Lecture Notes on Jan/16/2023

Chapter 1: Computer Security Introduction and Review (Part 03)

CYENG 351: Embedded Secure Networking

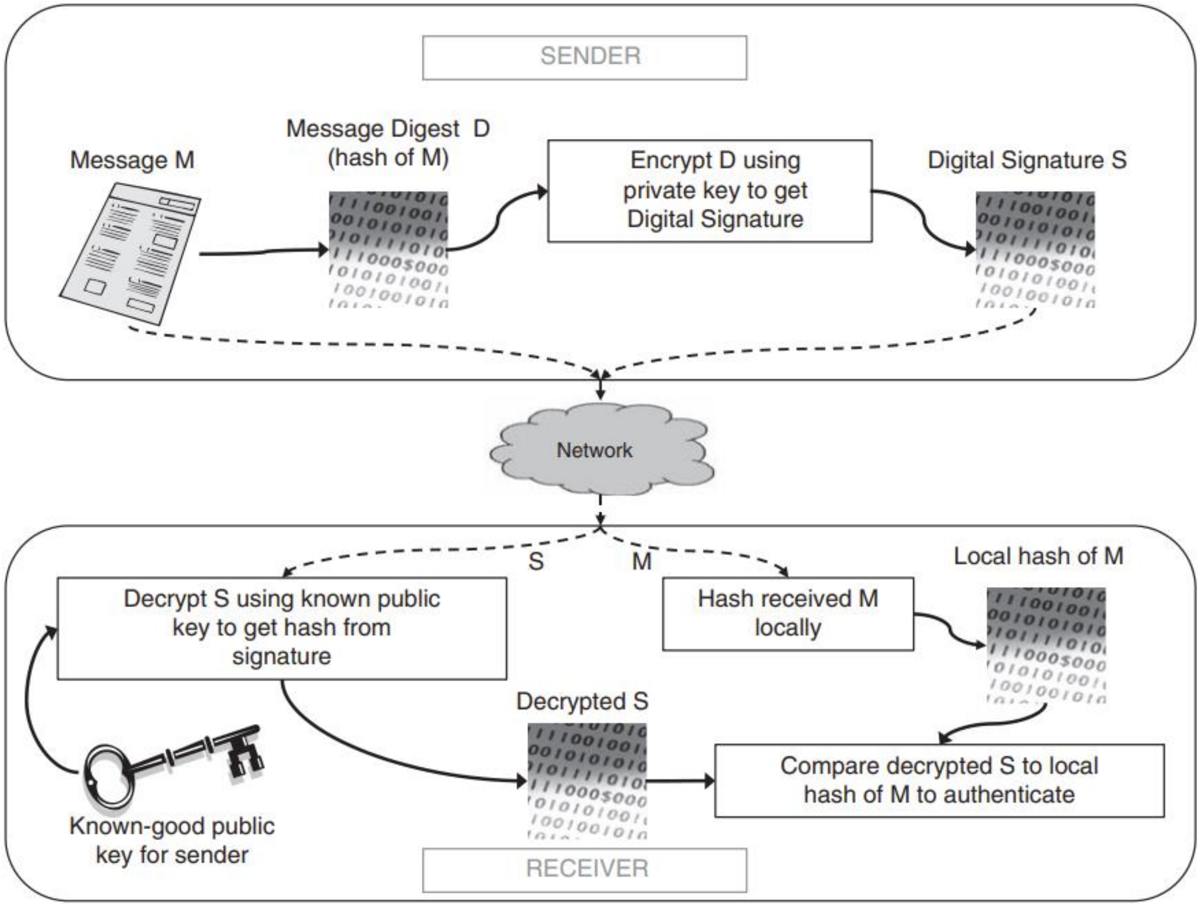
Instructor: Dr. Shayan (Sean) Taheri
Gannon University (GU)

Data Integrity and Authentication

- To enhance cryptography in a practical system, and sometimes even replace it, we can use mechanisms and methods that protect data integrity and authenticate entities.
- **Certain mechanisms** exist for Data Integrity **and/or** Authenticate Entities.
  - Cryptographic Hash.
  - Message Digest Algorithm.
  - Reversing Public-Key Algorithms.
  - Digital Signatures and Certificates.
  - Public-Key Infrastructure (PKI).
- **Message Digest Algorithm**
  - A **hash function**, enter some arbitrary data of arbitrary size, and the hash algorithm spits out a fixed-size number that is relatively unique for the input given.
  - A perfect hash can only be created if we can restrict the input to known values.
  - What makes a hash function into a message digest is a level of guarantee that if two input datum are different (even by a single bit), then there is a predictably small possibility of a hash collision (those two messages generating the same hash).
  - There is a very small probability of two messages generating the same hash value, and the chances of those two messages both containing legitimate data is even smaller.
  - No hash matching means the message has been altered, either **accidentally** (some transmission error where data was lost or corrupted), or **intentionally** (by a malicious attacker).
  - Most commonly used algorithms: MD5 and SHA-1

Data Integrity and Authentication (Cont.)

- **Message Digest Algorithm**
  - Example Usage in Embedded Network Security → **Transport Layer Security (TLS) Protocol** → Due to having **Hash-based Message Authentication Code (HMAC)**.
  - Hash algorithms are useful for protecting the integrity of data as well as authentication.
  - The existing algorithms require precautions due to possible weaknesses.
  - Hashing has usage in **Reverse Public-Key Cryptography**.


SENDER
Message M
Message Digest D (hash of M)
Encrypt D using private key to get Digital Signature
Digital Signature S
Network
S
M
Local hash of M
Decrypt S using known public key to get hash from signature
Hash received M locally
Decrypted S
Compare decrypted S to local hash of M to authenticate
Known-good public key for sender
RECEIVER

Usage of Hashing in Public-Key Authentication with Reverse Encryption-Decryption

Digital Signatures

- It is not efficient if every message had to be encrypted using the public-key algorithm.
- **Solution:** **Digital Signature** → A hash of the data to be sent (using one of the message digest algorithms) encrypted using the public-key authentication method.
- By encrypting only **the fixed-size message hash**, we remove the inefficiency of the public-key algorithm.
- Able to efficiently authenticate any arbitrary amount of data.
- The public-key must be trusted and the private key must always remain private.
- The basis of trust for most of the Internet and e-commerce.

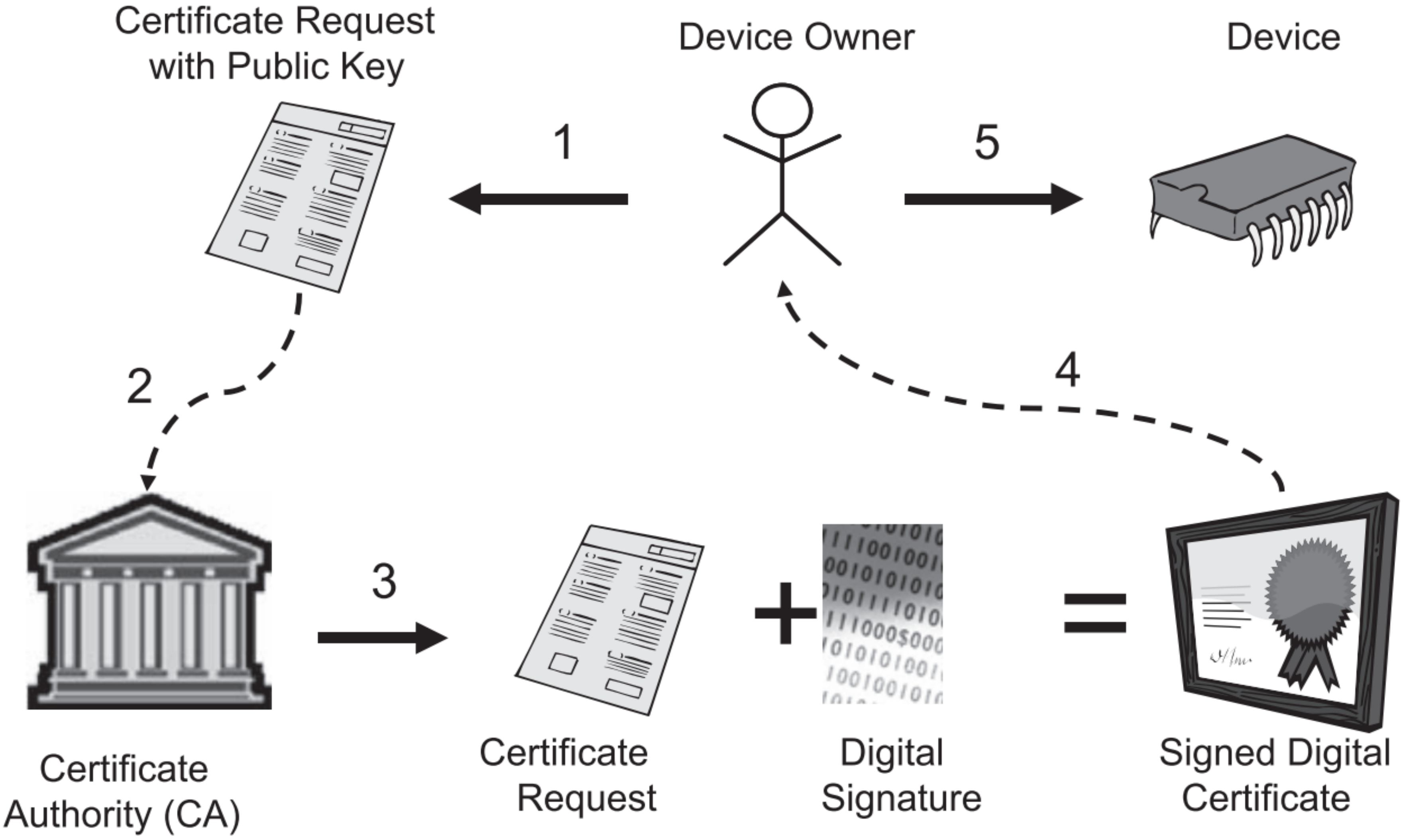
Digital Certificates

- How to provide security for our applications? → The most common use of the digital signature for authentication is a part of a **Digital Certificate**.
- **Digital Certificate** consists of three primary sections:
  1. Information about the owner (such as real or company name, Internet address, and physical address).
  2. The owner's public-key.
  3. The digital signature of that data (including the public-key) created using the owner's private key.
- Digital certificates are typically encoded using a language called ASN.1 (Abstract Syntax Notation).
- Its subset, **Distinguished Encoding Rules (DER)** → Flexible, allowing for any number of extensions to the certificate format, which are created and used by some users of digital certificates to provide additional information.
- **Major Steps:** (1) Providing a digital certificate; (2) Parsing the certificate; (3) Decrypting the digital signature; (4) Comparing the decrypted signature hash with the information hash; (5) If the last step is successful, then checking **Common Name (CN)** that represents the Internet address (URL or IP address) of the sending application.
- Certification with Expiration → The certificate should be trusted at a certain time.
- Assumption: The public-key in the certificate can be trusted. → Possible Problem, why?

Public-Key Infrastructures

- The only guarantees that we can glean from a single certificate are:
  1. If the digital signature matches, then the owner of the private key created the certificate and it has not been tampered with or corrupted.
  2. If the address of the sender and the common name match up, then the certificate was probably created by the owner of that address (although the address can be spoofed, leading to other problems).
  3. If the current date falls within the valid date range, the certificate is not invalid.
- **Problem:** Authenticity of the information provided on the certificate cannot be guaranteed. → There are certain solutions that resolve the issue to some extent.
- **Solution: Public-Key Infrastructure (PKI)** → A known, trusted, third-party source can provide trust to anyone who needs it.
- A license has some security features built in, such as a common format, identifying information (including a photograph of the licensee), and anti-copy protection (some licenses have difficult-to-copy holograms built in). → The license was issued by a third party that is inherently trusted.
- The companies that provide the signing services are generally referred to as "**Certificate Authorities**" **(CA)**, and they have a private key that is used to sign customer certificates.

Public-Key Infrastructures (Cont.)

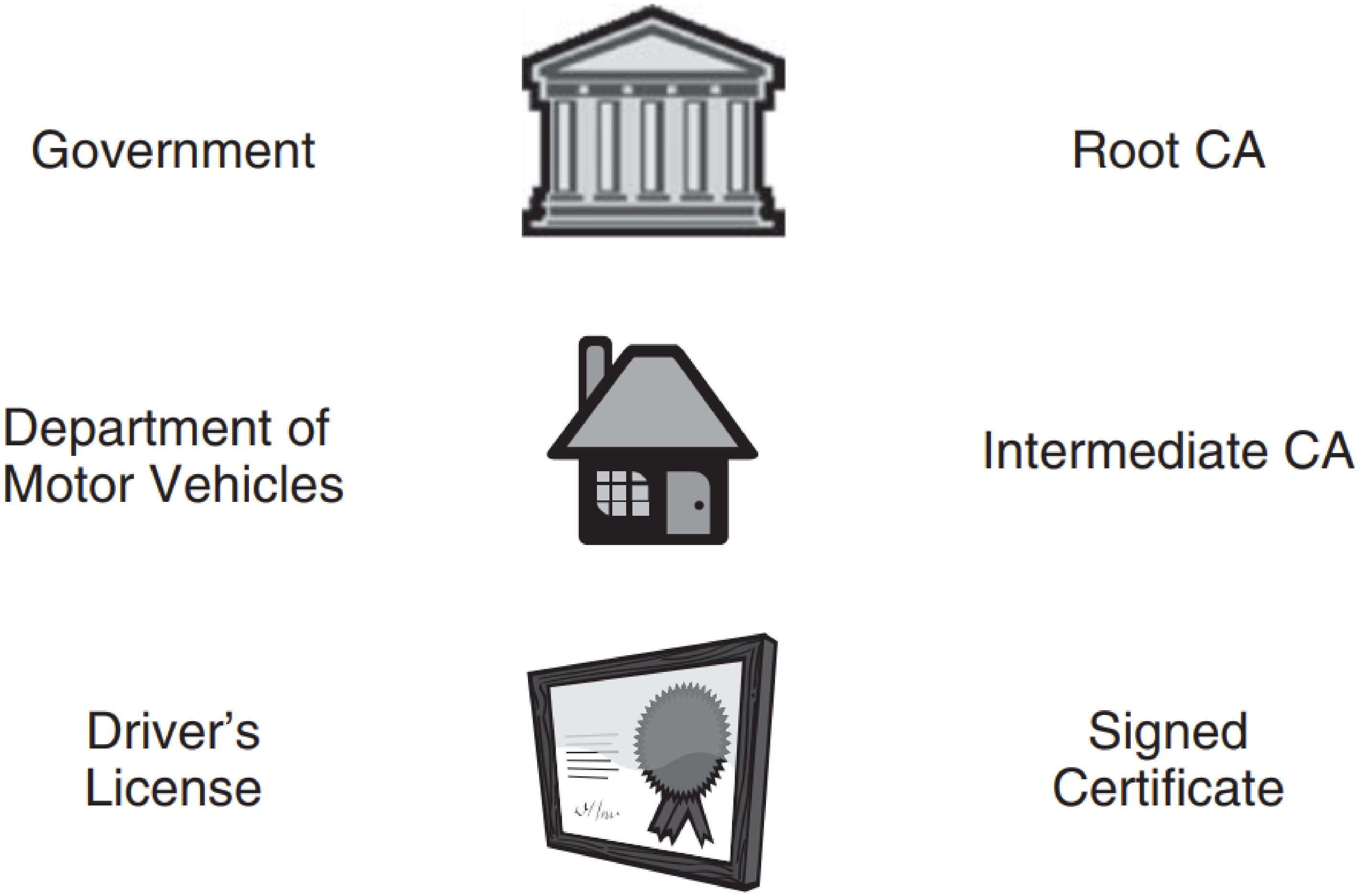

Certificate Request with Public Key
Device Owner
Device
1
5
2
4
3
Certificate Authority (CA)
Certificate Request
+
Digital Signature
=
Signed Digital Certificate

Digital Signature Signing Using a Certificate Authority

Public-Key Infrastructures (Cont.)

- This private key is associated with what is known as a “**Root**” certificate, which is the basis of trust in a PKI hierarchy.
- The basic idea is that a root certificate can be loaded from a known trusted site, providing a fairly high level of assurance that the certificate is valid.
- As long as a CA keeps the private key hidden, and provides the public-key in the root certificate to the public, the PKI infrastructure works.
- The trust in the CA is the most important link in a PKI chain.
- Certificate Authorities can also extend trust to other companies or entities that provide signing services under the umbrella of the root CA’s trust. → **Intermediate Certificate Authorities**.
- **Hierarchy – Certificate Chain:** Trust can be extended from the root CA to the intermediate CA, and finally to the end user.
- **PKI Drawback →** A single company or small group of companies controls the entirety of trust on the Internet, creating both a bottleneck and a single point of failure. → **Peer Networking**.

Public-Key Infrastructures (Cont.)



Government

Root CA

Department of Motor Vehicles

Intermediate CA

Driver's License

Signed Certificate

Department of Motor Vehicles Vs. Certificate Authority

Public-Key Infrastructures (Cont.)

Peer Networking

- A person establishes trust with someone they know, and then trusts documents sent by that person.
- Once a certain level of trust is achieved, then that trusted peer can vouch for the validity of documents provided by people or entities they know and trust, even if the recipient does not know the sender.
- By keeping the number of "**hops**" between <u>a sender and a recipient</u> short, the trust can be fairly easily maintained - without a central body providing that trust.
- Each person controls what is trusted, and there is no single point of failure.
- **Problem**
  - Establishing peer networks of trust takes time.
  - If the sender of a document is not connected to the recipient's network.

Assignment

- **Reading Assignment:**
  - Stapko, T., 2011. **Practical embedded security: building secure resource-constrained systems**. Elsevier.
    - ✓ "Chapter 1: Computer Security Introduction and Review", Pages 13-21.

Questions?