certicom

A Subsidiary
of Research In Motion Limited RIM

Certicom

Building Trust in Medical Devices

Agenda

- About Certicom
- Connectivity Trends & Security Implications
- How Cryptography Helps – A Primer
- Trust Anchors: Some Case Studies from Other Industries

About Certicom

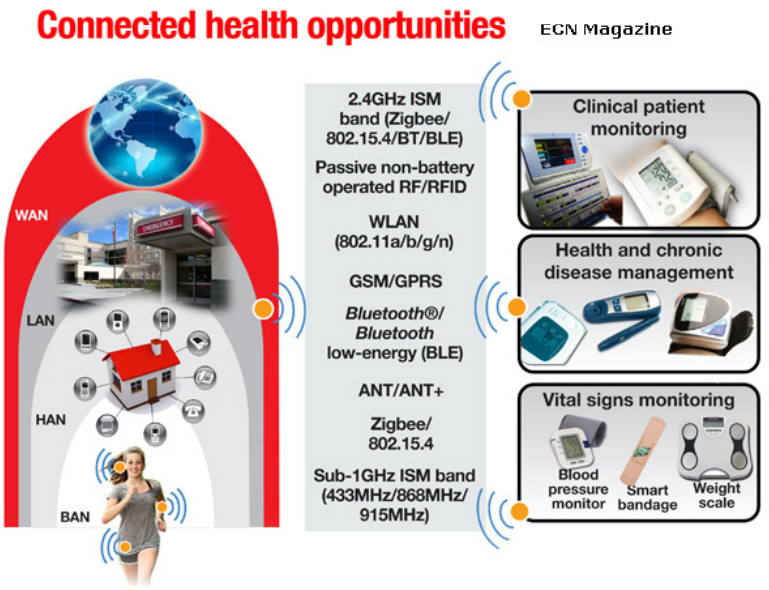
- Digital security/applied cryptography specialist
  - A subsidiary of Research In Motion since March 2009
  - Experts in Elliptic Curve Cryptography (ECC), highest security per bit of any known public key cryptosystem
- ~100 employees
  - HQ & principal engineering team in Mississauga, ON, Canada
  - Hardware (silicon) design team in Scotts Valley USA
- Develop and license software products and intellectual property for (embedded) security solutions
- Mission: promote adoption of Elliptic Curve Cryptography, help secure the RIM BlackBerry platform & ecosystem

certicom

Healthcare Connectivity Trends

- Clinical patient monitoring
  e.g. ECG, pulse oximeter, thermometer, etc.
  - Automated logging / record mgmt
  - Tracking portable assets
  - Eliminating wiring to reduce costs
- Chronic disease management
  e.g. Glucose & blood pressure monitors
  - Automated record keeping, growing mobile connectivity
- Wellness monitoring
  e.g. fitness equipment, weight scales, heart rate monitor
  - Support health consciousness
  - Independent living
- Pervasive wireless technology
  - WAN: mobile / WiMAX
  - HAN: Bluetooth, ZigBee, 802.15.4 & others
  - Body Area Networks ??

- Increasingly mobile
  - Connected, powerful, energy efficient
- Increasingly integrated
  - Continua IEEE 110173 medical device communications standards
- Moving to the cloud?
  - Electronic health records



http://www.ecnmag.com/Articles/2010/10/App-Solutions/Accelerating-mobile-health-systems-through-technology/

certicom

Security Implications

Increasingly connected, increasingly vulnerable

- More devices to manage, larger attack surface
  - One compromised device might be used to attack on others on the network
- Wireless devices a nice target – no physical access required
  - Easier to eavesdrop or inject messages
  - Encryption is common but robustness is often poor (e.g. shared keys)
- Wired network devices are also susceptible to attack
  - Often use weak passwords or default security settings

Potential Impact:

- A compromised device could cause injury/death
  - Medication errors from a tampered IV pump drug library; remote attack on a pacemaker
- Vulnerabilities could lead to breach of personal health info
  - Legal, administrative and reputational repercussions

How Cryptography Helps

- Confidentiality
  - Only authorized sender and receiver can read message
  - Need to consider data at rest as well as in transit
- Integrity of data and control signals
  - Detect and reject any corrupted messages or tampered files
  - Prevent “hidden” attacks (e.g. replay messages)
- Authentication/authorization of sensitive commands, software
  - Authenticate firmware updates, especially over-the-air updates
    - Is this the correct firmware? Has it been tampered?
    - Has the firmware received regulatory approval?
      - A case for code signing…
- Non-repudiation
  - Traceability / attribution to responsible parties

Crypto Building Blocks – A Quick Primer

- Hashing & Message Authentication
- Symmetric encryption
- Public key cryptography
- Key Management
- Digital Signatures
- Public Key Infrastructure

- And some discussion of the many pitfalls

Cryptographic Hash Functions

Cryptographic “checksum”; aka message digest, fingerprint

Process a large message quickly to create a short string that depends on all of the message. (Changing any bit should change the result.)

Designed to be “secure” one-way functions

- Given the result, difficult to figure out anything about the original message
  - Difficult to choose a hash value and find a message with that value
  - Difficult to find two messages with the same hash value

Examples:

SHA-1, SHA-256, SHA-384, ~~MD5~~

Keyed Hash Application

Message Authentication Code (MAC) prevents undetected message alteration; block cipher based MAC requires secret key distribution

Network Data Integrity:

File Integrity:

$$MAC = H_{bobpwd}(File)$$

Bob's file cannot be altered without his password

certicom

MAC Issues

Cannot prove message was sent by sender or receiver

Cannot be verified by a third party without revealing the secret key

Cannot provide non-repudiation

Symmetric Encryption

Widely used for data encryption (data in motion & at rest)

Features:

- Same key used to encrypt and decrypt data
  - Also called secret key algorithms
- Performance: relatively efficient to implement in SW, faster in HW
  - E.g. AES-128 @ 180 cycles/byte on AVR8, 40 cycles/byte on ARM, 28 cycles/byte on a P4, 3.5 cycles/byte on i7, < 1 cycle/byte on a GPU

Two general types:

- Block - operates on data blocks (e.g. 8 byte block)
  - DES, TDES, AES
- Stream - operates on single data units (e.g. 1 bit)
  - A5/1, A5/2, RC4 …

Drawback:

- Key must be shared between parties/devices

The Key Management Conundrum

Key management covers the generation, storage, transport, usage & destruction of keying material (especially shared keys)

If Alice wants to send an AES encrypted message to Bob, how does Bob get the proper key to decrypt the message?

For a network of 100 VPN encryptors, if each pair of encryptors uses a unique key to secure their corresponding transmissions, ~5000 key exchanges required!

- How to solve?

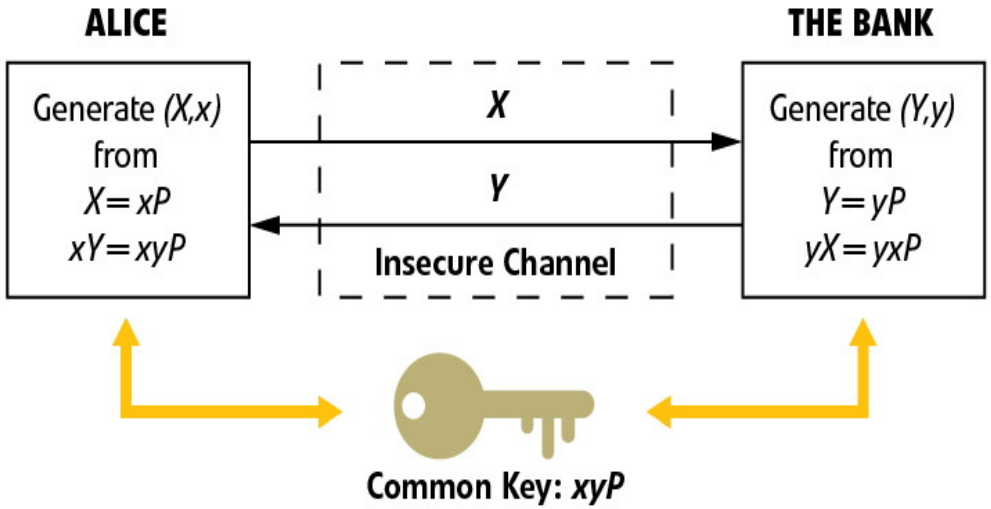
Public Key Cryptography

PK are asymmetric cryptosystems

- Encryption and decryption keys are different; conjoined
- A “public key” is to be shared
- A related “private key” which is kept secret

- Public key cryptosystems facilitate both key exchange / key management– help solve the key distribution problem…
  - Encrypt symmetric keys meant for distribution with the target recipient’s public key & the recipient decrypts with their private key; only the intended recipient can read the message
  - Or do “key agreement”
- .. and facilitate digital signatures (with non-repudiation)

(EC Diffie-Hellman) Key Agreement

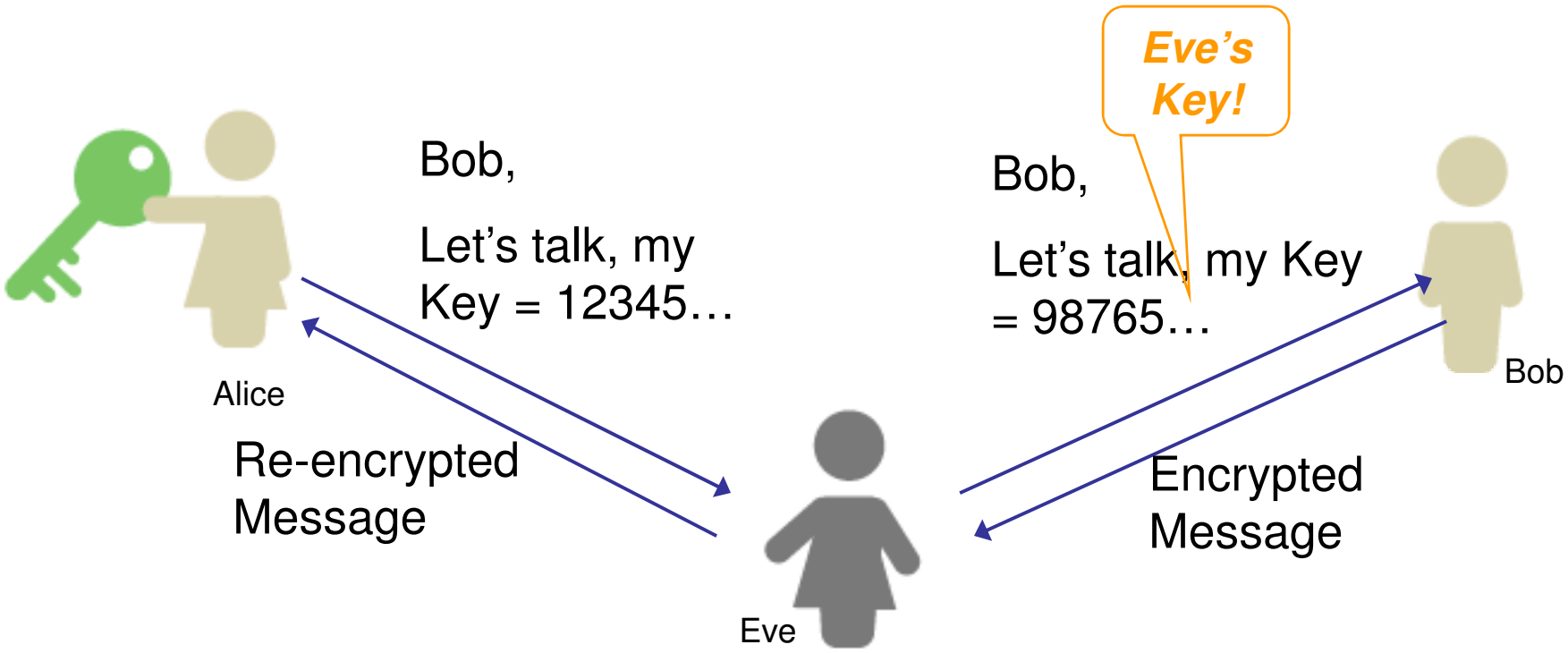


- Alice generates public and private key
  - Sends public key (X) to bank
- Alice combines her private key and bank's public key to form common key

- Bank generates public and private key
  - Bank's public key (Y) sent to Alice
- Bank combines its private key and Alice's public key to arrive at same common key

certicom

Vulnerability: Man-in-the-Middle Attack



MUST authenticate public keys to prevent attacks!

Decrypt Bob's Message

Read it

Re-encrypt with Alice's key.

certicom

Public Key Certificates

Bind a public key pair to identifiable attributes to form a digital identity; signed by a Certification Authority

- Owner and Key
- Like a driver's license or passport
- Allows public keys to be authenticated by verifying certificate signatures (provided you trust the CA and check for revocation)

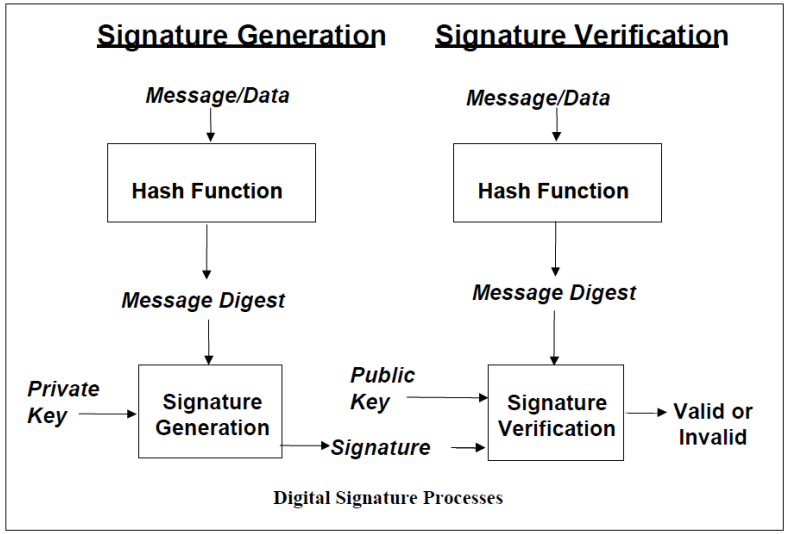
Public key certificates support:

- Authentication
- Authorization
  - Access Control
  - Permission
- Non-repudiation
- Can also facilitate encryption (e.g. S/MIME)

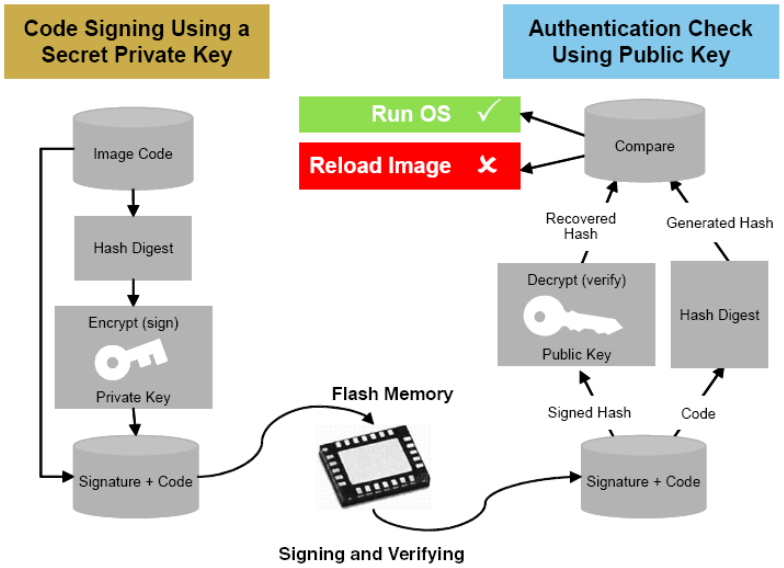
Digital Signature Applications

Alice's signature is a function of both her message and her secret encryption key

Bob can verify the integrity of the message and the validity of Alice's signature by using Alice's public key



Common use: secure boot



Digital signature schemes:

RSA, El Gamal, DSA, ECDSA, ECPV

certicom

Aligning Symmetric & PK Crypto Strengths*

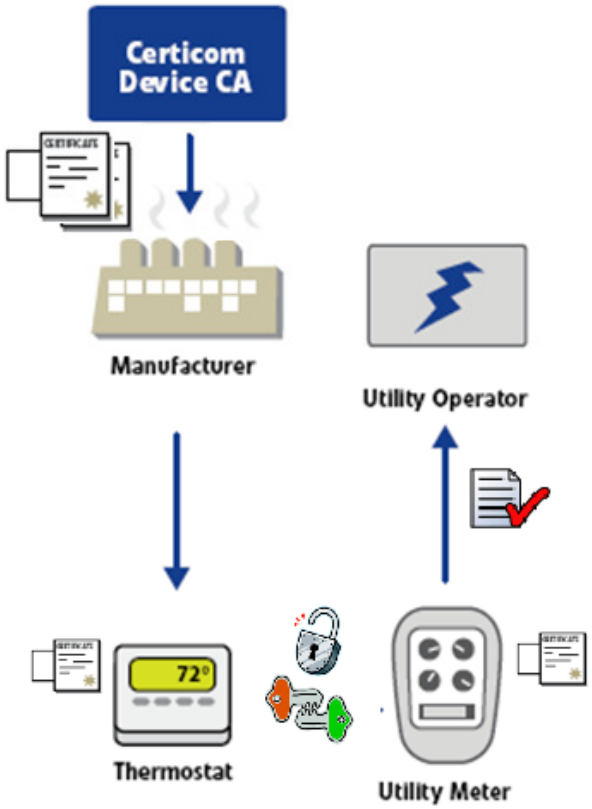
Cryptographic Strength	Symmetric Algorithm	Hash Algorithm	Elliptic Curve Asymmetric Algorithms	RSA/DSA/DH Asymmetric Algorithms	Expected Lifetime Expiry
56 bits	DES				Expired
60 bits		(MD5) **	111 bits	512 bits	Expired
80 bits	3DES (2 key)	SHA-1	160 bits	1024 bits	2010-2013
112 bits	3DES (3 key)	SHA-224	224 bits	2048 bits	2030
128 bits * **	AES-128	SHA-256	256 bits	3072 bits	+30 years
192 bits	AES-192	SHA-384	384 bits	7680 bits	+30 years
256 bits * **	AES-256	SHA-512	512 bits	15360 bits	+30 years

* Recommended by NIST
** MD5 has never been recommended by NIST and is now broken
*** 128-bit is standard commercial strength
and 256-bit is the new US gov Suite B standard (recommended with 384-bit ECC unless for classified security levels)

certicom

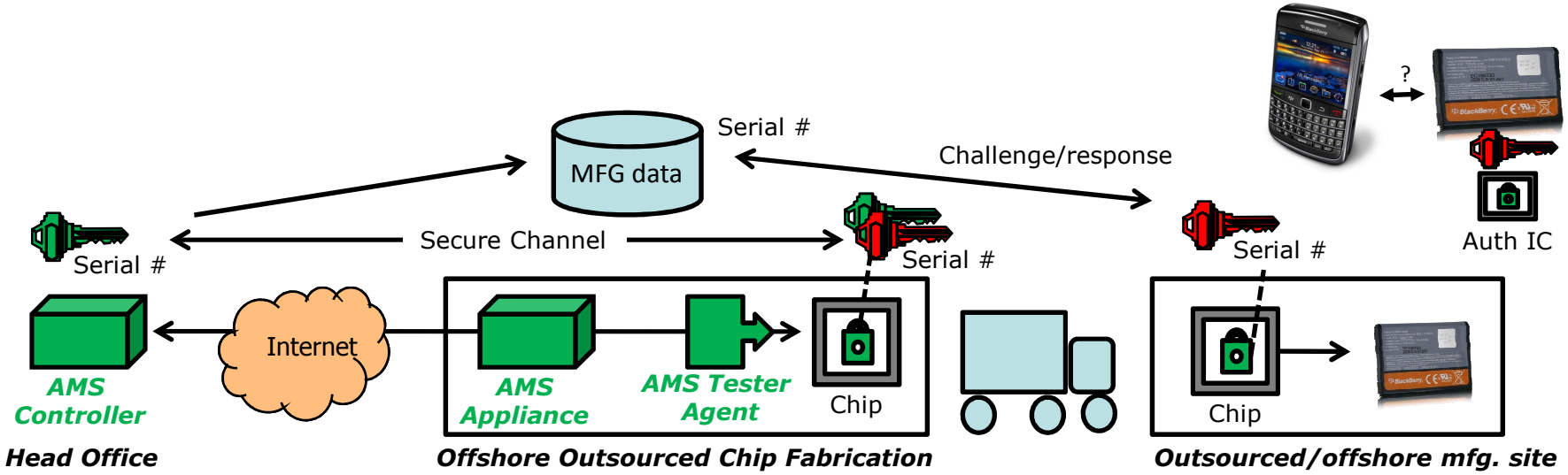
Case Study: Certicom CA
For ZigBee Smart Energy 1.x Devices

- Certicom Certification Authority issues ECC-based device certificates
  - ONLY for ZigBee Smart Energy certified devices
  - Binds key pairs to
    - Vendor ID
    - Unique MAC address
- Enhances ZigBee security
  - Certificate and keys are used to perform authenticated key agreement
- Enables out-of-box interoperability for branded ZigBee Smart Energy devices
  - Enhances provisioning and installation security
  - Helps ensure a robust ecosystem



certicom

Building Your Trust Anchors
Strong Security+Device Provenance -> Device



- Keying silicon and digitally signing software enables high assurance device manufacturing in low-security locations
  - Prevents counterfeit processors/devices from entering manufacturing system or authenticating with the ecosystem
- Core security protects your brand

  & creates a base for rich, secure, integrated services

certicom

Summary

- Cryptography plays a critical role in developing high assurance connected medical devices (and all kinds of other systems)
  - Many techniques to help protect integrity, confidentiality of commands, data in motion, data at rest
  - Public key cryptography helps build a chain of trust for strong authentication & critical key management operations
  - Many pitfalls along the way – both in design & implementation
- Strongest security goes down to the chip level
  - Who do you trust?
- For medical systems, think long-term security requirements
  - Need to protect systems and data for the lifetime of the assets –
  - This could mean the lifetime of the patient! (*Hint: use ECC*)

certicom
securing innovation

protect your content, applications and devices

with government-approved security