TCIPG

TRUSTWORTHY CYBER INFRASTRUCTURE FOR THE POWER GRID | TCIPG.ORG

KEY MANAGEMENT

OCTOBER 24, 2014

KARTIK PALANI and BEN UJCICH

UNIVERSITY OF ILLINOIS

UNIVERSITY OF ILLINOIS | DARTMOUTH COLLEGE | UC DAVIS | WASHINGTON STATE UNIVERSITY

FUNDING SUPPORT PROVIDED BY DOE-OE AND DHS S&T

What are Keys?

- A **key** is the variable used as part of encryption and decryption algorithms in cryptographic systems
- During **encryption**, a key transforms **plaintext** into **ciphertext**
- During **decryption**, a key performs the reverse operation and converts ciphertext into plaintext
- Motive: large-scale systems need a way to manage keys

Key Management: Generation

- How can many keys be generated given that they may be needed for different purposes?
- Varying levels of security needed depending on application
- Varying levels of trustworthiness when generating keys:
  - Key ceremony: generation of root keys for a chain of trust requiring specific procedures to ensure integrity (in a certificate authority)

Types of Key Algorithms

Symmetric

- One key used for both encryption and decryption
- Concept of **shared secret**
- Shorter lengths (e.g. 128 bits)
- Algorithms
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)
  - Triple DES (3DES)

Asymmetric

- One key used to encrypt (**public key**) and one key used to decrypt (**private key**)
- Public key infrastructure (PKI)
- Longer lengths (e.g. 2048 or 3072 bits)
- Algorithms
  - Diffie-Hellman Key Exchange
  - RSA

Diffie-Hellman Key Exchange (Video)

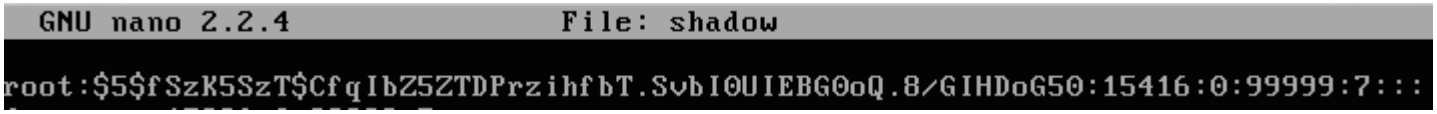
Key Management: Storage

- How should keys be stored such that only those with the correct authority be able to access them?
- Best practice: “Ensure that any secret key is protected from unauthorized access”
  - Separate where keys and data are stored
  - Encrypt keys themselves (passphrase)
  - Use key vaults
  - Store in trusted platform modules and/or hardware security modules

Source: https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet

Hashing

- Used for: authentication, non-repudiation, integrity
- “Digital Fingerprint” of an input: unique and irreversible
- MD5 and SHA: commonly used hashing algorithms.
- Applications:
  - Password security
  - File/memory integrity
  - Message Authentication



Digital Signatures

- Technique to guarantee authenticity
- Steps involved:
  - Generate secure hash
  - Encrypt hash with private key
  - Hash + Message = Signature
  - Receiver calculates hash of message
  - Decrypt signature using public key
  - Compare decrypted signature to hash
- Authenticity: Only I can generate my signature
- Non-repudiation

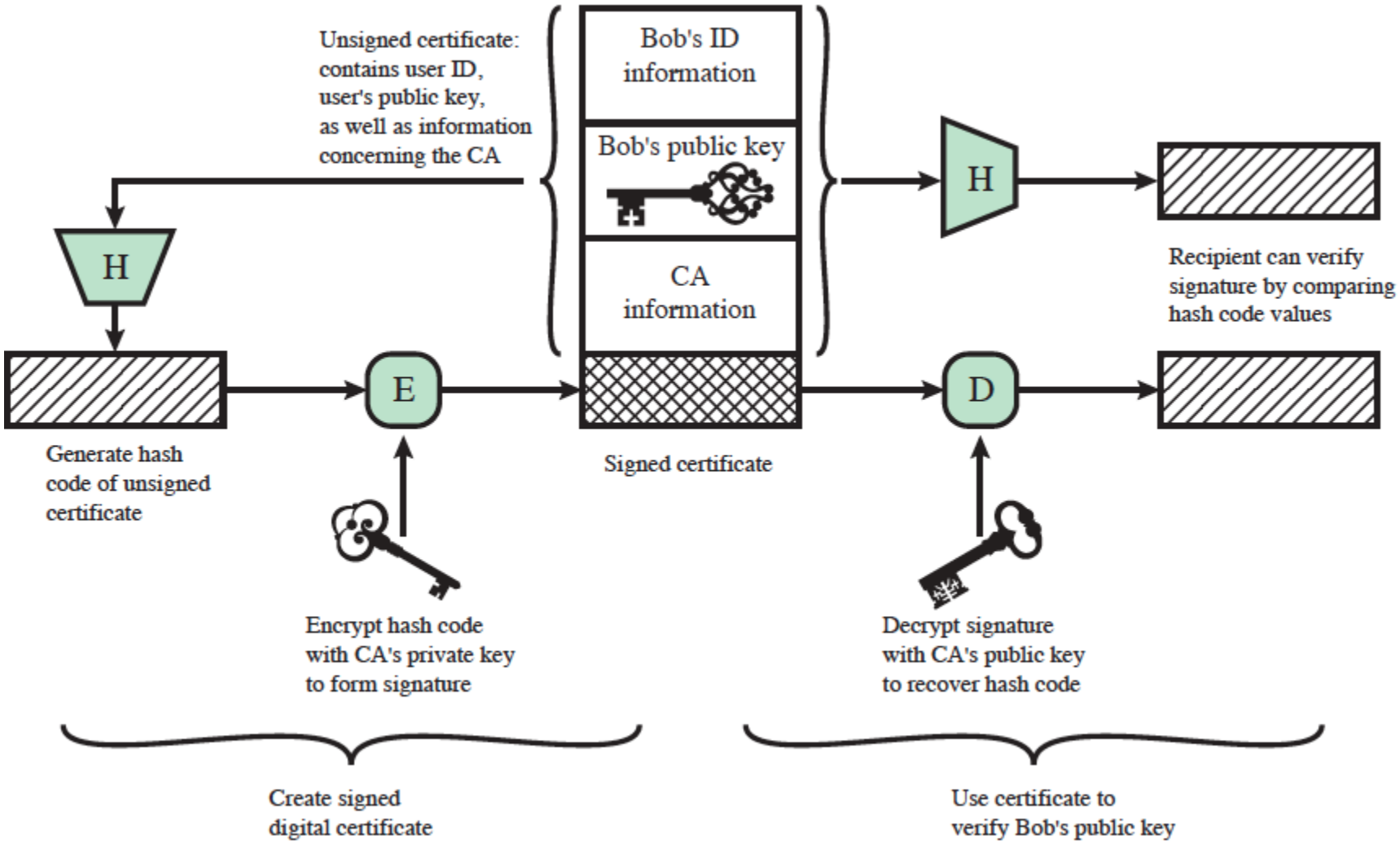
Key Management: Exchange

- How can keys be efficiently exchanged with concern given to scalability and trustworthiness?
- **Direct trust**: trust since origin is known
- **Hierarchical trust**: trust CAs and root CAs
- **Web of trust** (distributed): trust based on others whom you trust; mix of above two
- Public key server
  - Access others' public keys
  - Difficult to remove old keys
  - Example: PGP Global Directory

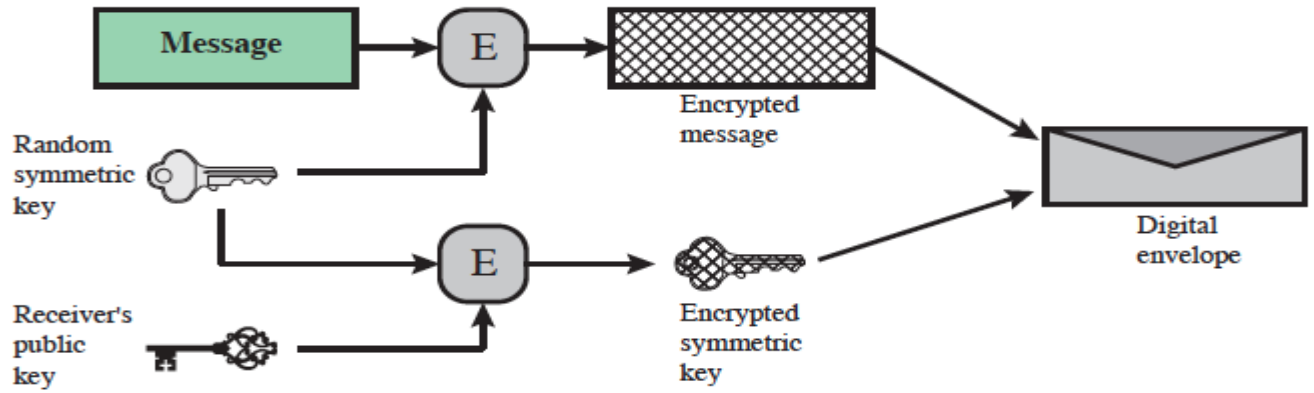
Public Key Certificates

- Problem: signatures can be forged
  - Anyone can claim to be me
  - Distribute their public key
- Certificate Authority
  - Trusted third party
  - DigiCert, Verisign
- X.509
  - Certificate standard for the internet
  - Establishes a chain of trust

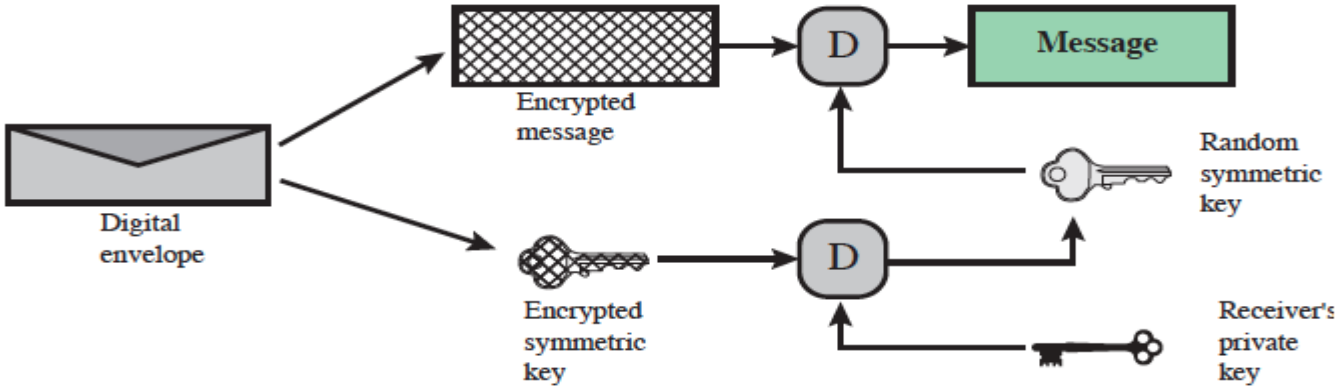
Public Key Certificates



Digital Envelope



(a) Creation of a digital envelope



(b) Opening a digital envelope

Key Management: Replacement

- What should the length of use be for a key?
- Is the key meant to last for one session, one week, one year, many years, etc.?
- Reasons for replacement:
  - Key has been revoked
  - Key has expired
  - Key has been compromised
    - Detected
    - Undetected
- Rekey encrypted data with new keys

Source: https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet

Revocation of Certificates

- CRL: Certificate Revocation List
  - Published by CAs
  - Every certificate must be checked against the CAs CRL.

- Problem?
  - CRLs have gotten way too big
  - Overhead of checking is too high
  - Current solution: your browser does not check CRLs!!
  - One of the unsolved problems of security

SSL/TLS

- Provides secure connection over the internet
- Protects the application layer
- HTTPS: HTTP protected by TLS
- Session:
  - Create association between client and server
  - Established using a handshake
  - Defines the set of cryptographic parameters
- TLS Heartbeat Protocol
  - Heartbleed!

Cipher Suite

- Named combination of cryptographic primitives:
  - key exchange algorithm (RSA/Diffie Hellman)
  - authentication algorithm (RSA, ECDSA)
  - bulk encryption algorithm (eg: AES/3DES)
  - message authentication (eg: HMAC-MD5)

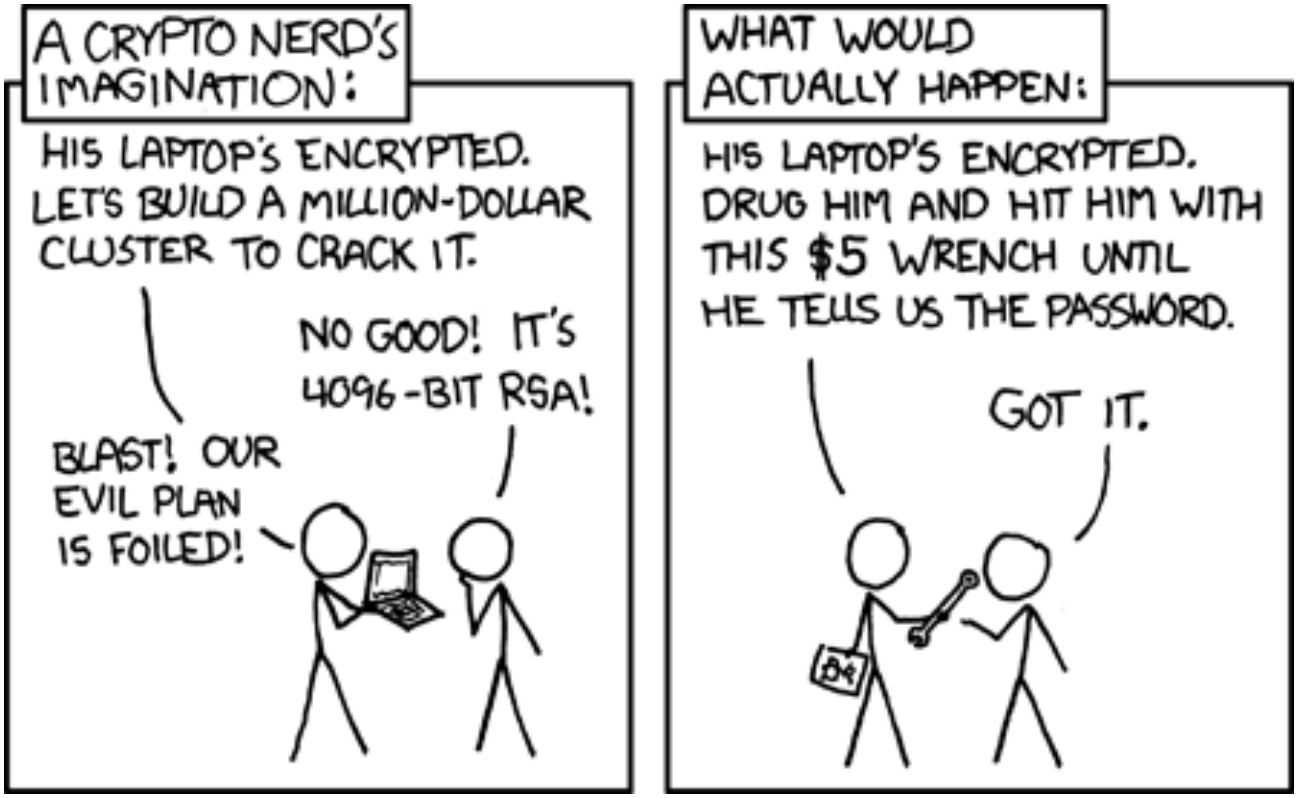
- Examples:
  - RSA-RSA-AES-SHA
  - DH-RSA-DES-SHA

Key Management in the Internet of Things

- Challenges:
  - Large number of devices
  - Limited processor and memory resources
  - Scaling difficulty with manual configuration
  - SSH as an example: What privileges can SSH access give you on a device?
- Specific challenges to the smart grid:
  - Sensitive information about devices and their electricity use

Source: http://www.iab.org/wp-content/IAB-uploads/2011/03/Turner.pdf

Thanks!


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.