BankID
(Sweden)

Telia/SEB ID
(Sweden)

Nordea ID
(Sweden)

NemID
(Denmark)

Identity card
(Belgium)

No standard :(

Often proprietary software :(

FineEID
(Finland)

EstEID
(Estonia)

EIC
(Italy)

... and
many
more

BankID

- Not a physical ID

- Smart cards and soft tokens
  - Unlike many (most?) other eID systems
  - Enrolment can be done at home:
    - Log in to bank, request certificate, done.
    - Valid for 1 year.

- Can store private key on SIM on a cell phone
  - Unsupported by most RPs so far.

FriBID – What is it?

- F/OSS client software for BankID
  - Reverse engineered
- One year since public release
  - Still in alpha
- Features:
  - PKCS#12 (soft tokens)
  - Smart cards through PKCS#11 (e.g. OpenSC)
  - Enrolment is being developed...

FriBID – Technical details

- Written in C
- GTK/Glade for GUI

- OpenSSL
- libp11 + OpenSC (or any other PKCS#11 provider)

- NPAPI/NPRuntime (plugin API)

BankID – Signature Protocol

1) Make Javascript calls to plugin
2) User is asked to enter PW or PIN
3) Plugin returns `xmldsig` signature

Example: Signature generation

```
<object id="signer" type="application/x-personal-signer2"></object>

                              ...

signer.SetParam(”Nonce”, ”MTIzNDU2Nzg5”);
signer.SetParam(”TextToBeSigned”, ”aGkK”);
sig = signer.PerformAction(”Sign”);
```

123456789

hi

Example: Signature generation

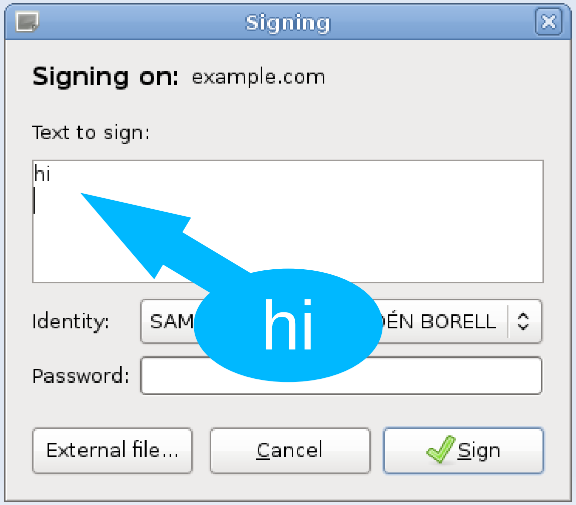
```
<object id="signer" type="application/x-
personal-signer2"></object>

                    ...

signer.SetParam(”Nonce”, ”MTIzNDU2Nzg5”);
signer.SetParam(”TextToBeSigned”, ”aGkK”);
sig = signer.PerformAction(”Sign”);
```

123456789

hi



Signing

Signing on: example.com

Text to sign:

hi

Identity: SAM ÉN BORELL

Password:

Example: Signature generation

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
...
<DigestValue>JN6EZ...41Broo=</DigestValue>
...
<SignatureValue>J9tq6Yc...HlA=</SignatureValue>
...
<X509Certificate>MIID5j...zz4fw==</X509Certificate>
<X509Certificate>MIID3j...Ra3JA==</X509Certificate>
...
<bankIdSignedData
xmlns="http://www.bankid.com/signature/v1.0.0/types"
Id="bidSignedData">
<usrVisibleData charset="UTF-8"
visible="wysiwys">aGkK</usrVisibleData>
<srvInfo><nonce>MTIzNDU2Nzg5</nonce></srvInfo>
<clientInfo><funcId>Signing</funcId>
<host><fqdn>example.com</fqdn><ip>198.51.100.200</ip></host>
<version>UGV...ODAm</version></clientInfo>
</bankIdSignedData>
...
```

hi

BankID – Enrolment Protocol

1) Send person name, etc, to plugin
2) Plugin generates key pair
3) Plugin returns CSR
4) Send certitifcate chain to plugin
5) Done

BankID – Enrolment Protocol

1) Send person name, etc, to plugin

DN

2) Plugin generates key pair

3) Plugin returns CSR

PKCS#10 wrapped
in a **PKCS#7** container
(+proprietary extension)

4) Send certitifcate chain to plugin

X.509s wrapped
in a **PKCS#7** container

5) Done

Difficulties

- Secret protocol – Does it follow the standards?
  - xmldsig, ASN1, PKCS#7
  - Better emit the same output as proprietary s/w
  - Debugging with legally binding signatures...
- Choice of security library?
- Protocol with blocking Javascript calls
  - Does not work well with NPAPI
  - Plugin designers: Don't do this!

Browser security software

Why? What? How?

Signing in the browser

Not just eID:

- Alternative to passwords
- Alternative to session ids
- etc.

Why?
Cookie stealing,
Stolen databases,
Dictionary attacks,
Sniffing,
etc.

Yes, TLS does this also.

Different users:

- Banks, government (need high security)
- ”Others” (need good privacy)

Signing from the browser

- When is TLS not enough?
  - Signatures (not just auth)
    - Can be verified by 3rd parties
  - WYSIWYS (users see what they sign)
- Timestamping (for long-lasting signatures)
  - Digital signatures from Trusted Third Party
  - Linked Timestamping

Key pairs in the browser

- Enrolment
  - Client certificates for TLS can be enrolled, not standardised. E.g. `<keygen>`
- How to protect privacy?
  - Don't always want to reveal your identity
  - Don't want to have an ”unique identifier”
  - Can use different soft tokens for less important sites?
  - Better solutions?

Build on existing standards

- PKCS#11
  - Can use SmartCards through OpenSC PKCS#11 library
  - Can use soft tokens too
- Signature format
  - xmldsig
  - X-ADES (extension to xmldsig)

Existing software / standards

- General-purpose in-browser signing:
  - Open Signature
  - WASP
- Auth-only: SSL, gpgAuth
- Also, there are open source eIDs:
  - EstEID
- Probably many more...

Links and slides

- http://tinyurl.com/fribid-fosdem-2011

FRI BID

Appendix: Backup/deleted slides

Browser Security Protocols

	Open spec.	Signatures	See what you sign (WYSIWYS)	OSS implementation
BankID	No	Yes	text/plain, Attachments?	Partial
EstEID	?	Yes	Separate from plugin	Yes
Open Signat. WebFirma	Yes	Yes	?	Yes
SSL	Yes	No	No	Yes
Wasp	Yes	Yes	HTML	In progress

How to extend the browser?

	Security Software – Browser interface	Cross-platform	Secure WYSIWYS	Used by
Local HTTPd	Standalone	Yes	If Javascript is enabled	Open Signat., FINEID
MIME type	Plugin or Extension	No / Yes	Yes	Wasp
HTML tag	Extension or Builtin	Yes / Yes	In separate window	Firefox <keygen>
Native Javascript	Extension or Builtin	Yes / Yes	In separate window	
<object> Javascript	Plugin	No	In separate window	BankID, EstEID
Java Applet	Applet	Yes	No	Norweigan eID