Process Control Systems Industry Conference

Intuitive, Easy, and Secure Provisioning and Configuration of Industrial Sensor and Control Networks

René Struik
Certicom Research

E-mail: rstruik@certicom.com

ABSTRACT

We present an intuitive, yet secure approach to provisioning and configuration of sensor networks. The approach hides security details from the user, thus allowing ease of device and network setup and flexibility of trust lifecycle management, while doing away with security hassles that have plagued more conventional approaches.

The proposal uses public-key cryptography, a security technology that allows reduction of trust lifecycle management to the management of trusted device identities (via so-called certificates), and security policy enforcement techniques based on lifecycle management of device roles.

Outline

- Networks vs. adhoc wireless networks
  - Networks and applications
  - Constraints
- Security Architectural Elements
  - Overview
- Pre-operational usage
  - Configuration
  - Initial setup
- Deployment scenarios
- Features and Benefits
- Cost
- The bigger picture

Networks vs. adhoc wireless networks

Outline

- Networks and applications
- Constraints

Underlying network technology (varying details)

Communication technology

- Transmission technology (e.g., frequency band for radio transmissions if wireless)
- Anticipated data rates (including, e.g., breakdown depending on device type)
- Hearing range (i.e., range over which direct communication between static and moving devices may be possible without routing)
- Cost factors (e.g., low power, low cost, low complexity devices)
- Special communication types (e.g., beacon and beacon-less communication)

Devices

- Anticipated application area of devices, networks

Network Characteristics

- Network topology (e.g., star network, mesh networks)
- Communication patterns (e.g., peer-to-peer, multicast, broadcast)
- Devices and their roles and capabilities (e.g., ordinary device, network coordinator)
- *Intra*-network behavior and *inter*-network behavior (roaming)

Interaction with outside world

- Gateway to other networks (e.g, node that communicates MAC service data units back and forth)
- Access to infrastructure (e.g., no online access to external infrastructure (external to network))

NOTE: Presentation relies on concepts only, but assumes distinct network coordinator role

Network security objectives

Access control to the network itself

Restriction of access to scarce network resources to authorized devices only, to ensure objectives including the following:

- proper bandwidth allocation;
- protection of commands (e.g., those regulating network membership);
- power drain savings.

Control of access to message traffic between network devices

Restriction of access to information secured between members of a group of network devices to precisely these group members. This includes any of the following objectives:

- Confidentiality:
  Prevent external parties from learning the content of exchanged messages.
- Data integrity/message authentication:
  Prevent external parties from modifying or injecting messages in undetected way.

- **Other** …

Wireless networking standards

Wireless Local Area Networks (WLAN)
- IEEE 802.11 family (11-55 Mbps and higher)
- WiFi Alliance

Wireless Personal Area Networks (WPAN)

Medium-rate WPAN
- Bluetooth: 1 Mbps
- Bluetooth Lite (upcoming)

High-Rate WPAN
- IEEE 802.15.3: 11-55 Mbps
- WiMedia Alliance, UWB Forum: up to 450 Mbps (using UWB)

Low-Rate WPAN
- IEEE 802.15.4: 20, 40, 250 kbps
- IEEE 802.15.4a
- ZigBee Alliance

Fixed/mobile broadband access standards (802.16, 802.20), 4G, 3GPP, etc.

Ubiquitous computing standards, DRM, etc.

Applications (sensor networks)

- Toys and games
- Consumer electronics
- PC peripherals
- Home automation
- Building automation (HVAC)
- Healthcare
- Automotive
- Industrial
- Supply chain management
- Asset tracking & localization
- Homeland security
- Smart dust …

Potential of wireless sensors for industry [US Dept of Energy (Dec 2002)]

- efficiency improvement: 25%; emission reduction: 10%
- significant reduction of 'wiring cost'; current range: from $50-100 to $2,000 per foot (if hazardous environment)

2001 figure: 'wiring cost' of $100B with $11B sensor market [WINA (2003)]

Constraints (1)

Constraints for sensor networks

High throughput is not essential, but rather

- Low energy consumption:
  Lifetime of 1 year with 2 AAA batteries (@750 mAh, 2V) yields 85µA average power consumption, thus forcing 'sleepy' devices (802.15.4 uses 40-60 mW for Tx/Rx)
- Low manufacturing cost:
  Low cost devices force small memory, limited computing capabilities
  (clock frequency: 4-16 Mhz; 10-32 kbytes ROM, 1-4 kbytes RAM, possibly no flash)

Constraints for adhoc networks

- No centralized management:
  No online availability of fixed infrastructure (so, decentralized key management)
- Promiscuous behavior:
  Short-lived communications between devices that may never have met before
  (so, trust establishment and maintenance difficult)
- Unreliability:
  Devices are cheap consumer-style devices, without physical protection
  (so, no trusted platform on device)

Constraints (2)

Security constraints for adhoc networks

- Decentralized key management:
  Due to no online availability fixed infrastructure, but also very ‘sleepy’ nodes
- Flexible configuration and trust management:
  Due to promiscuous, adhoc behavior, but also survivability requirements.
- Low impact of key compromise:
  Due to unavailability of trusted platform (tamper-proofing, etc.)
- Automatic lifecycle management:
  Due to virtual absence of human factor, after initialization

Security constraints for sensor networks

- Implementation efficiency: protocols should use similar cryptographic building blocks
- Parallelism: design protocols have the similar message flows
- Low communication overhead: protocols must avoid message expansion if possible

Constraints (3)

Chip cost considerations

Hardware vs. software cost (in USD)[1]

- Cost metrics of 0.33μ CMOS technology:
  1k gates ≈ 0.5¢
  1 pin ≈ 0.5¢
  RAM = 8 ROM
  1 kbytes ROM ≈ ½ mm^2
  15k gates ≈ 1 mm^2

- Cost *x* kbytes flash memory ≈ $a + b\,x$, where *a* is big constant (4-5 extra processes)

<u>Note:</u> Cost mainly determined by chip area (if mass produced)

- <u>Cost ratios:</u>
  (1 kgates : 1kROM : 1 kRAM) = (1 : 7.5 : 60)
- <u>Impact process technology:</u>
  Change from 0.33μ → 0.18μ CMOS technology yields *area* → 0.3 *area*.

[1]Figures indicative only, courtesy Melexis Telecom

Constraints (4)

Cryptographic building blocks

- Symmetric: all based on fixed block cipher, e.g., AES-128
- Public-key: all based on fixed elliptic curve, e.g., binary Koblitz curve K-163

	Block Cipher (AES)	ECC (K-163)
Software	ROM: 800-2,300 bytes RAM: 60 bytes	ROM: 3-5 Kbytes[1] RAM: 300 bytes
Hardware	≈ 6-7,000 gates	≈ 3,500 gates ROM: 1-2 Kbytes

[1]Figures on ATMel ATMega 128L processor, 16 MHz, 128K RAM

Hardware implementation cost

- AES-128. Cost: 3.5¢ → 1¢ (0.33μ → 0.18μ)
- K-163 curve. Cost: 2¢ → ½ - ¾¢ (0.33μ → 0.18μ) {Note: for K-283: 3¢ → 1¢}

Software implementation cost

- AES-128. Cost: 3 – 8.5¢ → 1 – 3¢ (0.33μ → 0.18μ)
- K-163 curve. Cost: 11 – 19 → 3.5 – 6¢ (0.33μ → 0.18μ)

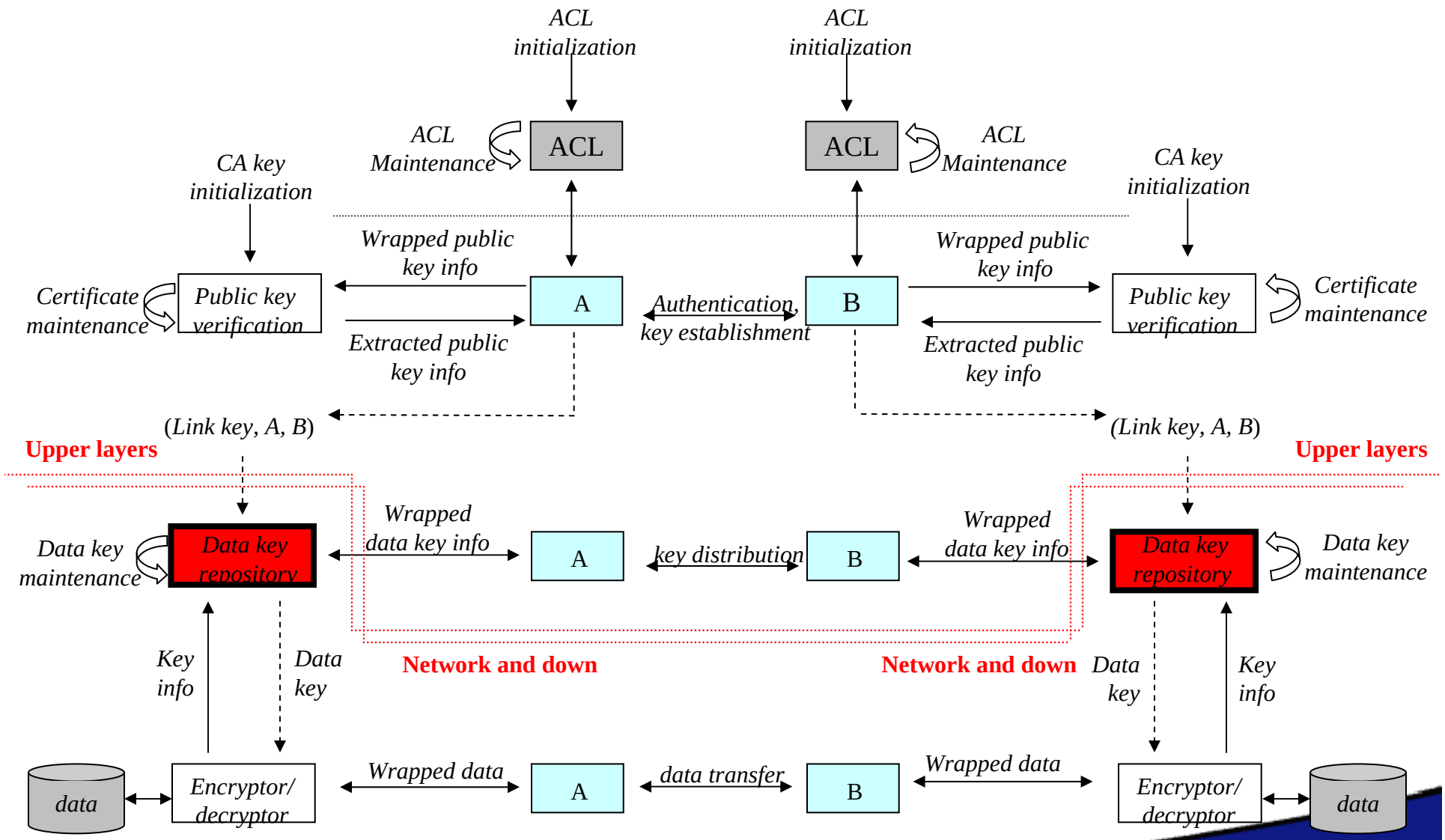
Hardware: significantly lower power consumption

Security Architectural Elements

Outline

- Overview
- Key establishment
- Key transport
- Data transport
- Security policies

Security Architectural Framework: Overview



Security Architectural Framework: Authorization (1)

Authorization and key establishment is based on each of the following:

(1) Evidence regarding the true identity of the other device;

(2) Evidence whether one wants to communicate with this explicitly identified device.

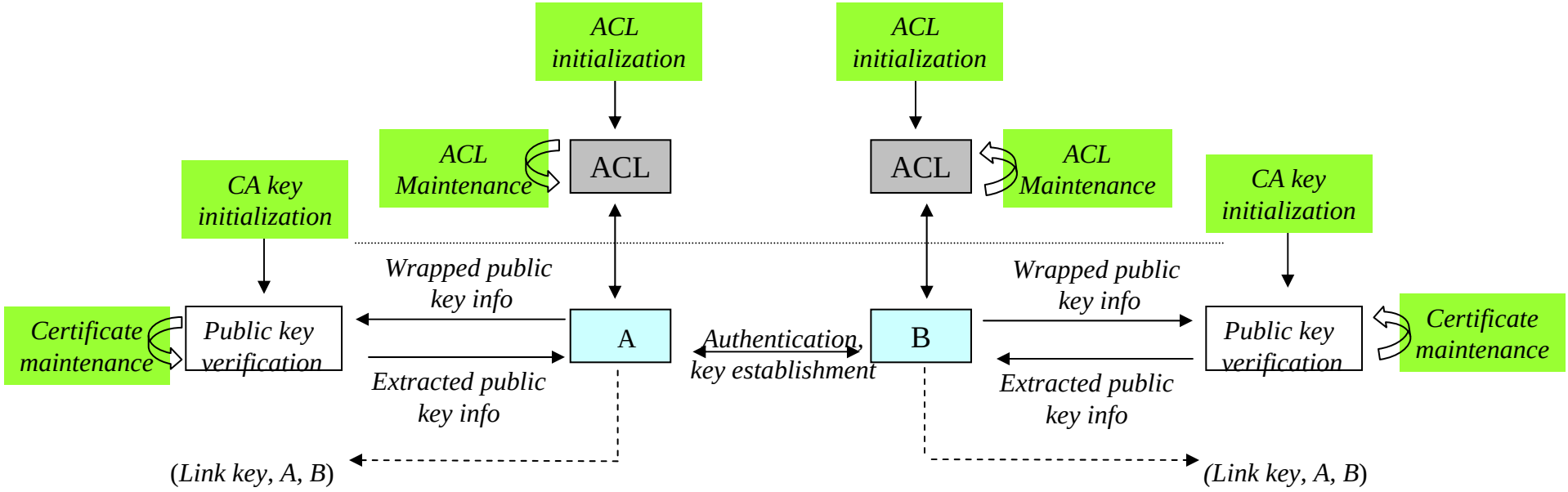
Cryptographic mechanisms:

1. Public-key *key establishment* mechanism. Derivation of link key between two devices, based on authentic public keys of both parties, including evidence on whom this link key is shared with.
2. Symmetric-key *key establishment* mechanism. Derivation of link key between two devices, based on secret and authentic pre-shared secret key between both parties, including evidence on whom this link key is shared with.

Non-cryptographic mechanisms:

- Acceptability *test*. Establishment whether a particular device is to be accepted, based on a membership test of a so-called Access Control List (ACL).

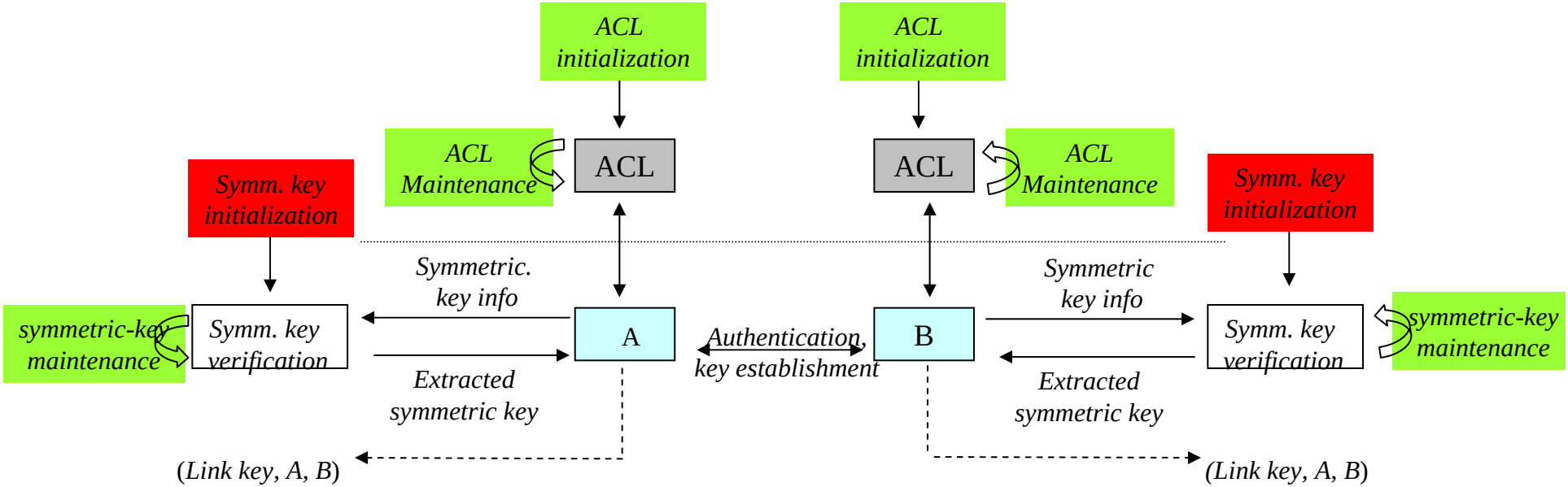
Security Architectural Framework: Authorization (2a) (public-key scenario)



Notes:

- The authentication protocol establishes a symmetric link key between the devices (since it is an authenticated key establishment protocol).
- Authentication is based on authentic public keys.
- Authorization is based on non-cryptographic acceptability test by either party.

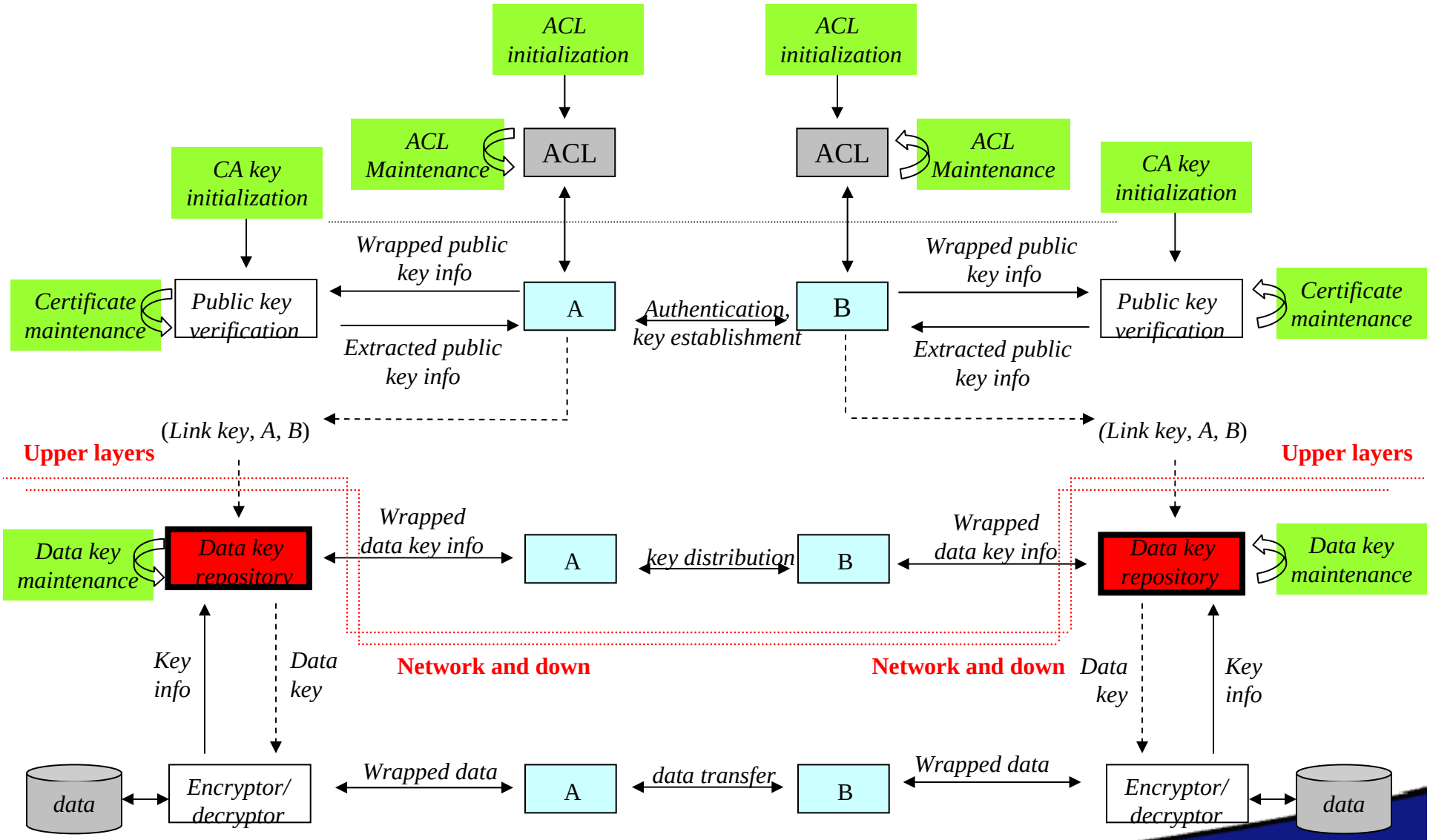
Security Architectural Framework: Authorization (2b) (symmetric-key scenario)



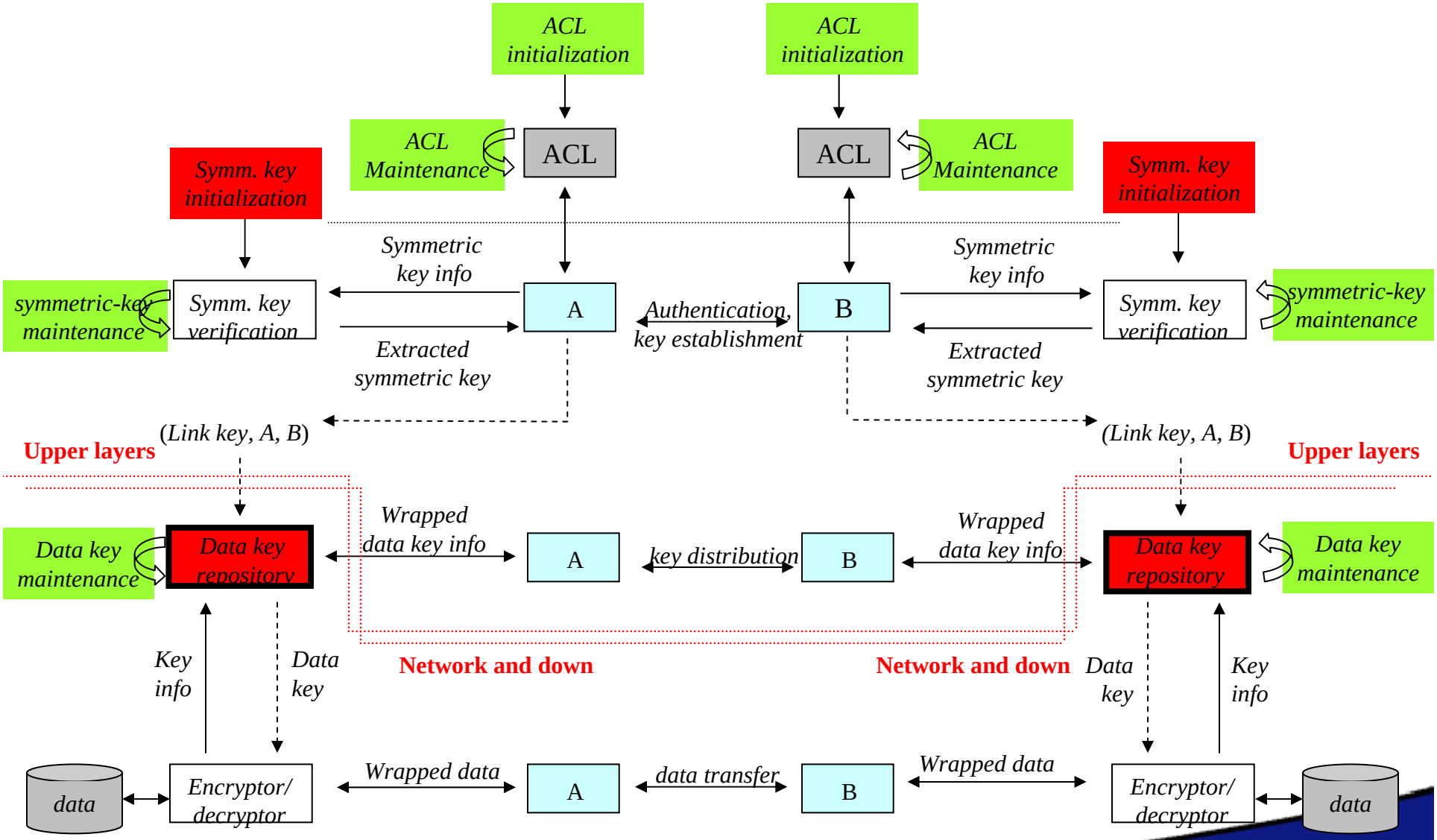
Notes:

- The authentication protocol establishes a symmetric link key between the devices (since it is an authenticated key establishment protocol).
- Authentication is based on authentication of pre-shared secret keys.
- Authorization is based on non-cryptographic acceptability test by either party.

Security Architectural Framework: Public-key scenario



Security Architectural Framework: Symmetric-key scenario



Pre-operational usage

Outline

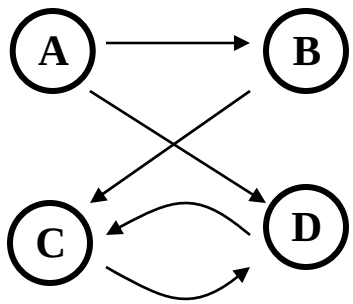
- Configuration
- Extended role model
- Initial setup

Configuration management (1)

Configuration

Information on which entities are supposed to communicate with which other entities.

Without endpoints:



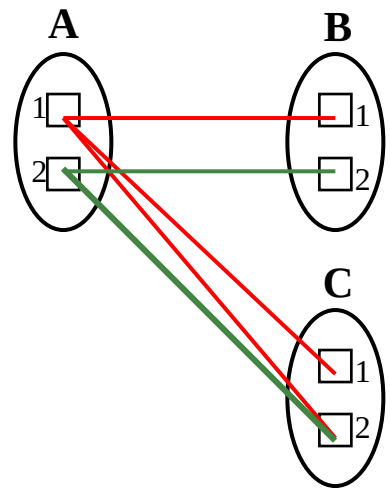
Each device keeps track of its own local neighborhood:

A → B, A → D;

B → C;

C → D; D → C.

With endpoints:



Each device keeps track of its own color scheme:

A1: red → B, C; A2: blue → B, C;

B1: red → A; B2: blue → A;

C1: red → A; C2: red, blue → A.

Process Control Systems Industry Conference

Configuration management (2)

Configuration manager

Responsible for selecting which entities are supposed to talk with which other entities:

- Definition of groups of devices that are supposed to communicate with each other;
- Communication of group membership to each individual member hereof.

Note: This may include allocation of communication settings (e.g., channel hopping sequences, transmit schedules) or application settings (e.g., sensing data poll schedules).

Configuration mechanisms

- via external configuration tool;
- via service discovery process.

Trust requirement

Each device needs to trust its configuration manager(s), in the sense that it does not hook it up with spy-ware or other devices that are supposed to be kept out of the loop.

Note: *Independent verification possible*

One can verify whether a configuration manager has done its job properly, i.e., has set up proper 'virtual wiring', without access to keying material or other privileged info (e.g, by querying the device).

Configuration management (3)

Selection of configuration manager

- Each device may choose its own set of favorite configuration managers (*ConfigSet*);
- This list should be adaptable by a device with the proper credentials (*Config-Δ-Set*). Each such device is a member of *ConfigSet*.

Trust guarantee options

- Evidence that info indeed originates from purported configuration manager;
- Inspection of conformance of configuration afterwards ('independent verification').

Notes:

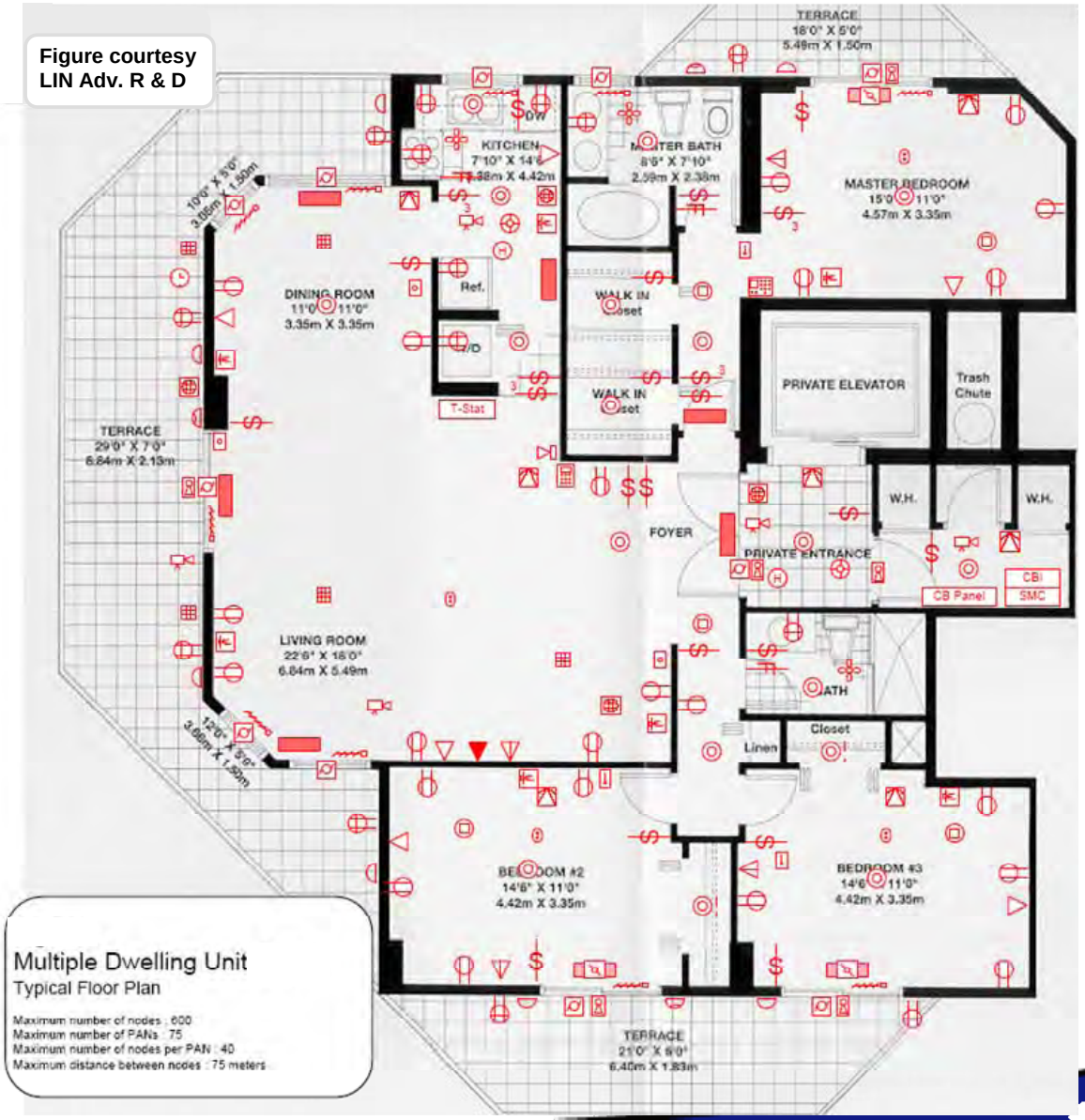
- If *Config-Δ-Set* is the empty set, this freezes the current set of configuration managers;
- If *ConfigSet* is the empty set, the configuration is fixed towards the future.

Initial setup via finite state machine (example – resurrecting duckling policy)

- Initial state: the sets *ConfigSet* and, hereby, *Config-Δ-Set* are the empty set;
- Bootstrap mechanism: non-cryptographic introduction of configuration manager to device (thus provoking state change from initial state to other state); *) tricks omitted!
- Return to initial state: via erasure of *ConfigSet* by member with proper credentials.

Configuration management (4)

Example:



Configuration management (5)

Assisted configuration (with help of external configuration device)

Step 1 (‘virtual wiring’)
Define the groups of physical devices that are supposed to communicate with each other.

Step 2 (device identification)
Determine the physical address of each device to be configured.

Step 3 (bootstrap mechanism)
Imprint each device with the physical address of the configuration device.

Step 4 (configuration step)
Send each device the configuration information corresponding to the groups it belongs to

Step 5 (verification step)
Verify the conformance of the device configurations via one of the trust guarantee options (i.e., via independent inspection or via data origin authentication).

Configuration management (6)

Manual configuration (without help of external configuration device)

Same as assisted configuration with help of external configuration device!!!
(simply, take one of the network devices to be the configuration device)

Note 1: The main difference between assisted and manual configuration is with the configuration step (Step 4):

Assisted configuration relies on an external configuration device with, usually, a far more powerful user interface (useful for conveying detailed configuration information) than would be possible with a network-internal device.

Note2: Another difference might be with *form factors* of the configuration device. An external configuration tool may be tailored towards ease of use, whereas a network device might have limitations (e.g., dimensions, weight) that may limit ease-of-use or prevent it altogether (e.g., try lifting a compressor or a heat pump!).

Configuration management (7)

Assisted configuration (with help of configuration device) – cont'd (ii)

Flexibility:

- *Parallelization.* The role of the configuration manager can be performed by multiple configuration managers in parallel, without requiring much coordination:

  Each configuration manager only needs to implement the assisted configuration of those devices allocated to him.

- *Configuration freeze*. The configuration setting can be locked by the configuration manager by setting the set *ConfigSet* to the empty set (if he has the proper credentials).

- *Limitation of liability.* Control can be handed over to, e.g., the network owner by setting the set *ConfigSet* to represent this network owner only (if he has the proper credentials). The configuration manager now does not carry responsibility/liability for leakage of security material (only for hooking up the correct devices, which can be checked).

Configuration management (8)

Assisted configuration (with help of configuration device) – cont'd (iii)

Security:

- *Unsecured configuration (no crypto aboard).*
  Verification of proper configuration (Step 5) needs to be realized using independent (manual) inspection of the a postiori configuration settings of each device.

- *Secured configuration (public-key based techniques).*
  If a device shares a root key with its configuration manager, the configuration info (sent in Step 4) can be secured using the data key derived after executing an authenticated public-key key establishment protocol between those two devices.

- *Secured configuration (symmetric-key based techniques).*
  If a device shares a master key with its configuration manager, the configuration info (sent in Step 4) can be secured using the data key derived after executing a symmetric-key key establishment protocol between those two devices.

Configuration management (9)

Assisted configuration (with help of configuration device) – cont'd (iv)

Security:

- *Unsecured configuration (no crypto aboard).*
  No cryptographic mechanism for secure network operation available.

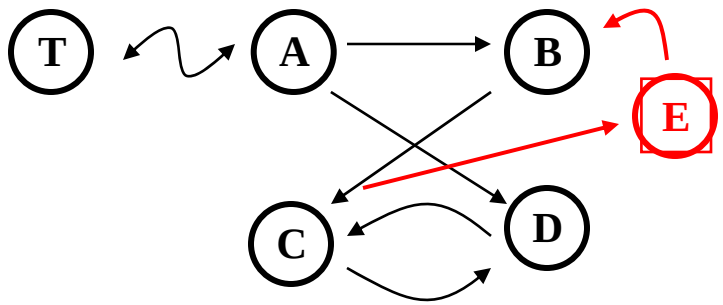
- *Secured configuration (public-key based techniques).*
  If a device shares a root key with each 'security manager' of each group in its configuration setting, then it may obtain all group keys automatically, without any further intervention (and, thereby, all keying material for secure network operation).

- *Secured configuration (symmetric-key based techniques).*
  If a device shares a master key with each 'security manager' of each group in its configuration setting, then it may obtain all group keys automatically, without any further intervention (and, thereby, all keying material for secure operation).

Configuration management (10)

Assisted configuration (with help of configuration device)

Example:



E_A={A → B, A → D};

E_B={B → C}; E_D={D → C};

E_C={C → D}; $\hat{E}_C$={C → D, C → E};

E_E={E → B}.

1. Bootstrap and device identification

– Devices A, …, D all accept T as configuration manager
– Configuration manager T learns address of devices A, …, D
– Devices A, …, D may set up secured channels with T

2. Virtual wiring

– Devices A, …, D may exchange capability info with T

3. Configuration step

– Devices A, …, D receive configuration information from T

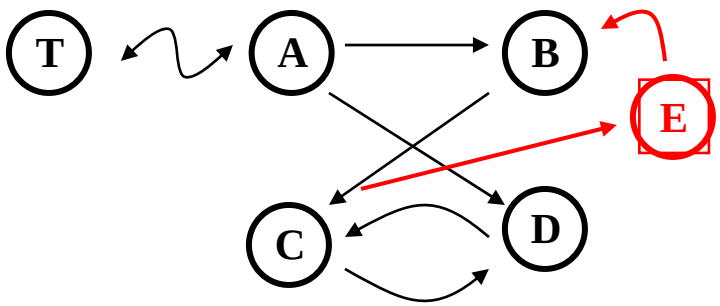
4. Verification step

Process Control Systems Industry Conference

Configuration management (10a)

Assisted configuration (with help of configuration device) – cont'd (v)

Example A: Public-key based techniques.



$CASet = \{\text{Root Key}_1, \text{Root Key}_2, \text{Root Key}_3\}$

	A	B	C	D	E
ConfSet	T	T	T	T	
Key est.	k_{AT}	k_{BT}	k_{CT}	k_{DT}	
Binding	E_A	E_B	E_C	E_D	
ConfSet					T
Key est.					k_{ET}
Binding			$\hat{E}_C$		E_E

$E_A = \{A \rightarrow B, A \rightarrow D\}$;

$E_B = \{B \rightarrow C\}$; $E_D = \{D \rightarrow C\}$;

$E_C = \{C \rightarrow D\}$; $\hat{E}_C = \{C \rightarrow D, C \rightarrow E\}$;

$E_E = \{E \rightarrow B\}$.

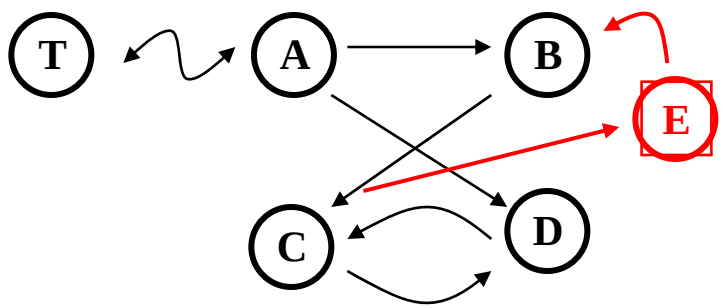
Added bonus: all other secure trust relationships automatic!

Process Control Systems Industry Conference

Configuration management (10b)

Assisted configuration (with help of configuration device) – cont'd (vi)

Example B: Symmetric-key based techniques.



	A	B	C	D	E
ConfSet	$T;K_{AT}$	$T;K_{BT}$	$T;K_{CT}$	$T;K_{DT}$	
Key est.	k_{AT}	k_{BT}	k_{CT}	k_{DT}	
Binding	E_A	E_B	E_C	E_D	
ConfSet					$T;K_{ET}$
Key est.					k_{ET}
Binding			$\hat{E}_C$		E_E

E_A={A → B, A → D};

E_B={B → C}; E_D={D → C};

E_C={C → D}; $\hat{E}_C$={C → D, C → E};

E_E={E → B}.

Disadvantage: all other secure trust relationships require involvement of some trusted party T'! (Here, T'=T if configuration manager and trust manager coincide.)

Process Control Systems Industry Conference

Operational Description – Set-Theoretic Definition of Roles

Informational elements (provided by device itself)

(1) *Global device Id.*

Each device has own *static* global device Id (e.g., IEEE MAC address).

(2) *Public key (in public-key scenario).*

Each device has its own public/private key pair (P_A, S_A).

{The public key P_A need not to be stored on the device itself.}

(3) *Access control list (ACL) (if desired).*

Each device has own set of devices it may wish to establish a secure peer-to-peer link key with.

(4) *TrustSet (in dynamic-trust scenario).*

Each device has own set of devices it trusts to assume the role of security manager.

(5) *CA-Set (in dynamic-trusted party scenario).*

Each device has own set of devices it trusts to assume the role of trusted third party.

(6) *ConfigSet (in dynamic configuration manager scenario).*

Each device has own set of devices it trusts to assume role of configuration manager.

(7) *ACL-Δ-Set (in dynamic-trust scenario).*

Each device has own set of devices it trusts to change the *ACLSet.*

(8) *Trust-Δ-Set (in dynamic-trust scenario).*

Each device has own set of devices it trusts to change the *TrustSet.*

(9) *CA-Δ-Set (in dynamic trusted party scenario).*

Each device has own set of devices it trusts to change the *CA-Set.*

(10)*Config-Δ-Set (in dynamic configuration manager scenario).*

Each device has own set of devices it trusts to change the *ConfigSet.*

Initial set-up (1)

<u>Bootstrap mechanism:</u> non-cryptographic introduction of device with specific role to other device (thus provoking state change from initial state to other state).

Pre:

A	B
A: x	B: y

Post:

B: y	A: x

(Here, x and y is authentic and/or secret information)

$$A \xrightarrow{\textit{challenge}} B$$

$$A \xleftarrow{\textit{response}} B$$

<u>Exchange of info:</u>

- Secret info via wire: resurrecting duckling security policy [Stajano, Anderson (1999)]
- Authentic info via limited channel [Smetters et al (2002)]
- 2-way authentic channel, 1-way secret and authentic channel, 2-way secret channel [Henk-Jan Hoekman (2004)]
- Proximity-based authentication [Chaum and Brands (1994), Struik (2004)]

Initial Set-up (2)

"Computer users have been taught for years that computer security systems can't be effective unless they are complex and difficult to use. In reality, this conventional wisdom is completely wrong."

— Lorrie Faith Cranor, Carnegie Mellon University

Public-key technology can make trust lifecycle management intuitive and hidden from the user.

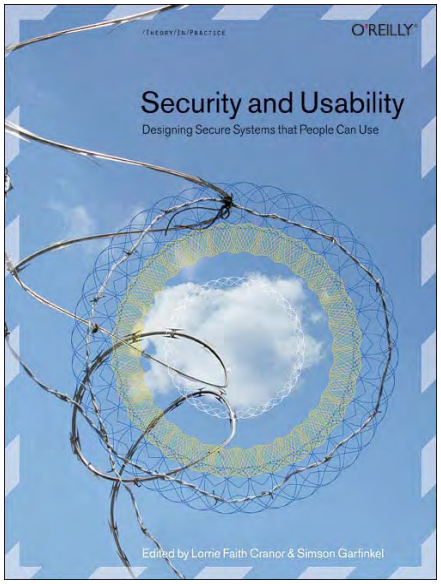
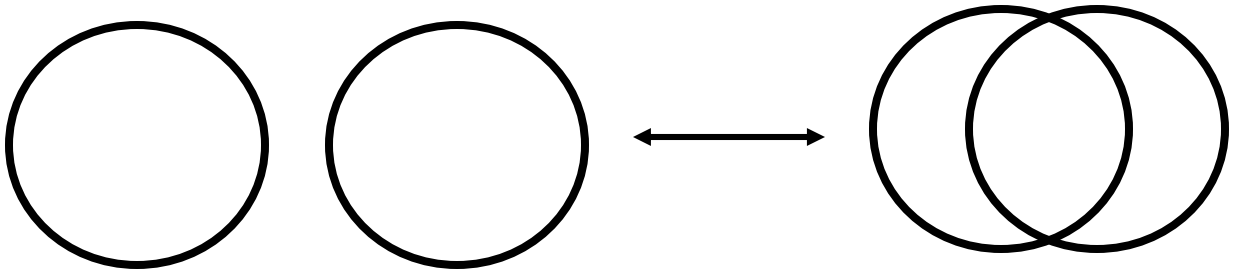


Figure 1. Connecting a laptop to a secured wireless network in 32 seconds. All the user has to do is briefly align the infrared ports of laptop and access point and press the Enter key twice. These are snapshots from a live Network-in-a-Box demonstration.

Source: D. Balfanz, G. Durfee, R.E. Grinter, D.K. Smetters, P. Stewart, "Network-in-a-Box: How to Set Up a Secure Wireless Network in under a Minute," in *Proceedings of the 13th USENIX Security Symposium*, August 9-13, 2004.

Ease of Configuration



Ease of configuration:
- Merging of networks
- Partitioning of networks
- Device portability and orphaning
- Hand-over of control (remote, backup)

Examples – Assignment of Device Roles

(1) Bootstrap mechanism:
CA $\rightarrow$ A: Imprint CA info.
<u>Pre-condition:</u> A is in virgin state (out-of-the-box).
<u>Post-condition:</u> $CA\text{-}\Delta\text{-}Set(A)=\{(CA, W_{CA})\}$ (i.e., only CA can introduce new CA's)

(2) Resurrecting duckling mechanism:
T $\rightarrow$ A: Imprint mother duckling info.
<u>Pre-condition:</u> A has no configuration manager (out-of-the-box).
<u>Post-condition:</u> $Config\text{-}\Delta\text{-}Set(A)=\{T\}$ (i.e., only T can introduce new configuration managers)

(3) Imprinting of configuration info:
T $\rightarrow$ A: Imprint configuration info.
<u>Pre-condition:</u> T is A's sole configuration manager.
<u>Post-condition:</u> $E_A=\{A \rightarrow B, A \rightarrow D\}$.

(4) Hand-over of configuration role:
T $\rightarrow$ A: Resign from configuration manager role and hand-over role to security manager T*.
<u>Pre-condition:</u> T is A's sole configuration manager.
<u>Post-condition:</u> $Config\text{-}\Delta\text{-}Set(A)=\varnothing$; $ConfigSet(A)=\{T^*\}$; $TrustSet(A)=\{T^*\}$.
(i.e., T* is A's fixed configuration manager, who also acts as security manager).

Collateral information on certificates

Outline

- Certificates
- Device relationships
- Certificate generation

Certificate management (1)

Device certification (with help of registration authority)

Step 1 ('virtual wiring')
Define the groups of physical devices that are supposed to get a device certificate.

Step 2 (device identification)
Determine the physical address of each device that is supposed to get a device certificate.

Step 2a (device initialization)
Determine the ephemeral public key of each device that requests a device certificate.

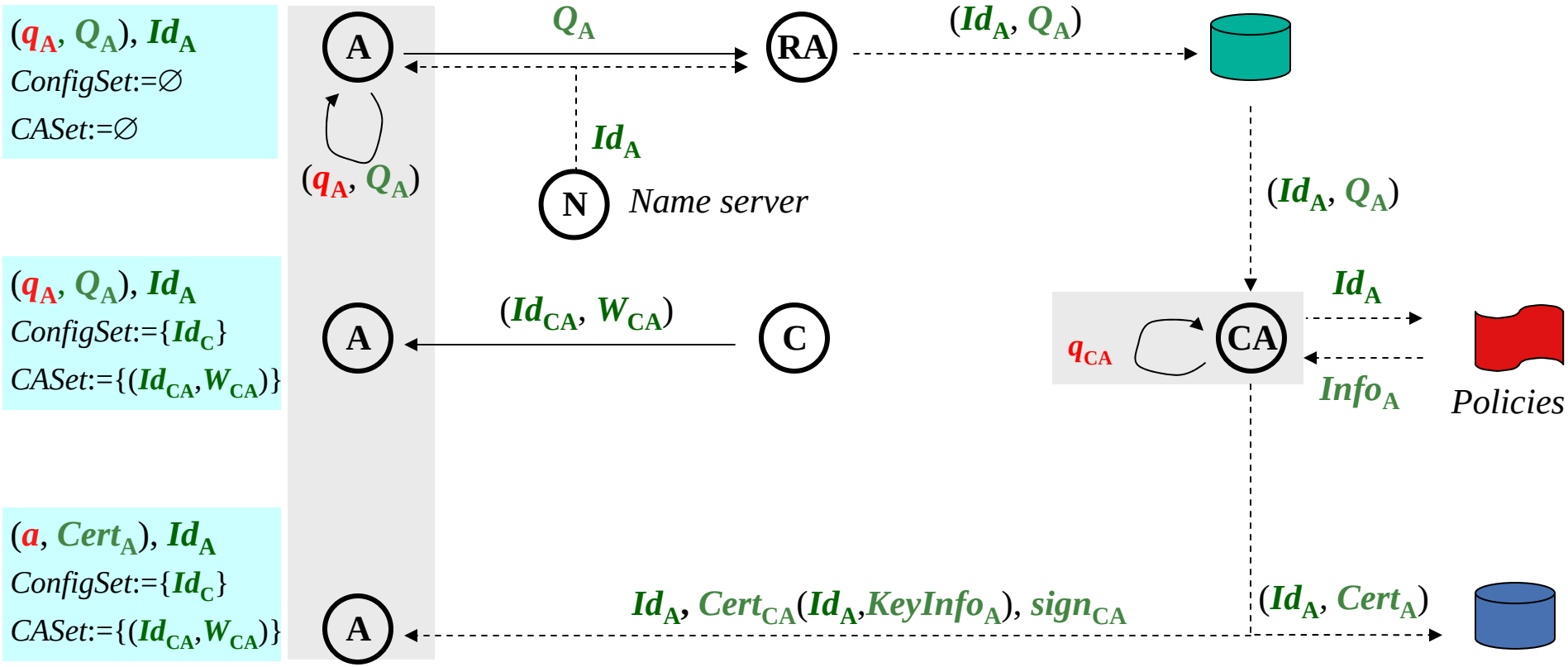
Step 2b (device registration)
Logically bind the physical address and ephemeral key of each device.

Step 3 (bootstrap mechanism)
Imprint each device with the physical address and root key of the certificate authority.

Step 4 (certificate generation step)
Send each device the device certificate corresponding to the certificate authority.

Step 5 (certificate verification step)
Verify the conformance of the device certificate via one of the trust guarantee options (i.e., via independent inspection or via certificate verification).

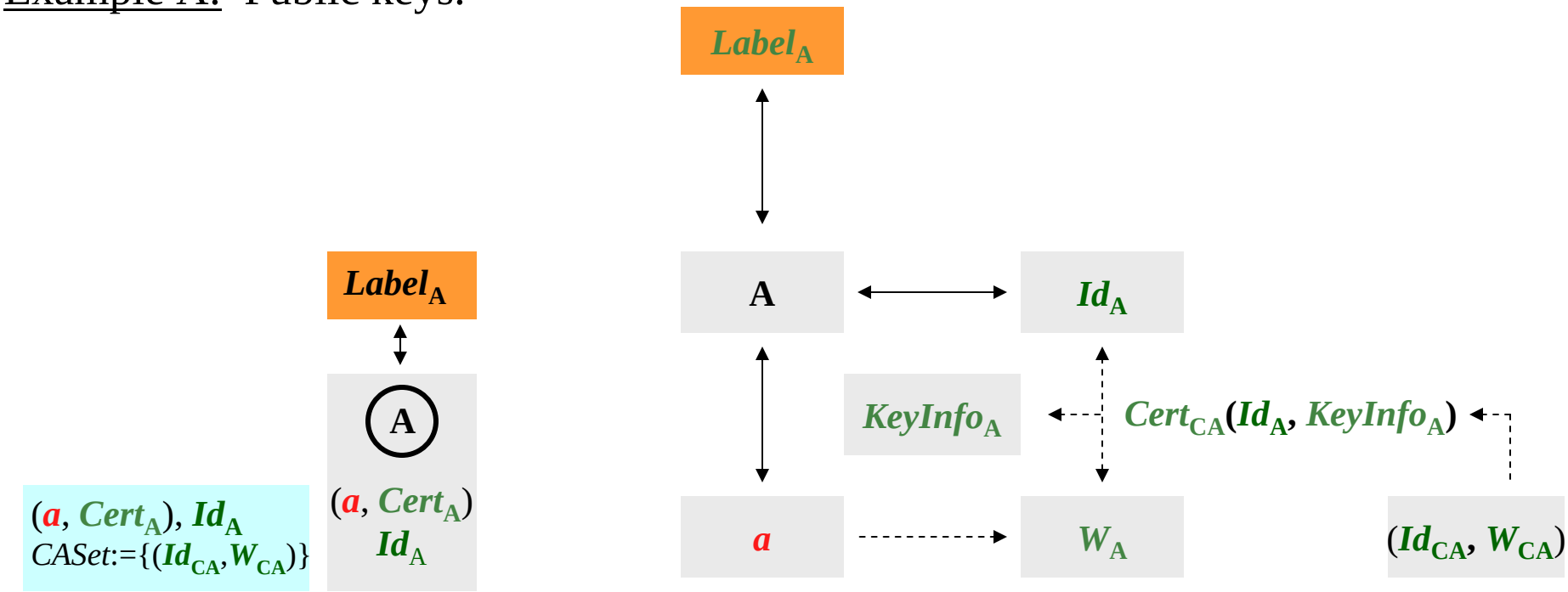
Certificate Generation (1)



Device roles		Keying information		Device information	
A	Ordinary device	(q_A, Q_A)	ephemeral private/public key A	Id_A	identifier of device A
RA	Registration authority	$(a, Cert_A)$	private key, resp. certificate A	Id_{CA}	identifier of CA
CA	Certificate authority	W_{CA}	public root key CA	Id_C	identifier of C
N	Name server	$sign_{CA}$	signature over certificate info		
C	Configuration manager	$KeyInfo_A$	keying information for device A		

Device relationships (1)

Example A: Public keys.



Device roles

A Ordinary device

CA Certificate authority

Type of binding

------- logical link

——— physical link

Keying information

$(a, Cert_A)$ private key, resp. certificate A

W_{CA} public root key CA

$KeyInfo_A$ keying information for device A

Device information

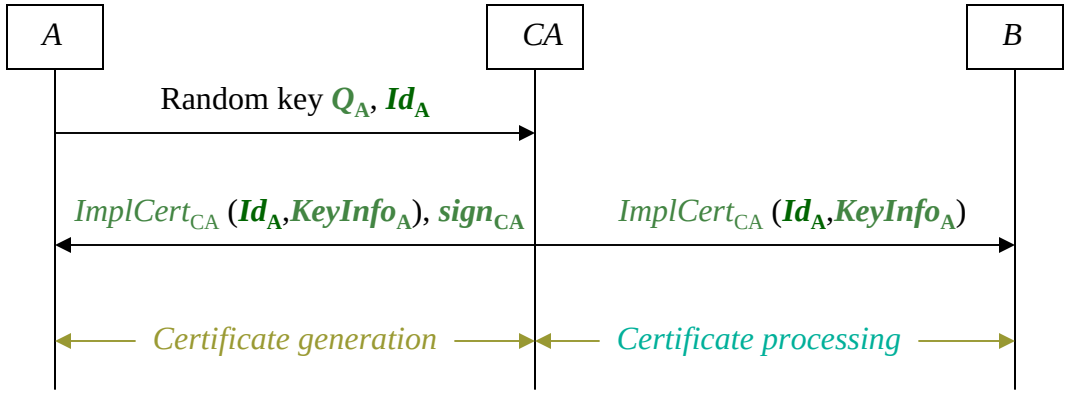
Id_A identifier of device A

Id_{CA} identifier of CA

$Label_A$ label of device A

Process Control Systems Industry Conference

Certificate Scheme (1)



- *Key contribution.* Certificate requester A randomly generates a short-term (ephemeral) public key pair and communicates this ephemeral public key to the other party (but not the private key).
- *Certificate generation.* Certificate authority CA computes the certificate based on the ephemeral elliptic curve point it received from the certificate A and based on evidence that it generated this public key itself. CA computes additional string that allows A to reconstruct its private key from the implicit certificate and this string (the so-called implicit signature).
- *Certificate verification.* Party A verifies whether the certificate it requested indeed originated from the CA, by reconstructing its public key in two different ways and checking whether either computation yields the same value (this involves a computation based on its implicit certificate, the implicit signature received, the ephemeral key contribution Q_A and the public key of the CA).
- *Key reconstruction.* Each party B may reconstruct a thus certified public key of another party A to obtain evidence that the only party that may have access to the corresponding key is, indeed, its perceived communicating party (this involves a computation based on the implicit certificate of the other party A and the public key of the CA).

Collateral information on initial symmetric-key keying material

Outline

- Initial symmetric-key keying info
- Device relationships
- Initial keying material generation

Symmetric-key keying material management (1)

Device initial keying (with help of registration authority)

Step 1 (‘virtual wiring’)
Define the groups of physical devices that are supposed to get a device master key.

Step 2 (device identification)
Determine the physical address of each device that is supposed to get a device master key

Step 2a (device registration)
Register the physical address of each device.

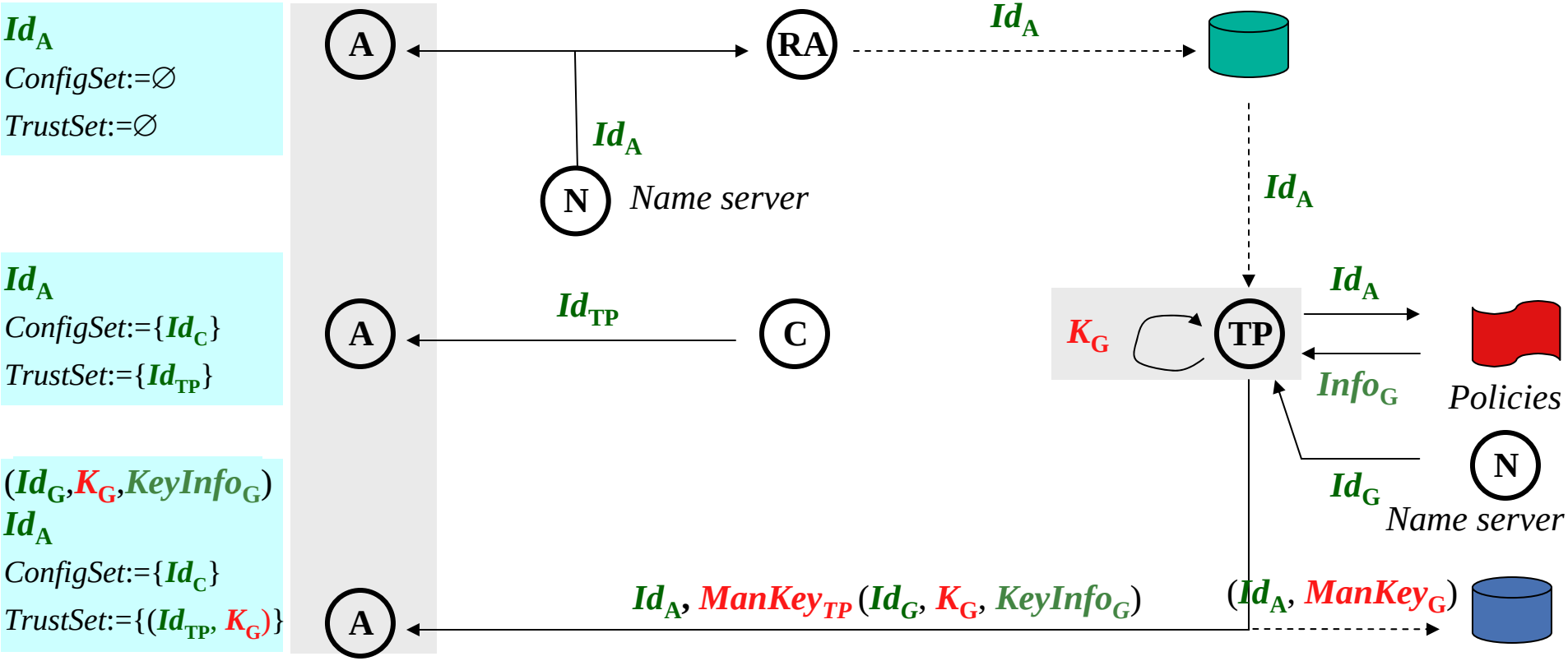
Step 3 (bootstrap mechanism)
Imprint each device with the physical address of the trusted party.

Step 4 (master key generation step)
Imprint each device with its device master key generated by the trusted party.

Step 4a (master key distribution step)
Imprint the master key with the security manager the device should communicate with.

Step 5 (master key verification step)
Verify the conformance of the device master key via non-cryptographic means (independent inspection).

Symmetric-Key Manual Keying Material Generation (1)



Device roles

A	Ordinary device
RA	Registration authority
TP	Trusted party
N	Name server
C	Configuration manager

Keying information

K_G	symmetric key for group *G*
$KeyInfo_G$	keying information for group *G*

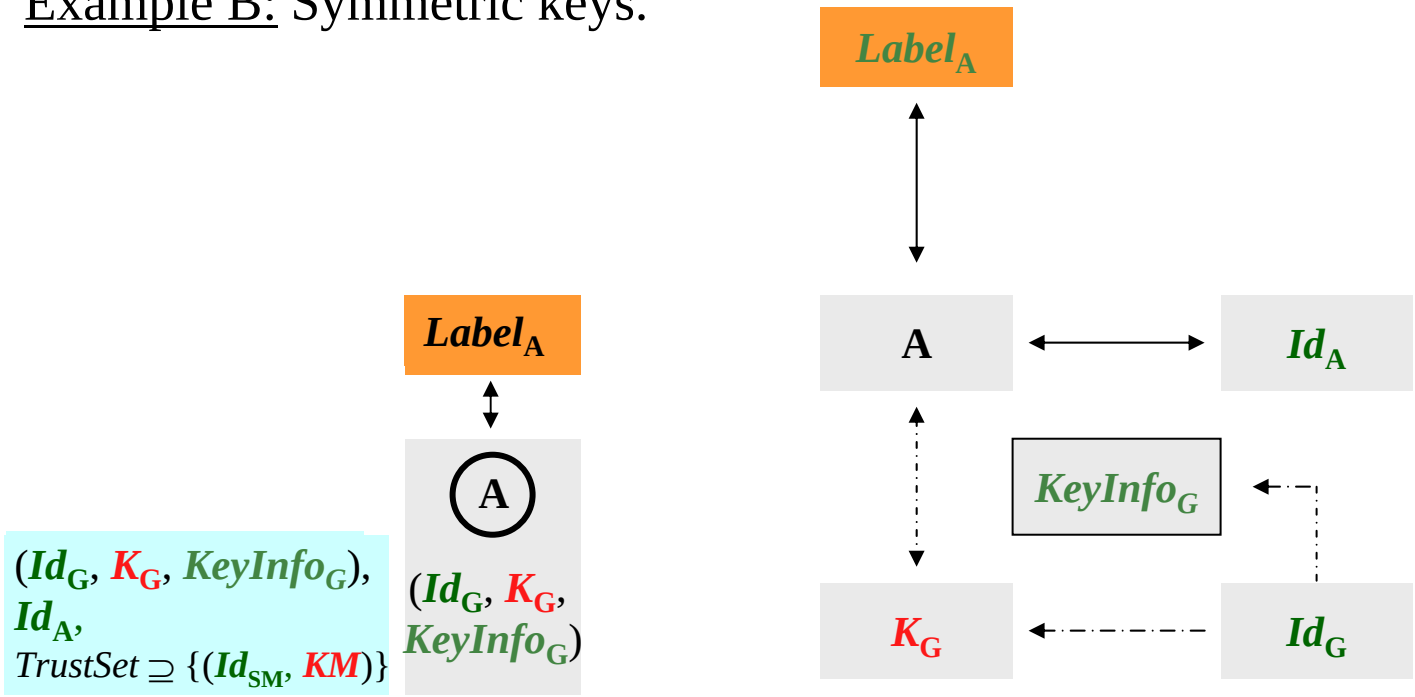
Device information

Id_A	identifier of device A
Id_{TP}	identifier of TP
Id_C	identifier of C
Id_G	identifier of group *G*

Process Control Systems Industry Conference

Device relationships (1)

Example B: Symmetric keys.



Device roles

A	Ordinary device
SecM	Security Manager

Type of binding

-·-·-·-	trust policy link
———	physical link

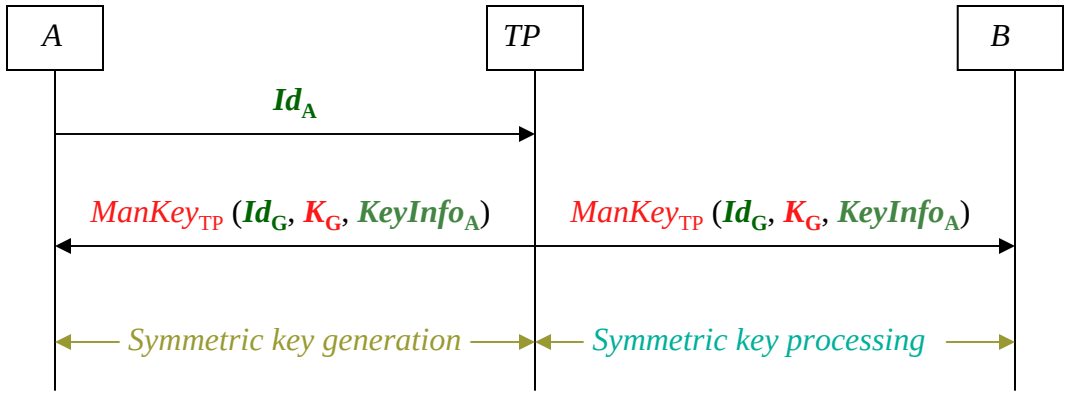
Keying information

K_G	symmetric key for group G
$KeyInfo_G$	keying information for group G
KM	pre-shared key between A and SecM

Device information

Id_A	identifier of device A
Id_{SM}	identifier of SecM
Id_A	identifier of group G
$Label_A$	label of device A

Symmetric-Key Manual Keying Information Scheme (1)



- *Key contribution.* Party A communicates his key request to trusted party TP.
- *Keying material generation.* Trusted party TP computes the initial keying material based on evidence that A generated this key request itself. This includes generation of a secret master key for A and some associated keying material (such as validity period, key originator, and other policy-related information).
- *Keying information verification.* Party A verifies whether the keying material it requested is properly formatted and receives non-cryptographic assurances that it originated from the TP.
- *Key reconstruction.* Each party B may reconstruct the symmetric key of another party A if it has access to the full keying material, including secret key, itself. Under no circumstances can it obtain any cryptographic evidence that the only other party that may have access to the corresponding secret key is, indeed, its perceived communicating party; instead, this involves non-cryptographic assurances from the TP.

Deployment scenarios

Deployment Scenarios (1)

Scenario #1:
mix-and-match of nodes from different vendors
Scenario #2:
addition of nodes to operational network
Scenario #3:
security audit
Scenario #4:
device repair and replacement (roaming in/out different user sites)
Scenario #5:
network separation (devices joining wrong network)
Scenario #6:
thwarting malicious attacks by (former) insiders
Scenario #7:
thwarting attacks by outsiders via insiders (held at ‘gunpoint’)
Scenario #8:
addition of subsystem (‘skid’) assembled elsewhere to operational network

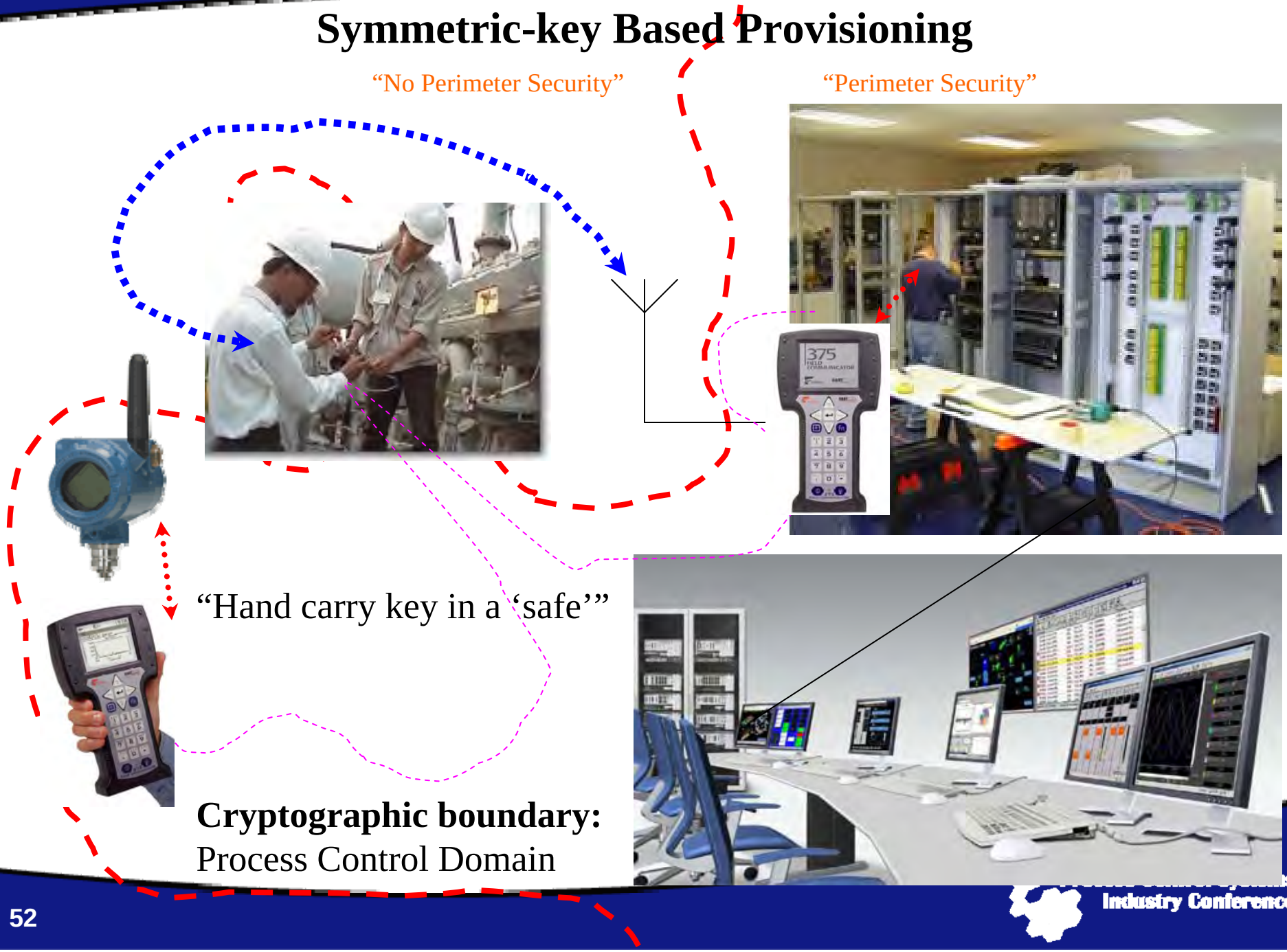
Evaluation criteria (1)

1. Does proposal implement 8 deployment scenarios – per-scenario comparison
2. Vulnerabilities of devices/subsystems in transit from vendor to user
3. Dependencies on trustworthiness personnel
4. Simplicity
5. Ease of use
6. Flexibility
7. Minimizes dependencies
8. System resources (bandwidth, time, energy consumption, join time)
9. Availability (24/7)
10. Human element
11. Externalities (robustness, do we need special boxes, tools, etc.)
12. Does facilitate over-the-air provisioning
13. Interoperability
14. Implementation cost (incremental cost)
15. Where cost occurred?
16. Ability to put more data on device (as required to bootstrap behavior on different layers)

Evaluation criteria (2)

17. Ability to put date on a device that does not have an 802.15.4 radio (e.g., gateway, backbone router, system manager), so that the same end state can be reached
18. Ability to de-provision device (i.e., bring back to ex-factory state and sell, e.g., on
    e-bay to someone who can then still securely use it)
19. Number of tools required
20. Ability to move from devices between sites (e.g., smart sea containers) – roaming
    in/out of different user sites
21. Economies of scale issues:

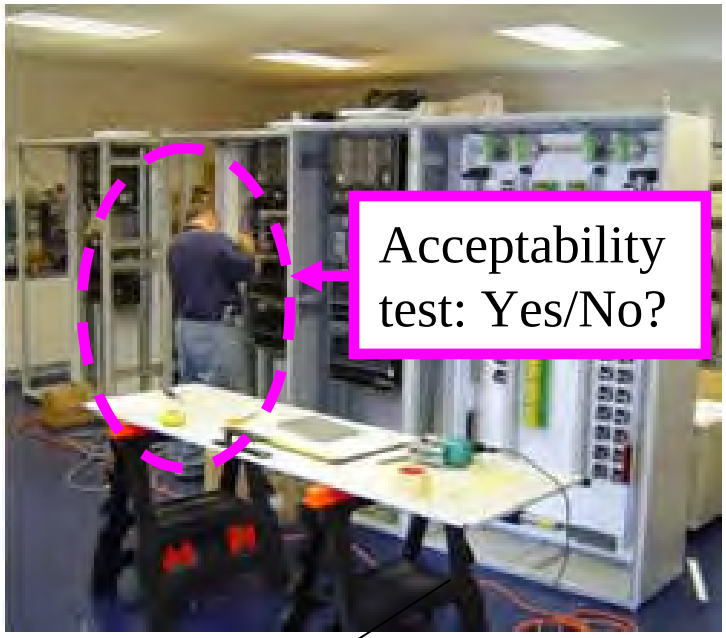
Symmetric-key Based Provisioning



Public-key Based Provisioning (1)

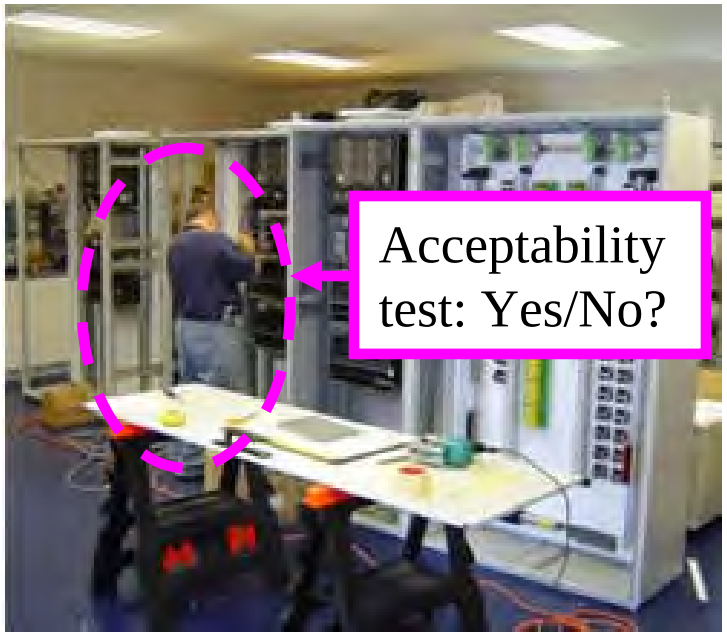
"No Perimeter Security"

"Perimeter Security"



Cryptographic boundary:
Device itself (intra-device secure module)

Public-key Based Configuration (2)



Acceptability test based on

- Device Id
- Tag Name
- Device Label
- Open enrolment
- Proximity-based techniques

…

Trust management via device identities

Trusted module

- **AES, ECC, RNG**
- **Security policy engine**
- **Storage of keys**

Features and Benefits

Features and Benefits (1)

Ease of use. Trust lifecycle management appears the *same* as that of an unsecured network and relies on

- proper identification of devices (e.g., reading off a label of physical module);
- proper management of device roles (e.g., adding these to, resp. removing these from a white list, e.g., via a workstation GUI).

Thus, trust lifecycle management relies completely on handling of *public* information.

Flexibility. Virtually no restrictions w.r.t. support for

- mix-and-match of devices from different vendors;
- changes to network topology (merging or partitioning of networks, device replacement or addition, addition of pre-assembled subsystem);
- changes to device roles (e.g., smooth hand-over of system manager, security manager roles, via ‘soft reboot’);
- back-up and failure recovery (since management fully relies on *public* information).

Features and Benefits (2)

Minimizes trust dependencies. Since secret information is never disclosed,

- Greatly reduced reliance on trustworthy personnel;
- Virtually no training requirements for operational personnel;
- Virtual removal of trust dependencies between different entities in value chain (whether OEM, vendor, system integrator, installer, or user).
- Ease of security auditability.

Support for flexible deployment and business models. Due to the fact that no secret information is ever disclosed during a device's or network's lifecycle, network topology changes or device role changes present a 'clean' logical separation between state prior to and after such an event (thus, allowing subscription-based services, outsourced management, re-contracting, etc.).

Enforcement of standards compliance. Enforced by only issuing a certificate to devices from vendors that passed conformance testing.

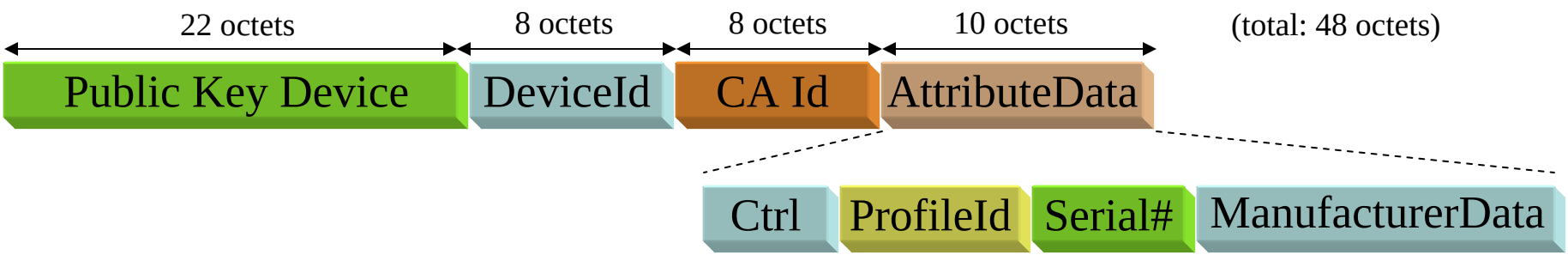
No reliance on configuration tools and out-of-band configuration steps. A configuration tool may be used, but is not strictly necessary for trust enforcement.

Cost

Conventional wisdom: Symmetric-key cryptographic functionality, let alone public-key cryptographic functionality, are expensive to implement with sensor networks.
Status anno 2008: conventional wisdom challenged for all but most mundane devices.

Examples: Bluetooth v2.1, ZigBee Smart Metering, RFID e-Passport.

- ZigBee Smart Energy Profile Certificate Structure:



- Low-energy hardware implementations:

	Smart Sensors[1]	RFID[2]
clock frequency	2 MHz	10 MHz
#gates	~10 kgates	~100 kgates
CMOS process	130nm	250nm
Energy exp.:	~ 250 μJ	< 100 μJ
Computation	signature verify	point multiple

Less than energy expenditure single IEEE 802.15.4 frame!

Sources:
[1]Certicom-internal
[2]SAC 2008 conference

The bigger picture…

Adhoc sensor networks

- Security models and trust management
- Semi-automatic lifecycle management
- Configuration and installation
- Low implementation cost
  - Protocols: re-use building blocks, parallelism flows, etc.
  - Keying material: key identification, key usage, key size

Adhoc networks
- No centralized management
- Promiscuous behavior
- Unreliability

Sensor networks
- Low energy consumption
- Low manufacturing cost

Security constraints
- Decentralized key management
- Flexible configuration and trust model
- Low impact key compromise
- Automatic lifecycle management
- Low communication overhead
- Low implementation cost

Certicom Corp.
5520 Explorer Drive, 4th floor
Mississauga, ON
Canada L4W 5L1
Contact: René Struik
E-mail: rstruik@certicom.com

Certicom Shanghai office
Suite 1019, Oasis Tower
555 Zhong Shan Road West
Shanghai 200051, China P.R.
Contact: James Zhou
E-mail: jzhou@certicom.com

Security Concepts – A Short Introduction

Basic Security Services

Authenticity

Evidence as to the true source of information or the true identity of entities:

- *Message authentication*.

Evidence regarding the true source of information:

(1) No undetectable modifications, deletions, and injections of messages by external parties (data integrity);

(2) No confusion about who originated the message (source authenticity).

- *Entity authentication*.

Evidence regarding the true identity of entities and on their active involvement:

(1) No confusion about whom an entity is really communicating with (authenticity);

(2) Proof that entity is actively participating in communications (i.e., is 'alive').

Secrecy

Prevention of external parties from learning the contents of information exchanges:

(1) Logical separation of information between parties that may have access to info and those that do not.

(2) No confusion about whom those privileged parties are (authenticity).

Cryptographic Building Blocks - Authentication (1)

Message authentication

Evidence regarding the true source of information:

(1) No undetectable modifications, deletions, and injections of messages by external parties (data integrity);

(2) No confusion about who originated the message (source authenticity).

<u>Realizations:</u>

- *Keyed hash function (or Hash Message Authentication Code (HMAC)).*

Mapping of arbitrary messages (of any length) to *compact representative image* hereof, using a secret key.

(1) Data integrity, since difficult to find distinct messages with same MAC value.

(2) Source authentication, since only parties that share the secret key can produce MAC-value (assuming there is no confusion about who has access to this key).

- *Un-keyed hash function.*

Mapping of arbitrary messages (of any length) to *compact representative image* hereof (digital fingerprint, or message digest), without secret key.

(1) Data integrity, since difficult to find distinct messages with same hash value.

(2) Source authentication, *only if* message digest is communicated authentically.

Cryptographic Building Blocks - Authentication (2)

Entity authentication

Evidence regarding the true identity of entities and on their active involvement:

(1) No confusion about whom an entity is really communicating with (authenticity);

(2) Proof that entity is actively participating in communications (i.e., is 'alive').

<u>Realizations:</u>

- *Entity authentication protocol (challenge response protocol).*

(1) Source authentication, since only parties that share the secret key can produce proper responses to unpredictable challenges (assuming there is no confusion about who has access to this key).

(2) Aliveness, since challenge messages are unpredictable and never repeated. (Hence, replaying previously recorded protocol messages does not leak info.)

Cryptographic Building Blocks - Secrecy

Secrecy

Prevention of external parties from learning the contents of information exchanges:

(1) Logical separation of messages between parties that may have access to info and those that do not.

(2) No confusion about whom those privileged parties are (authenticity).

<u>Realizations:</u>

- *Symmetric key cryptography.*

Logical separation of information, since only parties that share the secret key can learn the contents hereof (assuming there is no confusion about who has access to this key). Note that the symmetric key is used both for encryption and for decryption.

- *Public key cryptography.*

Logical separation of information, since only parties that have access to the private decryption key can learn the content of encrypted messages (assuming there is no confusion about who has access to this private key). Note that any party may obtain access to the public encryption key, since it does not reveal the decryption key.

Cryptographic building blocks – Authenticity and Secrecy (1)

Symmetric key cryptography

Security services based on exchange of secret and authentic keys:

(1) Logical separation of messages, by exchanging secret keys between privileged parties only;

(2) Authenticity of privileged parties by checking credentials of each party, by non-cryptographic means (certified mail, courier, face-to-face, etc.)

Public key cryptography

Security services based on exchange of authentic public keys:

(1) Logical separation of messages, by restricting access to each private key to the privileged party only (in practice, there is only 1 privileged entity);

(2) Authenticity of privileged parties, by checking credentials of each party by non-cryptographic means and (if successful) by subsequently binding the public key to this party (*certification of public keys)*.

Certification is done by a so-called Trusted Third Party, who vouches for the authenticity of the binding between an entity and its public key.

Cryptographic building blocks – Authenticity and Secrecy (2)

Public key cryptography (cont'd)

Certification of public keys depends on appropriately checking the credentials of a party and constitutes the following:

(1) Check, by cryptographic means, that the entity A that claims to have access to the public key P_A, has access to the corresponding private key S_A;

(2) Check, by non-cryptographic means, the claimed identity Id_A of A.

Certification is done by a so-called trusted third party:

- *Digital certificates (cryptographic binding).*

(1) Authenticity of binding, via signature over the pair (Id_A,P_A) by trusted party;

(2) Verification of authenticity of public keys *by any party*, by verifying signature of trusted party in the digital certificate (assuming the authentic storage of trusted party's signature verification string on each verifying device);

(3) Unrestricted transfer of certificates possible (hence, off-line certification possible).

- *Manual 'certificates'(non-cryptographic: pushing button, low power mode, etc.).*

(1) No cryptographic verification of the authenticity of public keys possible;

(2) No transfer of certificates possible (hence, on-line 'certification' only).