The dreaded ‘e’ word and how to minimize the pain

Max Pritikin
May 31, 2006
Do we have to address enrollment?

- Device ‘out of the box’ (SIMPLE) should be able to leverage its IDevID to obtain an LDevID
- How many LDevIDs? In this discussion I assume only one.
NOT a requirement

• Long term LDevID management (re-enroll etc).
• It would be interesting to provision an appropriate URL such as:
  Enrollmentprotocol://theLDevIDserver
  Probably out of scope
Imprint?

• This must be possible without manual configuration of the device, nor an "out of band" exchange of credentials.
• This is likely controversial.
  – Pro: devices that ship in imprint mode anyway (most of them) end up with drastically simpler deployment scenarios
  – Con: what if an attacker imprints the device? Well, then it doesn’t have a valid LDevID anyway so no real harm
• Support pre-configuration of a particular LDevID server.
• Reverting the device to factory defaults to get back to IDevID causes it to lose all other configuration information.
  – Is this too draconian? What means is that if a device has a wrongLDevID and is thus reset to use IDevID any security risks by the wrongLDevID server will be cleared out in the process.
  – Does this offer any “proof” of security anyway?
Minimal hard crypto operations

- RSA or otherwise. (Even for ECC.)
- We don’t want to force low end devices to perform complex enrollment operations including complex messages.
- Theoretical minimum(?):
  1. Proof of possession of IDevID key
  2. LDevID key generation
  3. Proof of possession of the LDevID key
  4. Key verification of the DevID server key
Strawman Diagram

(not a proposal; just solidifying the discussion)

Device (Authenticator)  AS/RA  CA

1. EAP-TLS w/ client auth; device does IMPRINT so not exactly EAP-TLS

   {launch inner EAP method}

   2. Gen keys
   3. Gen enroll request (PKCS10)

{forward PKCS10 via normal enrollment mechanisms}

LDevID & CA URL Delivered

(4) Device verifies that LDevID is signed by the same EAP-TLS root