Security

Cryptography Essentials

- Authentication & digital certificates

Objectives

- Gain understanding of three main ingredients of most security protocols & products
  - Symmetric encryption (last week)
  - Public-key cryptography (last week)
  - **Cryptographic hash functions**
- Learn about (public) key management using
  - **Digital certificates**

Data Integrity

Data Integrity

- Integrity refers to assurance of non-alteration
- Many systems and components have checksums or cyclic redundancy checks that are designed to detect *accidental* errors, etc.
  - For example, a credit card number contains a digit that is used to verify the others
- But such schemes are not sufficient to prevent *deliberate* modifications

Cryptographic Hash Functions

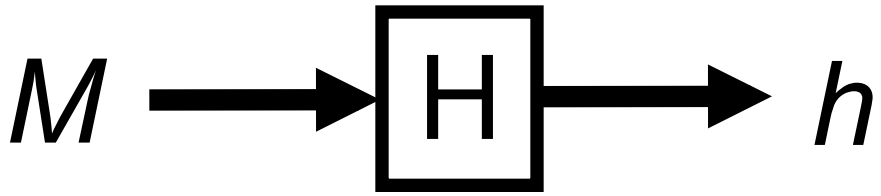
- Used to provide integrity of a message
- Purpose is to produce a fixed-size *hash-value:*

$$h = H(M)$$

where h is the hash value
H is the hash function
M is the message

- Any change in M, however small, should produce a different h-value

Cryptographic Hash Functions



Message (any size)

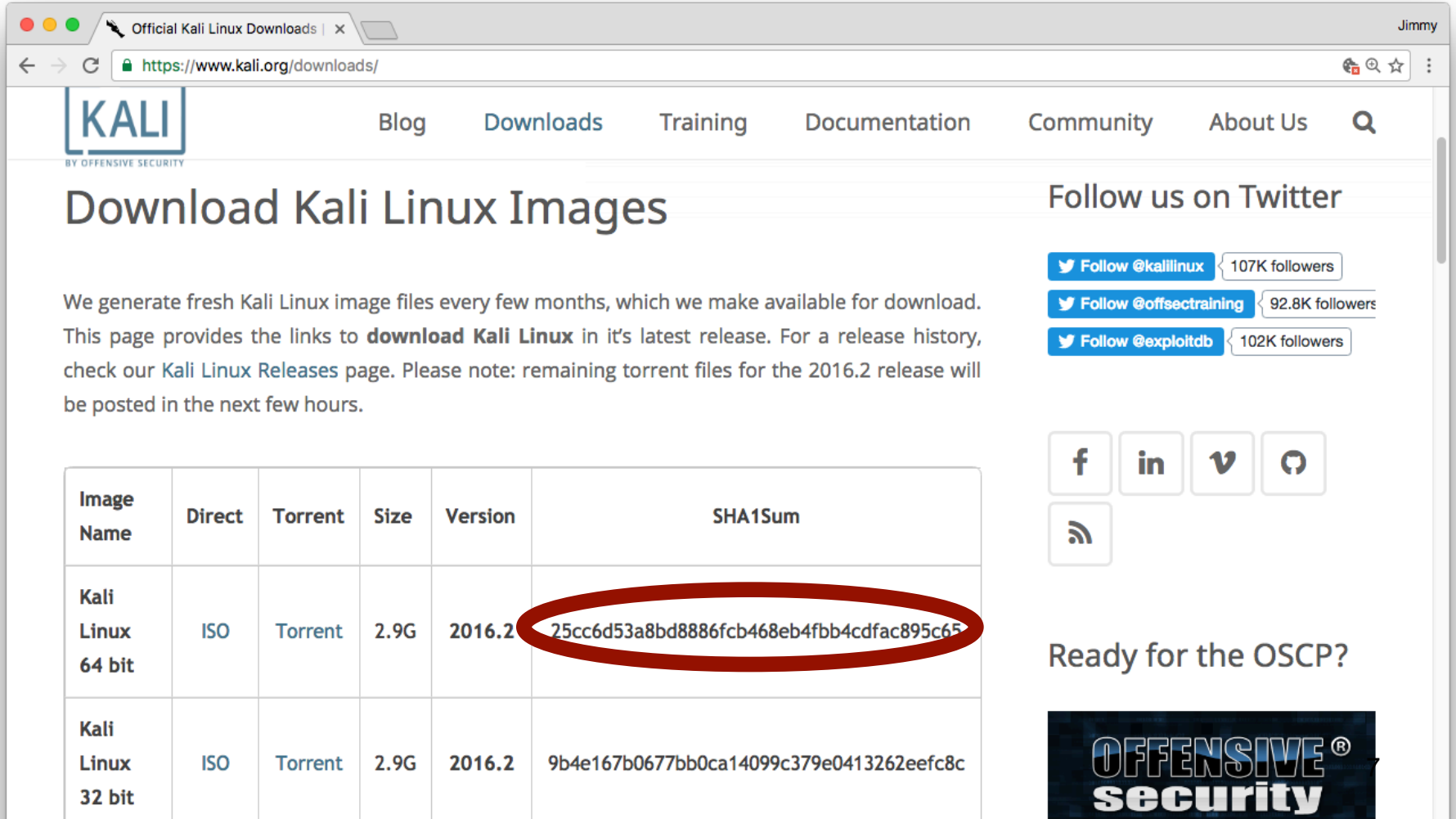
Hash Function

Hash Value (fixed-size; e.g. 160 bits)

- Note that a hash function is a many-to-one function. Potentially many messages can have the same hash, but finding these should be very difficult

Applications of Hash Functions

- As cryptographic checksum
  - e.g. to verify software downloads

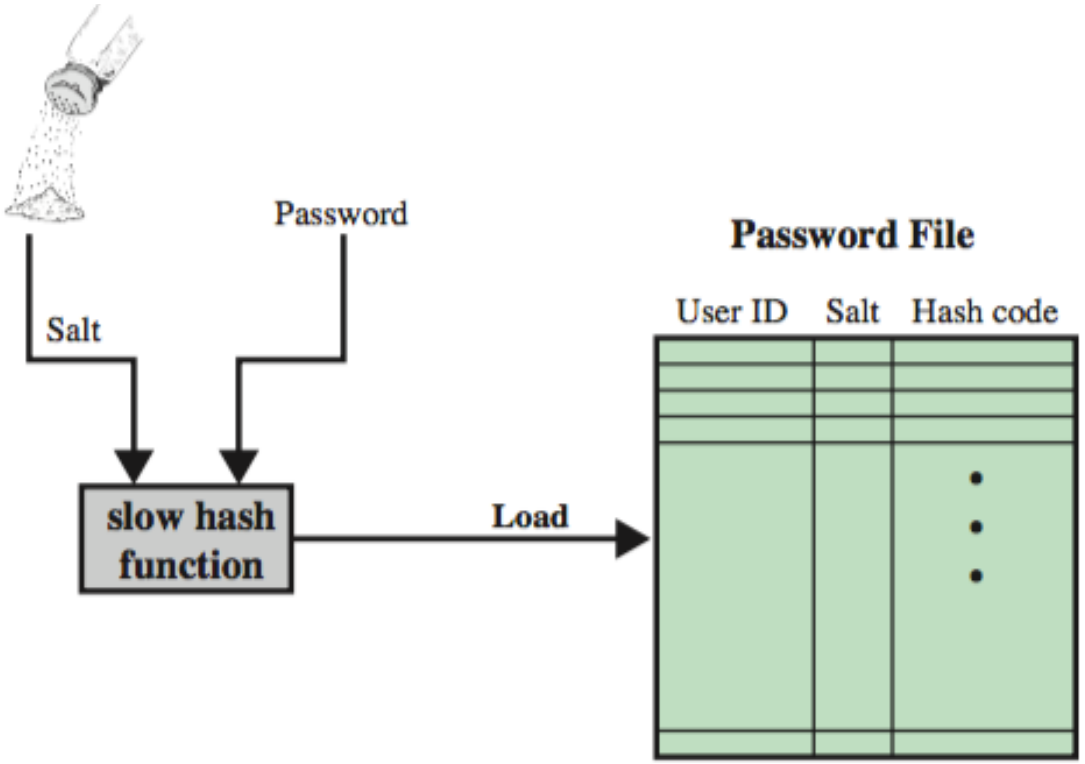


Applications of Hash Functions

- Authentication
  - It usually makes more sense to sign the hash of a message (with a private key) than to sign the original message
  - This is done with digital certificates and many other authentication schemes

Applications of Hash Functions

- Password storage
  - Store only the hash of password (+ *salt*)
  - e.g. Unix password scheme



Cryptanalysis: Breaking hash functions

- Strength depends on the length, n, in bits of the hash value

- Brute force attacks require time proportional to:
  - one-way property: 2^n
  - weak collisions property: 2^n
  - strong collisions property: $2^{n/2}$
    - This means the ability to find ***any*** two messages that hash to the same value:

Main Hash Algorithms

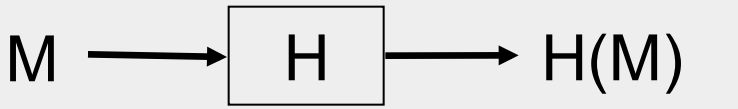
- MD5
  - Produces 128-bit hash value (i.e. 64-bit security)
  - Collisions found (2004)
  - No longer recommended for use
- SHA-1
  - Produces 160-bit hash value (80-bit security)
  - Collisions found (2017)
  - No longer recommended for use
- SHA-2
  - Set of 4 hash functions with different size outputs
  - SHA-224, SHA-256, SHA-384, SHA-512
  - Considered safe to use
    - (though new SHA-3 has been established due to concerns over structural similarities with SHA-1)

Authentication

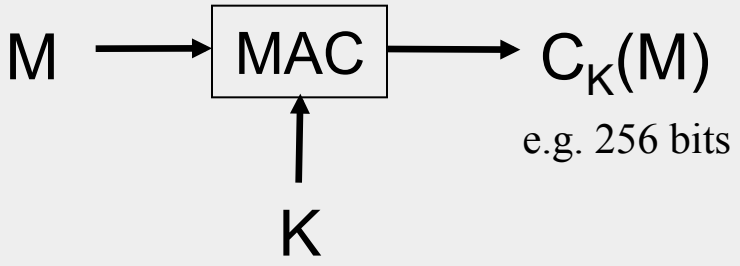
Message Authentication Code (MAC)

- Very similar to Hash Function
- Difference is the use of a **key**

Hash Function:

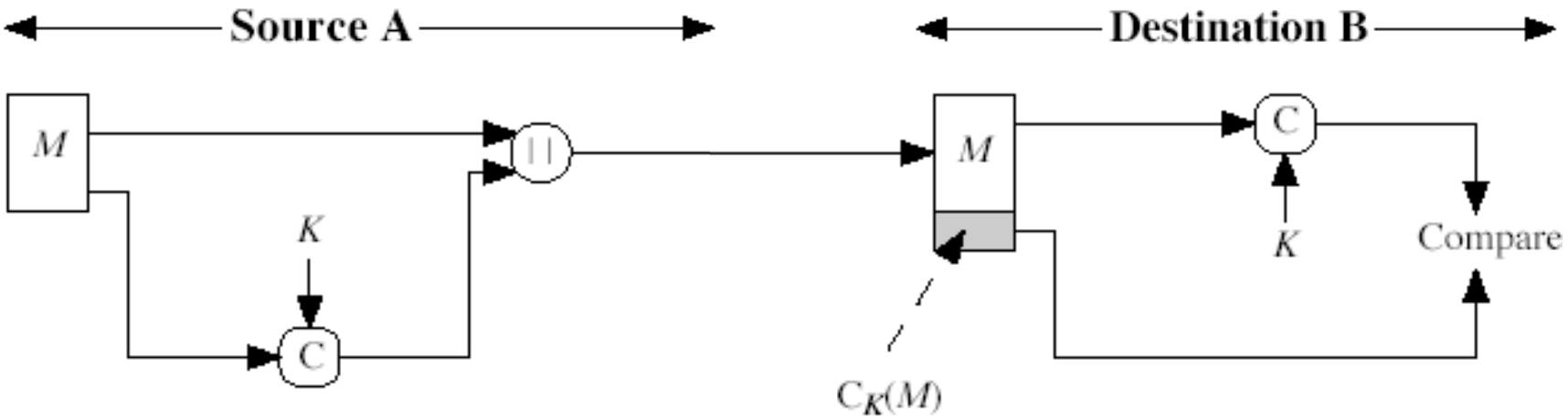


MAC:



Basic use of MAC for authentication

- Sender and recevier need to have shared secret



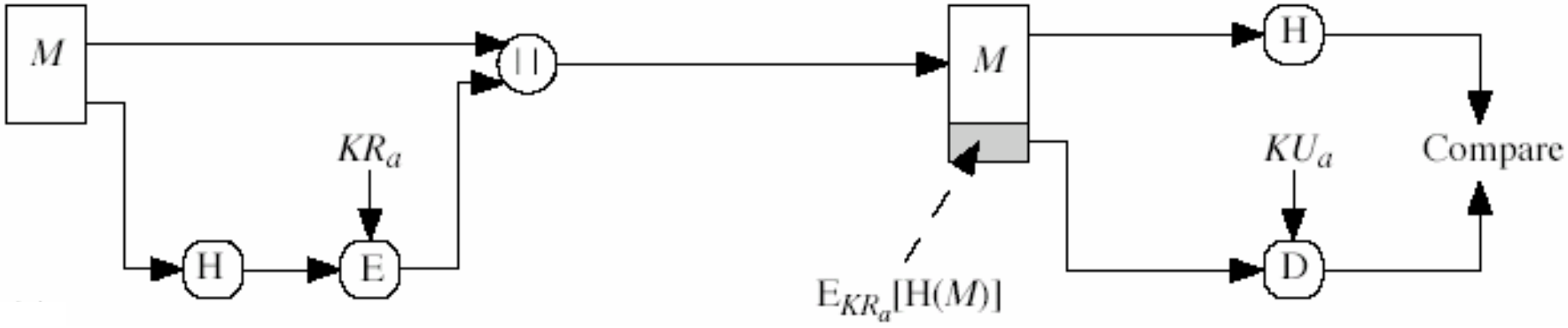
Note: The symbol with two vertical bars || means *concatenate*; i.e. join inputs together

Digital Signatures: signing the hash

- Digital signature created by adding a small authentication block to a message
- Often done by taking the hash of the message and **encrypt the hash** with the **sender's private key**
- The result is a very compact signature (relative to message size)
- And is just as secure as encrypting the entire message with the sender's private key
  - assuming that a secure hash function is used

Typical Use of Hash Function with Digital Signature

- Just sign the hash
  - much more efficient than signing full message



KR_a: Sender's Private Key

KU_a: Sender's Public Key

Note: The | | symbol means *concatenate*; i.e. join inputs together

Digital Certificates

(Public) Key Management

- **Q.** When you receive a public key, how can you be sure that it is authentic?

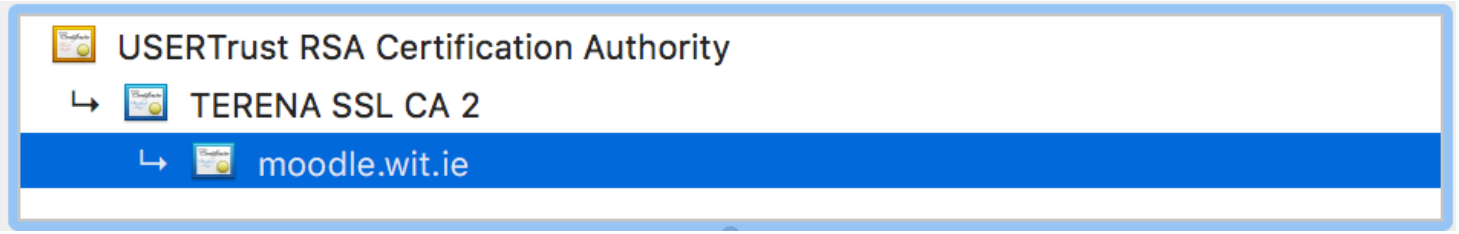
- **A.** If the received public key is **digitally signed** by someone whose own public key you have and are sure is correct **and** you trust them to sign keys responsibly.

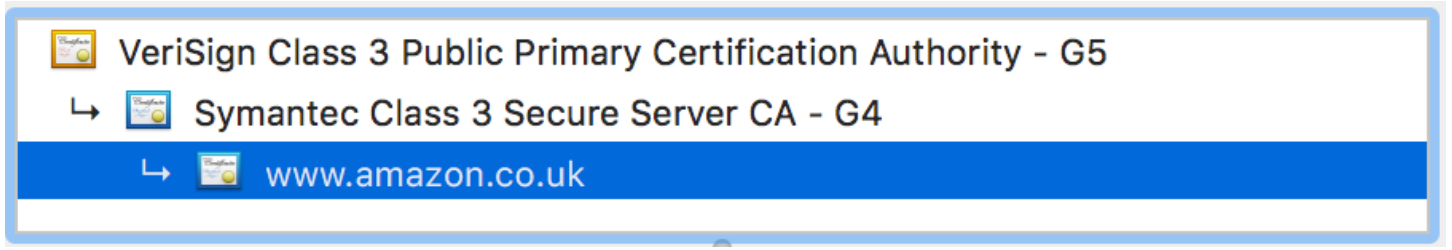
Digital Certificate – components

- Most important components of a digital certificate:
  - Subject (owner)
    - The name on the certificate – i.e. to whom it was issued
  - Subject's public key
    - The purpose of a certificate is to validate the public key of the subject
  - Issuer (Certificate Authority)
    - The identity of entity that signed the certificate
  - Issuer's digital signature
  - Serial number
    - Unique identifier for checking against revocation lists
  - Validity period
    - Start date; expiry date

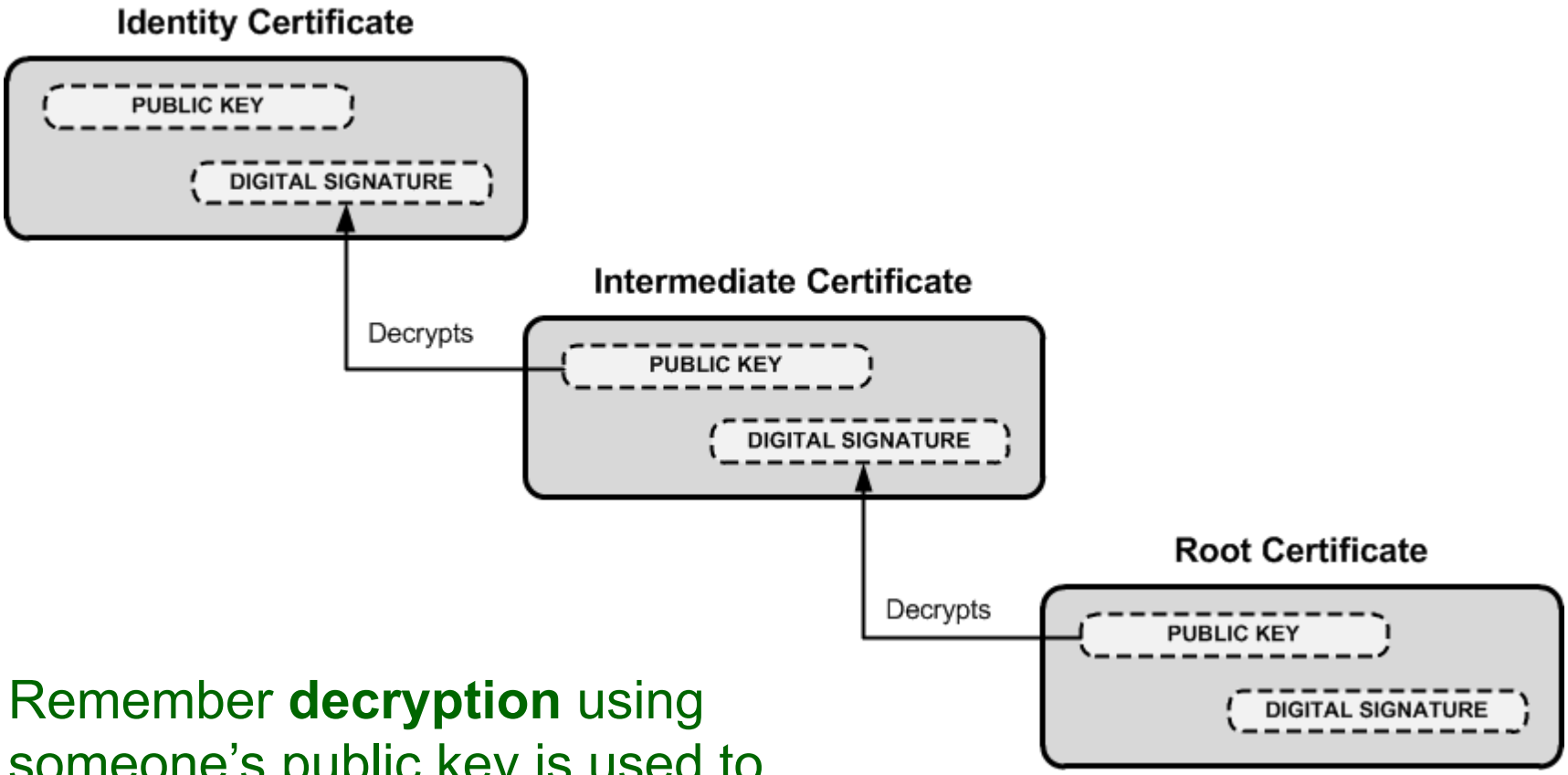
Chain of trust

- Can build up a chain of trusts with linked digital certificates
- This is the basis of what are known as Public Key Infrastructures (PKIs)





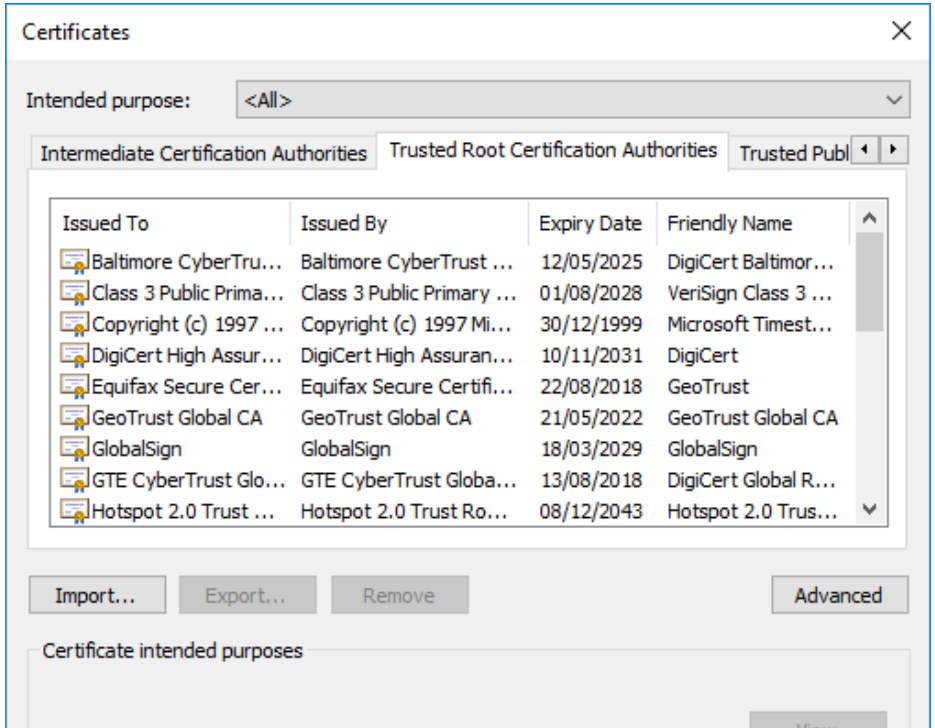
Verification using chain of trusts



Remember **decryption** using someone's public key is used to **verify** their signature

Chain of trust

- The buck must stop somewhere. Ultimately, at the end of the chain, you must trust a public key that is not signed (usually belonging to some recognised “authority”).
  - In your browser, this is one of the trusted root certificate authorities



Certificate Expiry & Revocation

- A Digital Certificate doesn't last for ever
- It normally **expires** after a certain time and must be renewed
- It may be **revoked:**
  - If the subject's private key is compromised
  - If there is a change in status of the subject
  - If the CA's private key is compromised
- Revoked certificates are placed on a Certificate Revocation List (CRL)

Certificate Revocation

- An issue is where to find CRL to check if cert has been revoked
  - One solution is to provide as part of certificate URL pointing to CRL
  - Another solution is OCSP (online certificate status protocol) which allows real time queries.
  - Another is to just rely on local list which is refreshed by browser updates (Chrome does this)