Digital Signature

Digital Signatures

- NIST FIPS PUB 186-4 - the result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity, and signatory non-repudiation

  no ~~confidentiality~~

- Based on asymmetric keys

Digital Signatures

H( )

- Asymmetric cryptography is good because we don't need to share a secret key
- Digital signatures are the asymmetric way of providing integrity/authenticity to data
- Assume that Alice and Bob can communicate public keys without David interfering

PKI

Third Authority Party

certificate

Sig(PK.)SK

correct

sender — receiver

private — public. key

sign — verify

correct

Digital Signatures: Definition

- Three parts:
  - KeyGen() → *PK*, *SK*: Generate a public/private keypair, where *PK* is the verify (public) key, and *SK* is the signing (secret) key
  - Sign(*SK*, *M*) → *sig*: Sign the message *M* using the signing key *SK* to produce the signature *sig*
  - Verify(*PK*, *M*, *sig*) → {0, 1}: Verify the signature *sig* on message *M* using the verify key *PK* and output 1 if valid and 0 if invalid
- Properties:
  - **Correctness**: Verification should be successful for a signature generated over any message
    - Verify(*PK*, *M*, Sign(*SK*, *M*)) = 1 for all *PK*, *SK* ← KeyGen() and *M*
  - **Efficiency**: Signing/verifying should be fast 比C ?
  - **Security**: Same as for MACs except that the attacker also receives PK
    - Namely, no attacker can forge a signature for a message

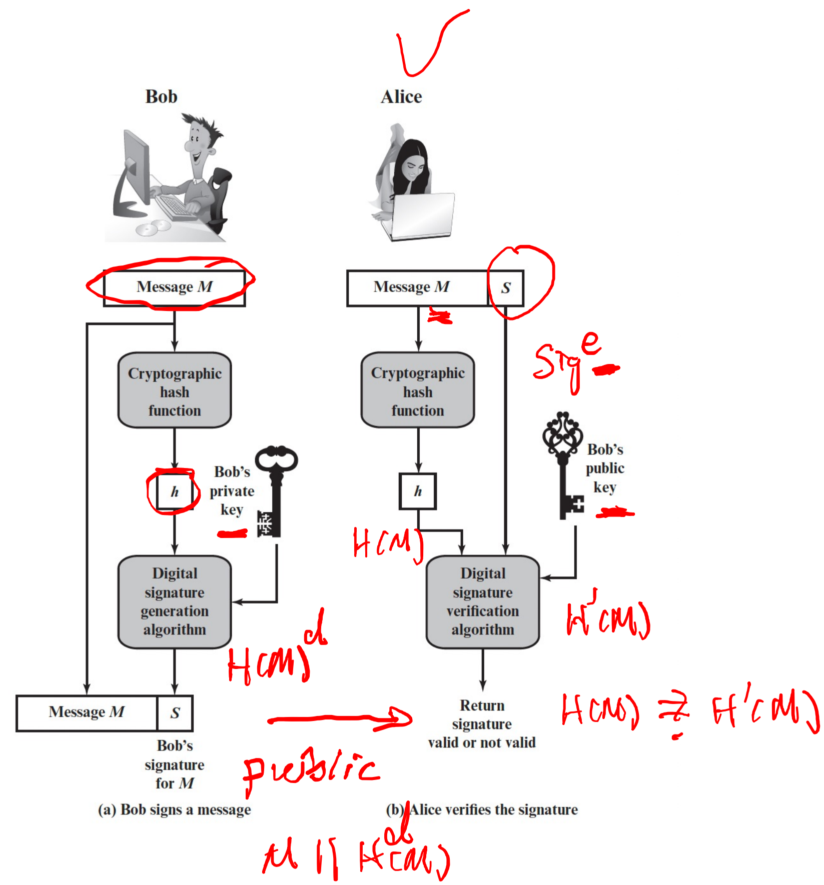
without private key

RSA Signature

- KeyGen(): Same as RSA Encryption
  - Randomly pick two large primes, p and q
  - Compute $n = pq$ → large number, public.
    - n is usually between 2048 bits and 4096 bits long
  - Choose e
    - Requirement: e is relatively prime to $(p - 1)(q - 1)$ $= \phi(n)$. $\gcd(e, \phi(n)) = 1$
    - Requirement: $2 < e < (p - 1)(q - 1)$ $= \phi(n)$
  - Compute $d = e^{-1} \bmod (p - 1)(q - 1)$ $\phi(n)$.
  - **Public key**: n and e → public
  - **Private key:** d

RSA Signatures

- Sign(d, M):
  - Compute $H(M)^d \bmod n$
- Verify(e, n, M, sig)
  - Verify that $H(M) \equiv sig^e \bmod n$



(a) Bob signs a message
(b) Alice verifies the signature

RSA Probabilistic Digital Signature Scheme (RSA-PSS)

signature to be fast.

H(M)

Step1: Generate a hash value, or message digest, mHash from the message *M* to be signed

Step2: Pad mHash with a constant value padding1 and pseudorandom value salt to form *M'*

Step3: Generate hash value *H* from *M'*

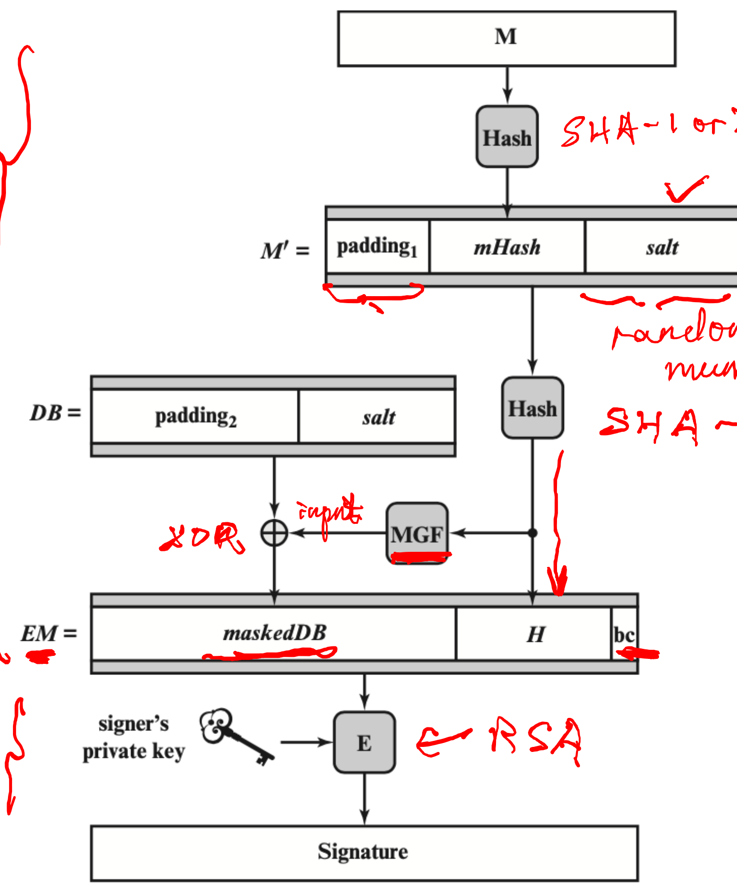
Step4: Generate a block DB consisting of a constant value padding 2 and salt

Step5: Use the mask generating function MGF, which produces a randomized out-put from input *H* of the same length as DB

Step 6: Create the encoded message (EM) block by padding *H* with the hexadecimal constant bc and the XOR of DB and output of MGF

Step 7: Encrypt EM with RSA using the signer's private key

H(M) d



RSA Signatures: Correctness

Same prove process as RSA decryption

$H(M)$ = fast, less bandwidth

Theorem: $sig^e \equiv H(M)$ mod N

Proof:

Because $Sig = H(M)^d$ mod N

$$sig^e = [H(M)^d\ ]^e\ mod\ N \quad = H(M)^{ed} \text{ mod N}$$

$= H(M)^{de}$ mod N

Because $d \cdot e = 1$ mod $\phi(n)$ by definition of modular area

$d \cdot e = 1 + k \cdot \phi(n)$ $\quad k \in \mathbb{Z}$

$= H(M)^{k \cdot \phi(N) + 1}$ mod N

$= [H(M)^{\phi(N)}]^k \cdot H(M)$ mod N

Euler's theorem: if $gcd(M, N) = 1$ then $M^{\phi(N)} = 1$ mod N

$= 1^k \cdot H(M)$ mod N

$= H(M)$ mod N ■

RSA Signatures: Correctness

Theorem: $sig^e \equiv H(M) \bmod N$

Proof:

$$sig^e = [H(M)^d\ ]^e \ mod\ N \quad = H(M)^{ed} \text{ mod N}$$

$$= H(M)^{k\phi(n)+1} \text{ mod N}$$

$$= [H(M)^{\phi(n)}]^k \cdot H(M) \quad \text{mod N}$$

$$= H(M) \quad \text{mod N}$$

Homework (Textbook) – no submission

- Review Question: 3.1, 3.2, 3.3, 3.4, 3.5, 3.6
- Problems:
  - prove correctness of RSA digital signature
  - 3.14 & 3.15

Homework 2 - individual

- Chapter 3
- **Deadline**: Friday, October 24 before class
- We will use the RaiderCanvas submission time as your final timestamp
- 10% penalty per day for late submission