

Lecture 21

RSA Signature

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

RSA Digital Signature Algo

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'

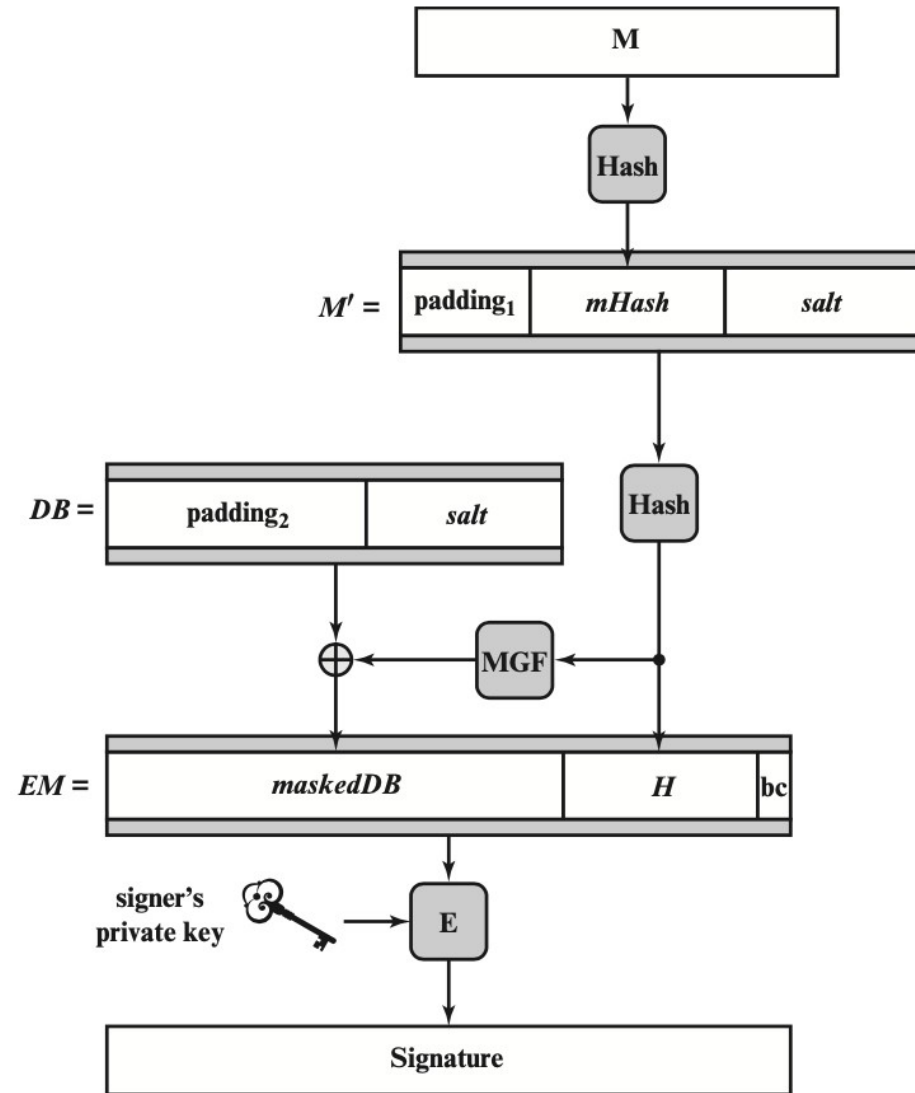
Step3: Generate hash value H from M'

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

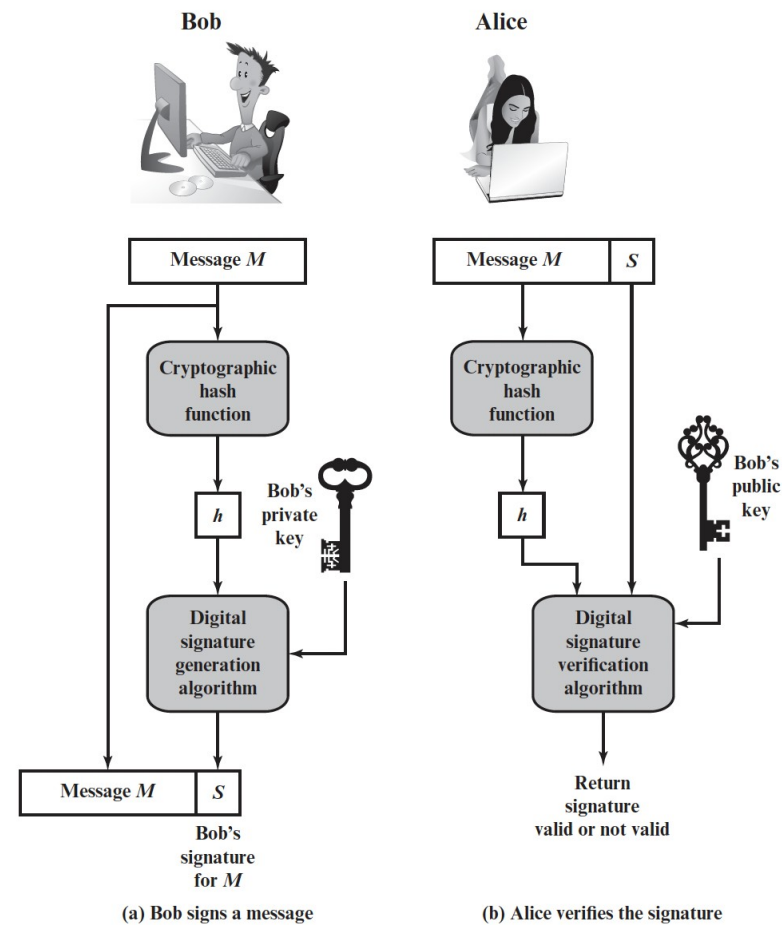
Step 6: Create the encoded message (EM) block by padding H with the hexadecimal constant bc and the XOR of DB and output of MGF

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



RSA Signatures

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



RSA Signatures: Correctness

Theorem: $sig^e \equiv H(M) \pmod{N}$

Proof:

$$sig^e = \quad \pmod{N}$$

$$\pmod{N}$$

$$i [H(M)^{\phi(n)}]^k \cdot H(M) \pmod{N}$$

$$= H(M) \pmod{N}$$

RSA Digital Signature: Security

- **Necessary hardness assumptions:**
 - **Factoring hardness assumption:** Given n large, it is hard to find primes p, q such that $pq = n$
 - **Discrete logarithm hardness assumption:** Given n large, $hash$, and $hash^d \bmod n$, it is hard to find d
- Salt also adds security
 - Even the same message and private key will get different signatures

Hybrid Encryption

- Issues with public-key encryption
 - Notice: We can only encrypt small messages because of the modulo operator
 - Notice: There is a lot of math, and computers are slow at math
 - Result: We don't use asymmetric for large messages
- **Hybrid encryption:** Encrypt data under a randomly generated key K using symmetric encryption, and encrypt K using asymmetric encryption
 - $\text{Enc}_{\text{Asym}}(\text{PK}, K); \text{Enc}_{\text{Sym}}(K, \text{large message})$
 - Benefit: Now we can encrypt large amounts of data quickly using symmetric encryption, and we still have the security of asymmetric encryption

Homework – no submission

- RQ: 3.1, 3.2, 3.3, 3.4, 3.6, 3.7
- Problems:
 - prove correctness of RSA digital signature
 - 3.14

Homework 2 - individual

- For Chapter 3
- Deadline: Oct. 26 (Thursday), 11:59 pm
- We will use the blackboard submission time as your final timestamp
- 10% penalty per day for late submission

Thank you!