Outline

• On-board Credentials (ObCs): What and Why
• ObC Architecture
• Instantiations of the Architecture
• Secure Provisioning of ObCs
• ObCs in Action
• Next Steps

This is a talk about a research project. Opinions stated here do not necessarily imply Nokia’s official strategy
On-board Credentials: What and Why

Authentication on the Internet

Username/password rules the Internet
- Cheap, easy-to-deploy, portable
- Annoying, vulnerable (phishing, dictionary attacks, password-stealing trojans...)

Attempts to improve usability and security
- Password-managers
- Single Sign-On
- Better protocols
Hardware tokens

Deployed for specific-services

- More secure, sometimes more intuitive
- More expensive, usually no trusted path to user,
- Single-purpose or issuer-controlled

Authentication is a security services that uses credentials
Credential: “credential secret” + a “credential program” to operate on it

Trusted hardware is becoming available

Proprietary secure environments for mobile device platforms
- Introduced to meet regulatory/business requirements
- secure boot, secure storage, secure execution of small programs
- E.g., Ti M-Shield (on several Nokia models), ARM TrustZone.

Trusted Platform Module (TPM)
- Specified by Trusted Computing Group (TCG)
- authenticated boot, secure storage and few operations (e.g. signing, encryption)
- cheap (0.10 €), deployed in many laptops (also servers/desktops)
On-board Credentials (ObCs)

An credential platform which leverages on-board secure environments

Secure yet inexpensive

On-board Credentials

Similar in spirit to multi-application smartcards

- But without any issuer control

Design should take limitations of existing trusted hardware into account
Design goals

• Credential programs can be executed securely
  • Use the secure execution environment
• Credential secrets can be stored securely
  • Use a device-specific secret in secure environment for secure storage
• Creating new types of credentials (i.e., new credential programs) is easy
• Anyone can create and use new credential types
  • Need a security model to strongly isolate credential programs from one another
  • Avoid the need for certification of credential programs
• Anyone can provision credential secrets securely to a credential program
  • Need a mechanism to create a secure channel to the credential program
  • (certified) device keypair; unique identification for credential programs
ObC architecture

(Minimal) secure environment based on trusted hardware
- Secure storage, using a device-specific ObC Platform Key (OPK)
- Secure execution
- Certified device keypair \( PK_{\text{dev}}/SK_{\text{dev}} \)

**Interpreter** runs in secure environment
- Credential programs (ObC programs) are scripts
- program hash as unique ID

---

Isolation of ObC Programs

Isolating the platform from programs
- Constraining the program counter, duration of execution, ...

Isolating programs from one another
- Only one ObC program can execute at a time
- An ObC program can “seal” data for itself
  - Sealing key is different for every independent ObC program
    Sealing-key = KDF (OPK, program-hash)
  - A program can invoke functions like “seal(data)” and “unseal(encrypted-data)”

---

M-Shield™: Example hardware secure environment #1

M-Shield provides

• Secure boot
• Chip-specific secret key (e-fuse)
• Secure execution of certified “Protected Applications” (PAs)
• On-chip RAM for PAs
• … (hardware RNG, crypto accelerators, …)

ObC using M-Shield secure hardware

- M-Shield secure boot used for validation of OS (Symbian OS)
- Interpreter (Lua), Provisioning subsystem are PAs
  - Use on-chip RAM
- OPK from chip-specific secret
- Device key pair
  - generated by Prov. PA
  - protected by chip-specific secret key
  - certified by manufacturer

TPM: Example hardware secure environment #2

TPM provides
- Authenticated boot
  - Components during boot measured and recorded in Registers (PCR) within TPM
  - A set of PCR values = a “configuration”
- Secure storage for keys bound to a specific configuration
- Ability to seal arbitrary data bound to a specific configuration
- Secure execution of selected cryptographic operations
- ... (remote attestation, ...
**ObC using Linux/TPM (2006)**

- Interpreter in kernel module on InitRD
- KeyInitializer in InitRD creates OPK on first use and seals for current configuration
- KeyInitializer unseals OPK on subsequent invocations.
- Security of execution can be improved using dynamic root of trust

MSc thesis work:

---

**Secure Provisioning of ObCs**
Requirements for Provisioning Credential Secrets

• Provisioning protocols typically focus on **user authentication** only
  • CT-KIP, Open Mobile Alliance Device Management (OMA DM), ...
• IETF keyprov working group is defining Dynamic Symmetric Key Provisioning Protocol (DSKPP)
  • Allows **device authentication** as well
• We need more...
  • provision a key so that it can be accessed by **specific credential programs**
• Subject to...
  • “Anyone can provision credential secrets securely to a credential program”
  • Support for multiple versions of credential programs
  • Support for several co-operating credential programs

Provisioning credential secrets (1/3)

Basic Idea: the notion of a **family** of credential secrets and credential programs endorsed to use them
Provisioning credential secrets (2/3)

- Provision a family **root key** to the device
  - using **authentic device public key** $PK_{\text{dev}}$

- Transfer encrypted credential secrets
  - using family **confidentiality key** $CK$

- Endorse credential programs for family membership
  - (program ID is encrypted)
  - using family **integrity key** $IK$

---

Provisioning credential secrets (3/3)

- 2-pass provisioning
  - Get (certified) device public key and validate it
  - Send ObCP/Init, ObCP/Endorse and ObCP/Xfer
  - CT-KIP 2-pass extensions defined by Magnus Nyström (RSA)

- Anyone can define a family by provisioning a root key
- Multiple credential secrets and programs can be added to a family
- Credential Programs can be encrypted as well

ObCs in action

An Example ObC: SecurID one-time password authentication

Joint research project with RSA security
1. ObC SecurID token: Full use of secure HW

- SecurID seed and algorithm protected by hardware secure environment
  - Provisioned data protected by encrypting with PK_{Dev}
  - Not accessible to Device OS
  - Hardware attack typically destructive and device-specific
- Encrypted seed stored in Credentials Database
  - Can be backed up
- Works on “named” devices of certain models (N95 and N96)

2. ObC SecurID token: Partial use of secure HW

- Credential Manager uses secure HW for encrypting OPK and SK_{DEV}
  - Symbian OS platform security provides run-time protection of seed & algorithm
- Encrypted seed stored in Credentials database
  - Can be backed up, with HW-encrypted OPK and SK_{DEV}
- Works on off-the-shelf devices of several S60 models
3. ObC SecurID token: Emulated

- SecurID seed and algorithm protected by Symbian platform security
  - Secure HW used for secure boot
- OPK, $SK_{DEV}$ cannot be backed up
- Works on any S60 3.1+ device

ObC implementation supports all 3 variants

- Implementation contains code for emulating secure environment (interpreter+provisioning+crypto) in S60
- Same software package can be installed in any S60 device
  - automatically decides the variant to use
Next Steps

Current/Future work

Refinements to ObC Core
- Piece-wise execution: longer programs, extensible crypto library (paper at STC ’08)
- Asymmetric credentials: enrollment augmented with attestation evidence

New target applications
- VPN access, web banking, browser integration, support for standard APIs...

New types of secure environments
- Provisioning for other “ObC” platforms (e.g., Javacard)

Credential migration
- Atomicity, mental models for usage

Better Usability
- User-friendly local authentication, Usable policy management

More formal Security analysis and validation
Summary

On-board Credentials aim to be
- An open credential platform based on general-purpose secure hardware
- Inexpensive to deploy: uses already deployed secure hardware
- Open: allows anyone to design and deploy new credential algorithms
- Secure: isolates credential programs by default

... and can help realize the vision of personal trusted devices
strong yet scalable user authentication

Current status
- Prototype for Nokia S60 3.1+ devices (installable as a software package)
  - Full use of secure environment currently possible on N95 and N96 models
- Emulators and development tools will be available on request
- A number of open problems remain

How to build trustworthy information protection mechanisms (for devices, services and systems) that are simultaneously easy-to-use and inexpensive to deploy while still guaranteeing sufficient protection?