Crypto II: Public Key Cryptography

Chengyu Song

Slides modified from
Dawn Song, Dan Boneh, David Wagner, Doug Tygar

Administrivia

- Midterm
- Lab2
  - Due: next Monday

Overview

- Cryptography: **secure communication over insecure communication channels**
- Three goals
    - Confidentiality
    - Integrity
    - Authenticity
- Last lecture: symmetric-key -> confidentiality and authenticity
- This lecture: HMAC for integrity, public key, digital signature, certificate

Hash functions

- Properties
- Variable input size
- Fixed output size (e.g., 512 bits)
- Efficient to compute
  - One way only!
- Pseudo-random (mixes up input well)
  - Block cipher: two ways

Collisions

- Collision occurs when
- $x \neq y$ but $H(x) = H(y)$
- Since input space >> output space, collisions are guaranteed to happen
  - The question is, can you control it

Birthday paradox

- Ignore leap days
- Probability that two people are born on same day is 1/365
- How many people until probability of at least one common birthday > 50%?
- How many people until probability of at least one common birthday > 99%?

Birthday paradox

- Ignore leap days
- Probability that two people are born on same day is 1/365
- How many people until probability of at least one common birthday > 50%?
    - **Only 23**!
- How many people until probability of at least one common birthday > 99%?
    - **Only 70**!

Probability of collision

- Suppose hash value range is n
- And k inputs are hashed
- Probability of collision is

$$P(n,k) = 1 - \frac{n!}{(n-k)!n^k} \approx 1 - e^{-k^2/2n}$$

Cryptographic hash functions

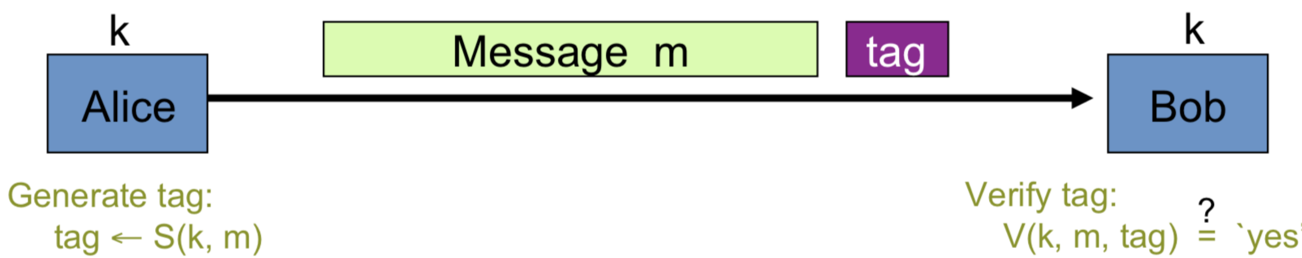
- Cryptogtaphic hash functions add conditions
- Preimage resistance
  - Given h, intractable to find y such that $H(y) = h$
- Second preimage resistance
  - Given x, intractable to find $y \neq x$ such that $H(y) = H(x)$
- Collision resistance
  - Intractable to find x, y such that $y \neq x$ and $H(y) = H(x)$

We have a hash function crisis

- Popular hash function MD5
  - Thoroughly broken
- Government standard function SHA-1, SHA-2
  - Theoretical weaknesses
  - SHA-1 is broken this year
- "New" cryptographic hash function SHA-3
  - Too new to fully evaluate
  - Maybe good enough

Message authentication code (MAC)

- Goal: provide message integrity (no confidentiality)
  - Example: Protecting public binaries on disk



NOTE: non-keyed checksum/hash is an insecure MAC !!

Secure MAC

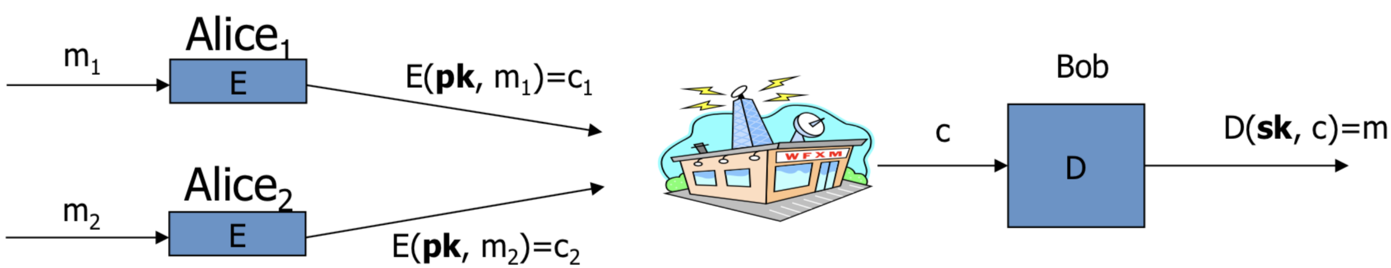
- Attacker's power: chosen message attack
  - for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k, m)$
- Attacker's goal: existential forgery
  - produce a **new** valid message/tag pair (m, t) such that
  - $(m, t) \notin \{(m_1, t_1), (m_2, t_2), \ldots, (m_q, t_q)\}$
- A secure PRF gives a secure MAC:
  - $S(k, m)$ = $F(k, m)$
  - $V(k, m, t)$: output 'yes' if $t = F(k, m)$ and 'no' otherwise

HMAC (Hash-MAC)

- Most widely used MAC on the Internet
- Building a MAC out of a hash function H (e.g., SHA-256)
- Standardized method:
  - opad, ipad: fixed strings
  - $S(k,m) = H(k \oplus opad, H(k \oplus ipad, m))$

Public-key encryption

- Motivation: how to securely exchange keys?



- E – Encryption alg. pk – Public encryption key
- D – Decryption alg. sk – Private decryption key

Algorithms E, D are publicly known.

Public-key encryption

- **Definition**: a public-key encryption system is a triple of algorithms (G, E, D)
    - $G()$: outputs a valid and randomized key pair (pk, sk)
    - $E(pk, m)$: given $m \in M$ outputs $c \in C$
    - $D(sk, c)$: given $c \in C$ outputs $m \in M$ or $\perp$
    - Consistency: $\forall (pk, sk)$ outputs by G
        - $\forall m \in M : D(sk, E(pk, m)) = m$

Trapdoor function (TDF)

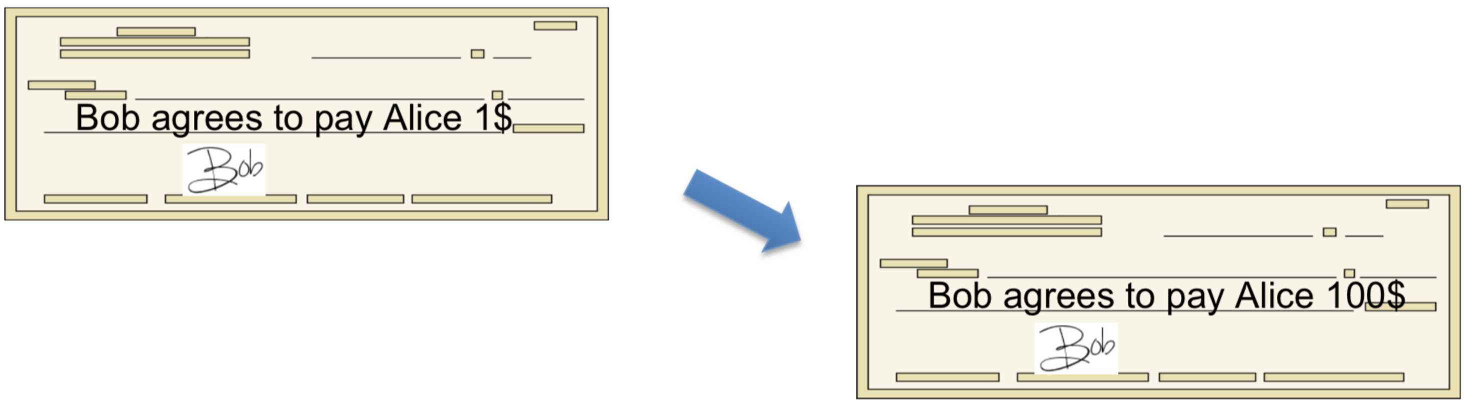
- A trapdoor function is a function that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information, i.e.,
  - $E(pk, \cdot)$ is efficient
  - $D(sk, \cdot)$ is also efficient
  - But given $c = E(pk, m)$ and pk, it is difficult to find m

Example TDF

- RSA (Rivest–Shamir–Adleman): integer factorization
- DH (Diffie–Hellman): discrete logarithm
- ECDH (Elliptic-curve Diffie–Hellman): discrete logarithm

Digital signatures

- Goal: bind document to author



- Problem: attacker can copy Bob's signature from one document to another

Digital signatures

- Solution: make signature depend on document
- **Example**: signatures from trapdoor functions (e.g. RSA)

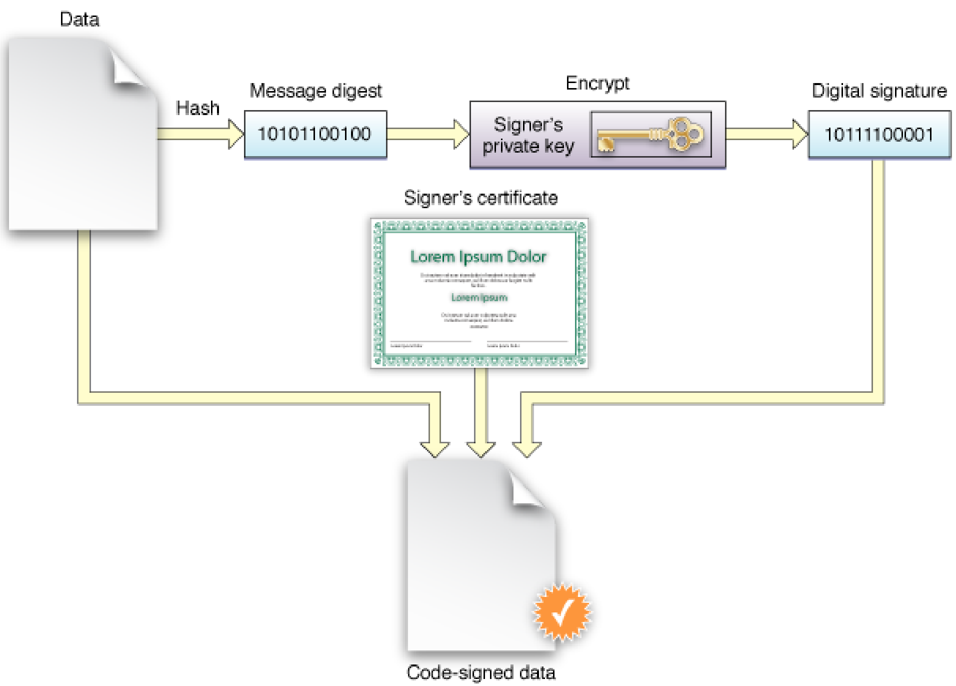
sign(sk, m) := F^{-1}(sk, H(m))

Verify(pk, m, sig) := accept if **F(pk, sig) = H(m)**

reject otherwise

Digital Signatures: applications

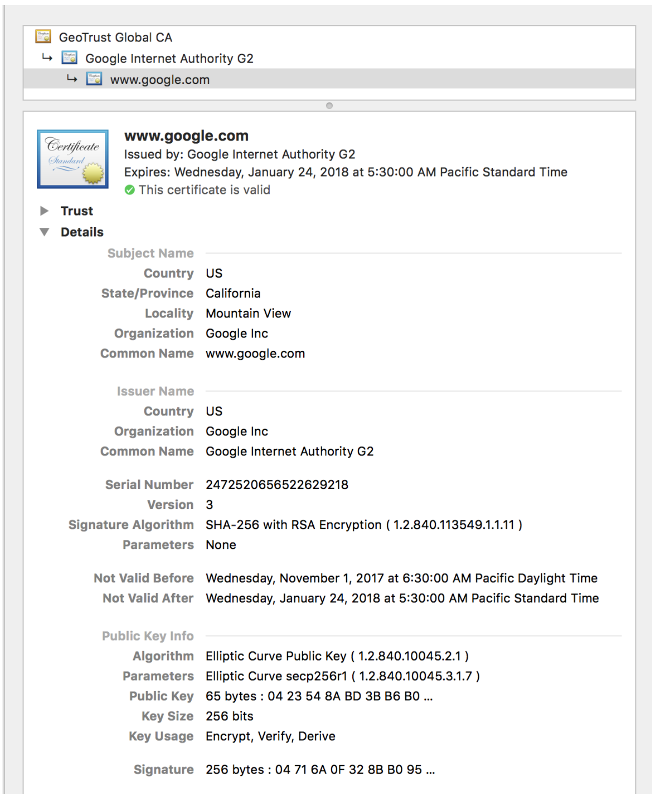
- Software distribution



Certificates

- Problem1: how do I know which key belongs to whom?
- Certificate: bind Bob's ID to his pk

Example


GeoTrust Global CA
Google Internet Authority G2
www.google.com
www.google.com
Issued by: Google Internet Authority G2
Expires: Wednesday, January 24, 2018 at 5:30:00 AM Pacific Standard Time
This certificate is valid
Trust
Details
Subject Name
Country US
State/Province California
Locality Mountain View
Organization Google Inc
Common Name www.google.com
Issuer Name
Country US
Organization Google Inc
Common Name Google Internet Authority G2
Serial Number 2472520656522629218
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters None
Not Valid Before Wednesday, November 1, 2017 at 6:30:00 AM Pacific Daylight Time
Not Valid After Wednesday, January 24, 2018 at 5:30:00 AM Pacific Standard Time
Public Key Info
Algorithm Elliptic Curve Public Key (1.2.840.10045.2.1)
Parameters Elliptic Curve secp256r1 (1.2.840.10045.3.1.7)
Public Key 65 bytes : 04 23 54 8A BD 3B B6 B0 ...
Key Size 256 bits
Key Usage Encrypt, Verify, Derive
Signature 256 bytes : 04 71 6A 0F 32 8B B0 95 ...

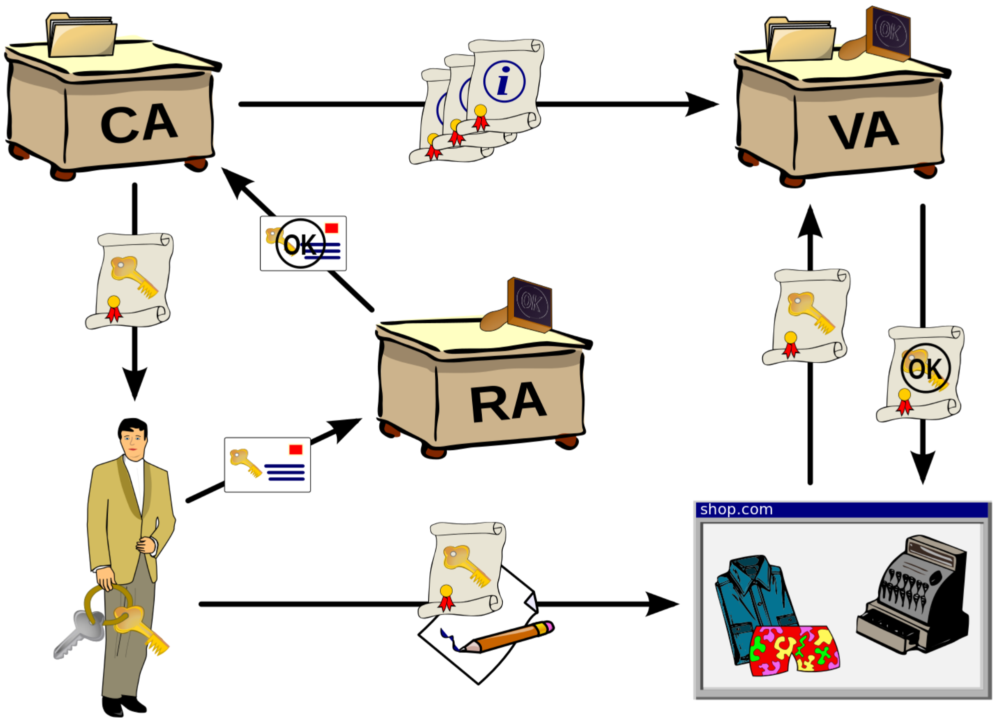
Certificate Authority (CA)

- Problem2: how would I trust a certificate?
- CA: digitally signs a certificate to make it valid

Root CA

- Problem3: how do I obtain the CA's public key to verify the certificate?
- Root CA: public key pre-installed in your system/browser and trusted by default

Public key infrastructure (PKI)



https://en.wikipedia.org/wiki/Public_key_infrastructure

Certificate revocation

- What happens if Bob loses his secret key sk?
  - Certificate on pk_{bob} must be revoked
- Revocation methods:
  - Expiration: certificates active in fixed time window (one year)
  - Certificate Revocation Lists (CRLs):
    - CA publishes a list of revoked certificates
- Online Certificate Status Protocol (OCSP)

Problems of PKI

- Hard to acquire a certificate
  - Better now with services like "Let's Encrypt"
- Rogue root CA
  - Comodo and DigiNotar CAs hacked, incorrectly issue certs for gmail.com, yahoo.com, and many others
  - CA:Symantec Issues
  - Distrusting WoSign and StartCom Certificates

For next class ...

- Network Security I: Protocol Security