Identity in the Browser
2011
Michael Hanson, Dan Mills, Ben Adida
A quick history - current and proposed browser identity features

- Password store & sync
- “Account Manager” authentication and session metaprotocol
- Contacts API prototype
- OpenID sniffer / ID presentation widget
2010: Account Manager

- Metaprotocol for site advertisement of authentication capabilities and session state
- Profiles for HTTP Basic, HTTP Form
- Very difficult to handle federated cases: huge number of error paths, no clear user model, hard to get agreement across browsers
subgoal:

Managing Session State

• DOM-level announcement of session(s) with identifiers, termination URL or JS callback, optional cookie “trigger”

• e.g. navigator.id.sessions =
  [ { id: “username@email.com”, end: “http://site.com/logout” } ]
Step back:
Minimal distributed identity system

• Distributed means hostnames. Identifier at hostname?

• Current RP systems are all based on email addresses: user-memorable, convenient, recognizable. (but: spammable, correlatable)

• RPs treat control of email address as stronger than username/password.

• Directed pseudonymity (anonymous remail) is a well-understood property of email.

• Discovery of attributes is well understood - stable identifiers help.
a proposal: Verified Email Assertions

- navigator.id.getVerifiedEmail(<callback>, challenge)
- window.onVerifiedEmail(function(assertion) {...})

native gives a nice UI and stronger security but we’ve got a pure JS, streamed-in API working
Identity provider’s half:

- `navigator.id.registerVerifiedEmail(id, <pubkey-callback>)`
- `navigator.id.certifyVerifiedEmail(id, <cert>)`
- public key advertisement/discovery

much to discuss:
- automated pseudonym provisioning
- limitations on register: session-only, non-persist, encrypted only
- limitations of key and certificate: long-lived key, short-lived cert?
- automatic cert refresh - but with what credential?
assertion (id, audience, valid-until, challenge, cert); signed with private

requestVerifiedEmail(<callback>)

[ user@domain: { private key, certificate, refresh/storage params} + ]
user confirmation flow (SMTP?)

trust determination & public key discovery

relying site

assertion (id, audience, valid-until, challenge, cert, issuer): signed with private

requestVerifiedEmail(<callback>)

secondary authenticating site

register (email, <callback>)

public key (certified, and returned)

[ user@domain: { private key, certificate, refresh/storage params} + ]
- Machines are multiuser
- Users are multipersona
- Core questions:
  - From site to user, “who are you”, and “how do I talk to you?”
  - From user to site, “I am <facet of me>”, and “You may know this about me”