Crypto: Public-Key Cryptography

Slides credit: Dan Boneh, Doug Tygar

Overview

- Last lecture: symmetric-key encryption to achieve confidentiality
- This lecture
  - HMAC for integrity and authenticity
  - Public-key encryption (RSA)
  - Digital signature
  - Certificates

Hash functions

- Properties
  - Variable input size
  - Fixed output size (e.g., 512 bits)
  - Efficient to compute
  - Pseudo-random (mixes up input well)

Collisions

- Collision occurs when
- $x \neq y$ but $H(x) = H(y)$
- Since input size > output size, collisions happen

Birthday paradox

- Ignore leapdays
- Probability that two people are born on same day is 1/365
- How many people until probability of at least one common birthday > 1/2
- Surprising answer 23 (!)

Probability of a collision

- Suppose hash value range is n
- And k input points are hashed

- Probability of a collision is

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

Cryptographic hash functions

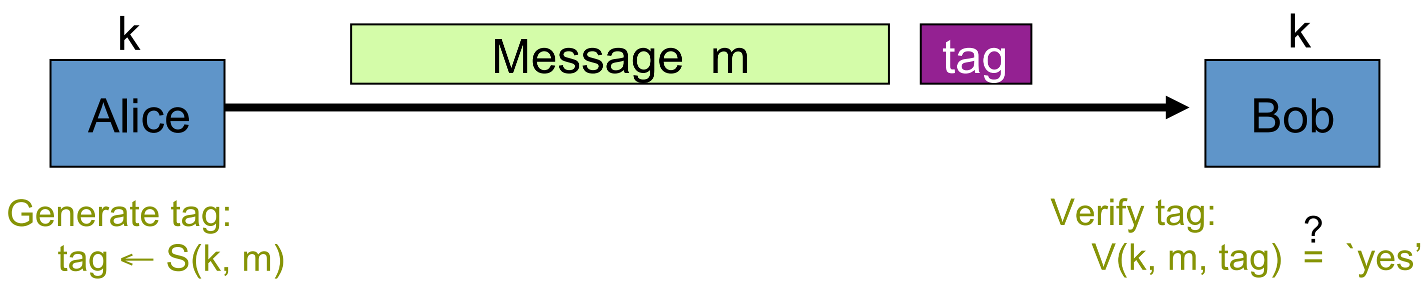
- Cryptogtaphic hash functions add conditions
- Preimage resistance
  - Given h, intractable to find y such that $H(y)=h$
- Second preimage resistance
  - Given x, intractable to find $y \neq x$ such that $H(y)=H(x)$
- Collision resistance
  - Intractable to find x, y such that $y \neq x$ and $H(y)=H(x)$

We have a hash function crisis

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

Message Integrity: MACs

- Goal: provide message integrity. No confidentiality.
  - ex: Protecting public binaries on disk.



note: non-keyed checksum (CRC) is an insecure MAC !!

Secure MACs

Attacker's power: chosen message attack.

- for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k, m_i)$

Attacker's goal: existential forgery.

- produce some **new** valid message/tag pair (m,t).

  $(m,t) \notin \{ (m_1,t_1), \ldots, (m_q,t_q) \}$

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.

H: hash function.
example: SHA-256 ; output is 256 bits

Building a MAC out of a hash function:

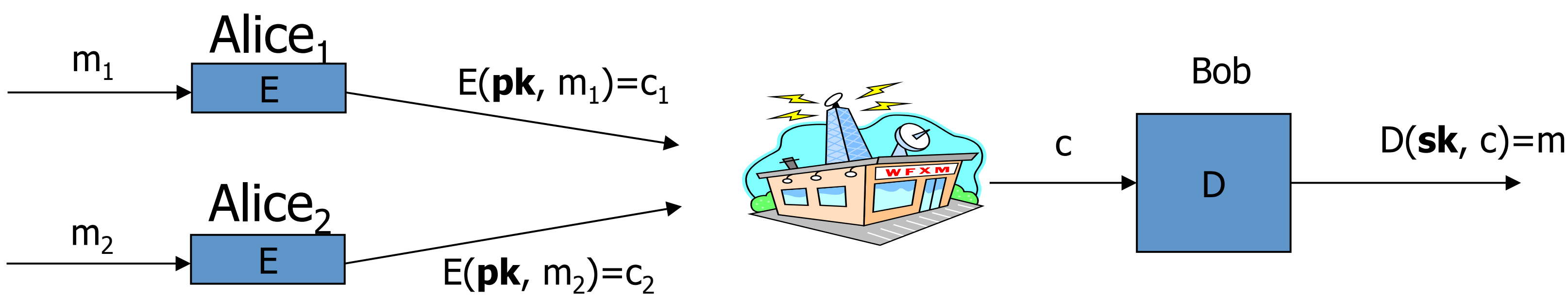
opad, ipad: fixed strings

– Standardized method: HMAC

$S(k, m) = H(k \oplus opad, \mathbf{H(k \oplus ipad, m)})$

Public-key encryption

Tool for managing or generating symmetric keys



- E – Encryption alg. pk – <u>Public</u> encryption key
- D – Decryption alg. sk – <u>Private</u> decryption key

Algorithms E, D are publicly known.

Public key encryption

Def: a public-key encryption system is a triple of algs. (G, E, D)

- G(): randomized alg. outputs a key pair (pk, sk)
- E(pk, m): randomized alg. that takes $m \in M$ and outputs $c \in C$
- D(sk, c): det. alg. that takes $c \in C$ and outputs $m \in M$ or $\bot$

Consistency: $\forall$ (pk, sk) output by G :

$$\forall m \in M: \quad D(sk, E(pk, m)) = m$$

Building Block: Trapdoor Functions (TDF)

Def: a trapdoor function over X is a triple of efficient algs. (G, F, F^{-1})

- **G**(): randomized alg. outputs a key pair (pk, sk)
- **F**(pk, ⋅): deterministic alg. that defines a function $X \longmapsto Y$
- $\mathbf{F^{-1}}$(sk, ⋅): defines a function $Y \longmapsto X$ that inverts F(pk, ⋅)

for all x in X: $F^{-1}(sk, F(pk, x)) = x$

Security: (G, F, F^{-1}) is secure if F(pk, ⋅) is a "one-way" function:

given **F(pk, x)** and **pk** it is difficult to find **x**

Example TDF: RSA

- alg. G(): generate two equal length primes p, q

  set $N \leftarrow p \cdot q$ (3072 bits ≈ 925 digits)

  set $e \leftarrow 2^{16}+1 = 65537$; $d \leftarrow e^{-1} \pmod{\varphi(N)}$

  pk = (N, e) ; sk = (N, d)

- RSA(pk, x) : $x \longrightarrow (x^e \bmod N)$

  Inverting this function is believed to be as hard as factoring N

- RSA^{-1}(sk, y) : $y \longrightarrow (y^d \bmod N)$

Public Key Encryption with a TDF

G(): generate pk and sk

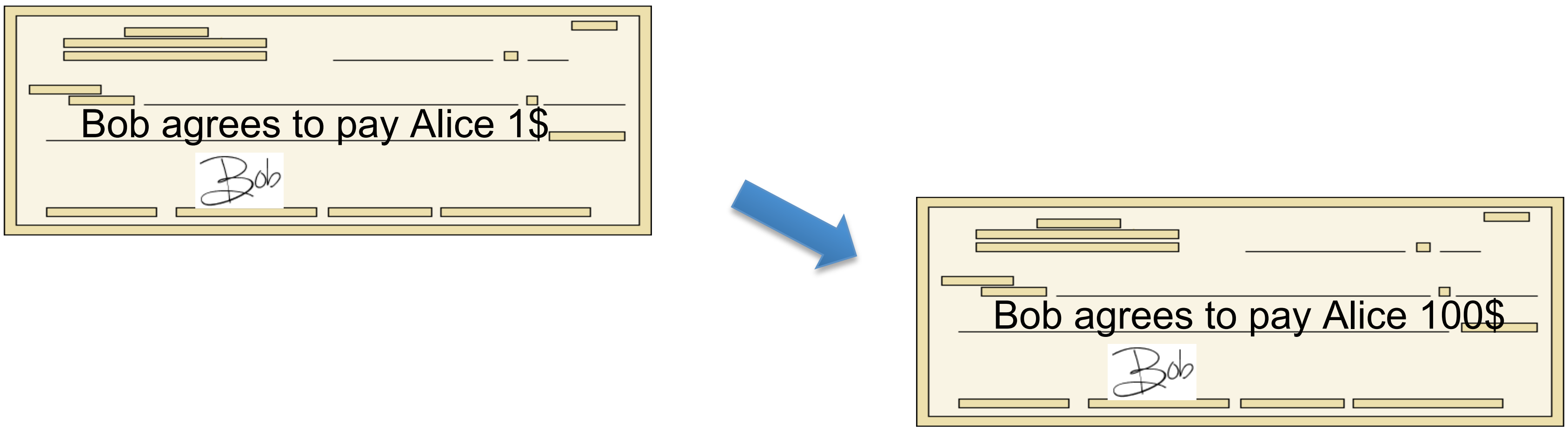
E(pk, m):

- choose random $x \in \text{domain}(F)$ and set $k \leftarrow H(x)$
- $c_0 \leftarrow F(pk, x)$, $c_1 \leftarrow E(k, m)$ (E: symm. cipher)
- send $c = (c_0, c_1)$

D(sk, $c=(c_0,c_1)$): $x \leftarrow F^{-1}(sk, c_0)$, $k \leftarrow H(x)$, $m \leftarrow D(k, c_1)$

Digital signatures

Goal: bind document to author



Problem: attacker can copy Bob's sig from one doc to another

Digital signatures

Solution: make signature depend on document

Example: signatures from trapdoor functions (e.g. RSA)

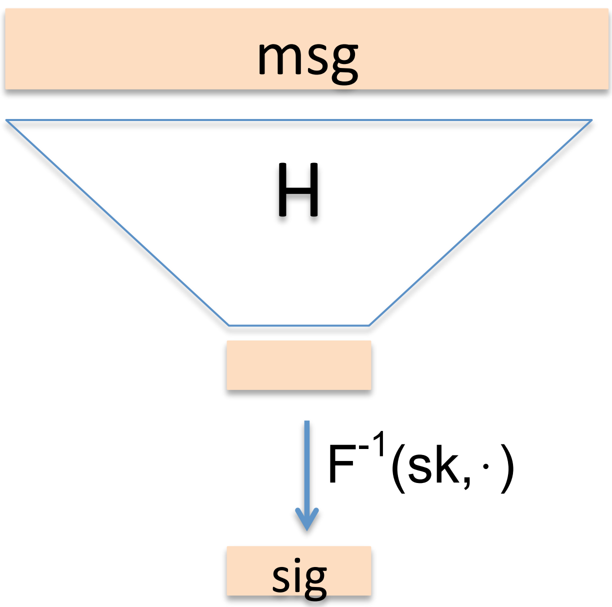
sign(sk, m) := F^{-1} (sk, H(m))

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

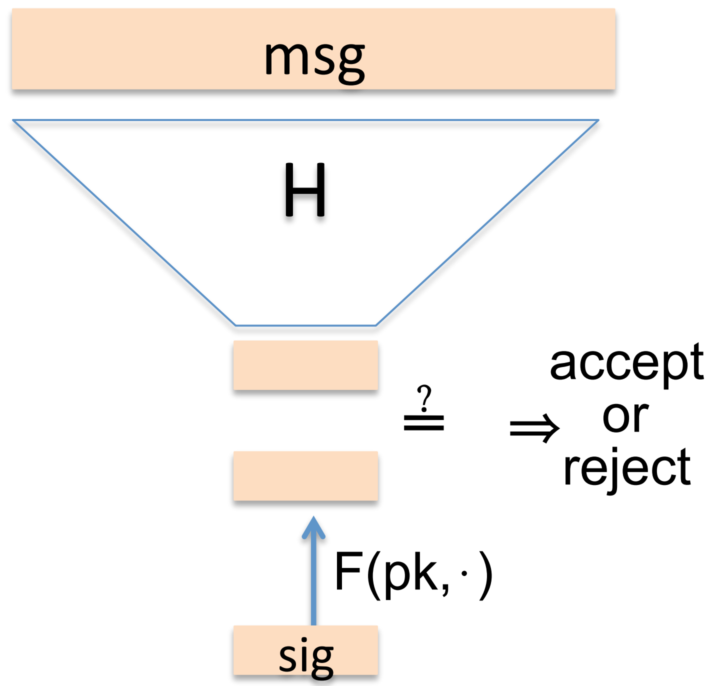
reject otherwise

Digital Sigs. from Trapdoor Functions

sign(sk, msg):



verify(pk, msg, sig):



Digital Signatures: applications

- Software distribution