Deployable Security for Small Sensors

draft-arkko-core-security-arch-00.txt

Jari Arkko & Ari Keränen

Ericsson Research

Challenges in Securing Smart Objects

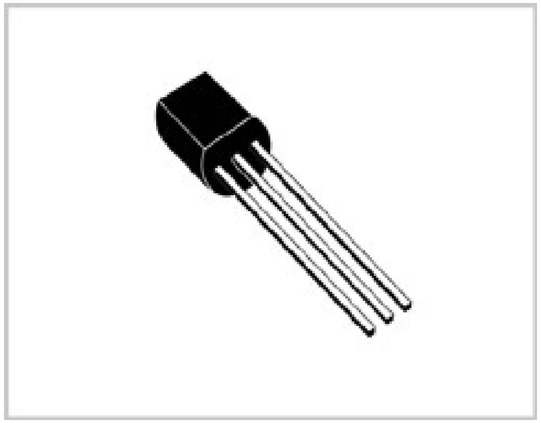
1. Implementation constraints
2. Provisioning difficulties
3. Layering and communication model issues

Implementation Constraints

- Computational effort & implementation complexity difficulties
- Message size growth issues
- Should not be overemphasized, if you need cryptographic security you'll have to add it
- Still, do it the right way, just once, etc.

Provisioning Difficulties

- Perhaps the most fundamental issue
- No keyboard, no display
- Maybe not even a button
- Untrained users
- 10s, 100s, 1000s of devices



How do you configure shared secrets or certificates on these?

Layering Issues

- Link layer security does not protect communications to peers multiple hops away
- Caching nodes, proxies and gateways terminate IP-level security connections
- Any sleeping node intermediation, storage, or filtering action also terminates these connections

The Secure Identity Architecture

- Provisioning approach
- The concept of secure identities
- Layer choice
- Initial protocol formats (alternatively, use WOES)

Secure identities:

ID = h(P)

”urn:dev:cgi:B7098D39781AABC6FF17”

Similar to what HIP, PGP fingerprints, or CGAs do (IPR warning)

The Provisioning Approach

- Read the identity off the sensors you install
- Few last digits, write down, bar code reader, …
- Feed the list of sensors to a server
- Often done anyway, while recording locations
- **Nothing** to configure in the sensors themselves

- Could even do this for a kit of sensors:

IDgrp = h(Psensor1 | Psensor2 | … | Psensorn)

Using the Identities

- Identities are not secret
- But receiver can use them to see if the message came from the correct source:

  Message = <Data, Psender, Signature>
- Others can't sign a message for that identity

```
{ "jmsg": { "temp": 27.5 },
      "jid":  { "id": "device:cgi-27611bc81020716627ff0000cfaa1234",
                    "ipb":  "4e26b808cd05d4e26b912ae3e43fe4eb45f82" },
      "jts":  { "s": 1311176727, "f": 123987 },
      "jsq":  23,
      "jsig": "18929abqxc67juil7ff2310000912927755bRRwlkadbfddceab"}
```

Conclusions

- Can't really talk about security without understanding the provisioning model
- Our architecture provides a practical, minimal-configuration approach to smart object security
  - Matches the existing provisioning practices
  - Matches the suitable communications models
- Trade-offs: requires PK crypto and in information-centric communication model replay protection is harder than in interactive security protocols
- For exact formats, actuator networks, detailed security considerations... read the draft

ERICSSON