Secure Device Identity Tutorial

Mike Borza 

John Viega 

with Charles Qi 

Karen Zelenko 

Agenda

- Secure Device Identity – 100,000 foot view
- The 5 criteria
- The 10,000 foot view
- Related technologies
- Related presentations
  - Charles Qi, Broadcom Broadsafe HSM
  - Karen Zelenko – Phoenix Technologies security modules

The 100,000 Foot View

What is it?

- Unique per-device identifiers
  - Method or methods for authenticating that the device is uniquely bound to that identifier
  - Manufacturing ID may be augmented with Locally Significant Identity (LSID)
- What is a device?
  - *"Anything with an identity"* – Norm Finn
- Distributed trust model, no central registry
  - no extra administration or cost within IEEE to implement

Why is it needed?

- No standard ways to identify devices across network architectures
- MAC addresses are not sufficient
  - Multiple per device
  - Reconfigurable
  - Not cryptographically bound
- Device identity is important for completing chains of trust
  - Closes a significant window of vulnerability

Where will it be used?

- **Network equipment provisioning**
- **Identification of devices joining a network**
- **Authenticated key exchange in other protocols**
  - E.g., 802.1AF, 802.1X
- Inventory management
- Internal component identification
- LLDP chassis IDs
- …

The 5 Criteria

1 – Broad Market Potential

- Potentially applicable in any 802 protocol
  - also applicable to non-802 network protocols
- Specifically motivated by 802.1AE/AF need to authenticate network nodes
- By intent, will be light and robust enough to implement in a vast range of network devices across the cost spectrum

2 – Compatibility

Will maintain compatibility with

- 802 Overview and Architecture
- 802.1D, 802.1Q, 802.1f
- Managed objects consistent with 802.1 practice

3 – Distinct Identity

- No existing 802 standard provides this capability across different network architectures
- A single framework will be defined
- The standard will be self-contained and complete

4 – Technical Feasibility

- Existing identity modules provide a proof-of-feasibility for the technology
- No unknown functionality is envisaged to complete development of this standard

5 – Economic Feasibility

- The envisaged solution adds a small amount of cost to the devices and equipment that incorporates it
- There are potential savings on operations and asset management that may offset these costs
- Secure Device ID enabled devices may be deployed incrementally in existing networks

The 10,000 Foot View

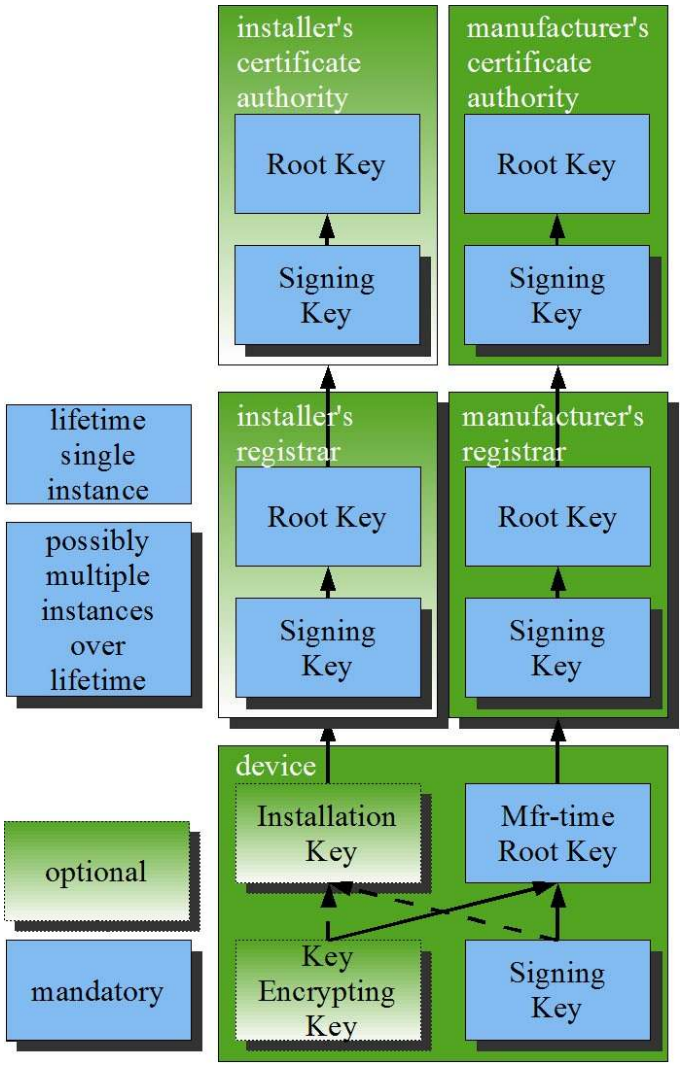
How are Identities used today?

- During provisioning
  - to associate subscriber data for a network service with devices they use for access
- During network connection establishment
  - to authenticate devices attaching themselves to the network – usually
  - to authenticate the network to its subscribers – sometimes (but should be always!)
  - to limit exposure of the network to attacks from unauthorized users
- During connection teardown
  - to protect users from attacks that try to force a disconnect

What is being identified?

- There has been a tendency to be less than precise
  - a port is associated with equipment it is installed in... is associated with users of that equipment
  - this has sometimes been implicit in the use of the identity, not actually intended
- For DevID purposes in 802 networks, an identity could be associated with any network device
  - e.g. port, bridge blade, chassis, network host
  - devices may be physical or virtualized, e.g. ports in MultiAccess MACsec, virtual machines
  - recall, devices are uniquely distinguishable
    - globally unique device identities in multi-port systems can be derived from this DevID
    - many ports could possibly be bound to a single device ID

Device ID Chain of Trust



- Creators of Device IDs must have access to a PKI
  - There is no central root for this PKI, instead many PKIs
  - frees manufacturers to deploy device ID on their own schedule
  - Service operators, enterprises that use Locally Significant IDs must likewise have access to a PKI

Probable Implementation Technologies

- KISS – objective will be to define a single cryptographic binding object
  - no optional cryptographic operators
  - possibly single required strength for any operator
  - Asymmetric cryptography – RSA most likely
    - most broadly deployed at present, offers greatest level of reuse
    - but not the lightest operator to implement
  - Hash operator – (HMAC)SHA-n
  - AES if necessary, e.g. for key wrapping
  - X-509 certs – ugly, but they work and are everywhere

Other considerations

A complete standard must also deal with

- initial creation of manufacturer and LSIDs
  - includes in-house, out-source and ODM issues
- allowed uses
- storage requirements for sensitive data
- end-of-life issues
  - credential destruction and/or revocation
  - credential re-issue
  - use of devices following credential re-issue

Related Technologies

Why do we care?

- Leverage the work of others where we can
  - use what's successful where we can
  - avoid what didn't work or was broken
- Provide compatibility across a variety of network types
- Backwards compatibility with existing hardware where possible

802.11i Wireless LAN

- It punts! Establishment of identity occurs during authentication using 802.1X with unspecified EAP method
  - RSN (robust secure networks) only
  - alternatively through pre-shared keys
- Implementors define both the identity objects and methods that pertain to them
- Typically multiple implementations are installed in a single end station if multiple services are accessed through different access points

802.16 Broadband Wireless

- PKMv1 in Rev. 2004
  - supported only one-sided strong identification
    - stations are authenticated to the network
    - no corresponding authentication of the network ==> rogue basestations can hijack users
- PKMv2 in Rev. E
  - mutual authentication based on X.509 certificates
  - includes key negotiation algorithms during connection establishment

802.16 X.509 Cert Format

X.509 v3 field	Description
tbsCertificate.version	Indicates the X.509 certificate version. Always set to v3 (value of 2).
tbsCertificate.serialNumber	Unique integer the issuing CA assigns to the certificate.
tbsCertificate.signature	Object identifier (OID) and optional parameters defining algorithm used to sign the certificate. This field shall contain the same algorithm identifier as the signatureAlgorithm field.
tbsCertificate.issuer	Distinguished Name of the CA that issued the certificate.
tbsCertificate.validity	Specifies when the certificate becomes active and when it expires.
tbsCertificate.subject	Distinguished Name identifying the entity whose public key is certified in the subjectpublic key information field.
tbsCertificate.subjectPublicKeyInfo	Field contains the public key material (public key and parameters) and the identifier of the algorithm with which the key is used.
tbsCertificate.issuerUniqueID	Optional field to allow reuse of issuer names over time.
tbsCertificate.subjectUnique ID	Optional field to allow reuse of subject names over time.
tbsCertificate.extensions	The extension data.
signatureAlgorithm	OID and optional parameters defining algorithm used to sign the certificate. This field shall contain the same algorithm identifier as the signature field in tbsCertificate.
signatureValue	Digital signature computed upon the ASN.1 DER encoded tbsCertificate.

- Manufacturer or network operator installs prior to provisioning
  - for security, critical parts should be stored in protected NV storage
- 1024 or 2048 bit RSA used for public key algorithm
  - slow or expensive in small hardware, esp. 2048 bit
- HMAC SHA-1 used in authentication schemes
  - robust, commonly available

DOCSIS

- used for cable modem data services subscriber authentication
- also uses embedded X.509 certificates
  - past security issues caused significant tightening of requirements for storage and use of these
  - only manufacturers may install certs
- requires a PKI with authentication based on Radius
- cryptography: RSA-1024 for PK operations

TPM

- Trusted Computing Group develops standards for Trusted Platform Module (TPM)
- TPM engenders many credentials, incompletely...
  - Endorsement Credential (Endorsement Key EK)
    - unique per TPM, generated in manufacturing
  - Platform Credential
    - incorporates the EK public key, making it a unique identifier of the platform
  - Identity Credential (Attestation Identity Key AIK)
    - signed by a CA used to provide anonymous “attestation”

Possible approaches to DevID with TPM

- Use the Platform Credential as a DevID
  - contains extra information not needed, but not necessarily prohibited
- Derive a DevID from the PC
- Derive a new credential unique to DevID, unrelated to the PC except by platform association
  - presumably stored as a protected BLOB outside the TPM

Why not simply adopt TPM?

- TPM includes features well beyond what's required to meet the minimum requirements of DevID
- TPM in gates will be larger than desirable for the bottom tiers of devices
- A full (compliant) TPM implementation presupposes the existence of a PKI with both distributed certificate authorities and a trusted third party to distribute and/or authenticate AIKs

Other Presentations