

MH4311 Cryptography

Lecture 15

Digital Signature

Wu Hongjun

Lecture Outline

- Classical ciphers
- Symmetric key encryption
- Hash function and Message Authentication Code
- Public key encryption
- **Digital signature**
 - **RSA signature scheme**
 - **ElGamal signature scheme**
 - **Digital Signature Standard (DSS)**
 - **Digital Signature Algorithm (DSA)**
 - **RSA Digital Signature Algorithm**
- Key establishment and management
- Introduction to other cryptographic topics

Recommended Reading

- CTP: Chapter 7
- HAC: Chapter 11
- FIPS 186-3 (2009)
 - Digital Signature Standard (DSS)
http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
- Wikipedia
 - Digital signature
http://en.wikipedia.org/wiki/Digital_signature
 - ElGamal signature scheme
http://en.wikipedia.org/wiki/ElGamal_signature_scheme
 - Digital signature algorithm (DSA)
http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

Digital Signature

- Handwritten signature
 - for personal/object authentication (identification)
 - signature verification at bank for account access
 - signatures on attendance sheet
 -
 - for message authentication
 - signing a document, contract
- Security requirements on signature
 - Difficult to forge a signature
 - Theoretically it is **not difficult to forge a handwritten signature**
 - Combine the signature with the document being signed

Digital Signature

- How to authenticate a **digital** document?
 - handwritten signature in digital document
 - Inconvenient
 - Not strong
 - Easy to forge: copy & paste
 - Difficult to combine the signature together with the document
 - digital signature is needed


Digital Signature

- Digital signature
 - Message authentication using public key cryptosystem
 - Alice signs documents using her private key
 - Everyone can verify Alice's signatures using Alice's public key
 - Strong security, easy to verify
 - **Message authentication**
 - **Non-repudiation (a user cannot deny the digital signatures signed by him/her since only one person can generate that digital signature)**
 - Different from message authentication code
 - Digital signature: public/private key; non-repudiation
 - MAC: symmetric key;

Digital Signature

- A digital signature algorithm includes:
 - Key generation
 - Public key/private key
 - Signature generation
 - Signature verification

Digital Signature

- Digital signature schemes
 - RSA signature scheme
 - ElGamal signature scheme
 - **Digital Signature Standards (DSS)**  Used in applications
 - Digital Signature Algorithm (DSA)
 - RSA Digital Signature Algorithm
 - Elliptic Curve Digital Signature Algorithm (ECDSA)

RSA Signature Scheme

Key Generation of RSA Signature Scheme

- Public key: (n, e)
- Private key: d
- Key generation of RSA Signature is identical to that of RSA encryption
 - Avoid using the same public key and private key for both signature & encryption

RSA Signature Generation & Verification

- Signature generation:

$$s = (H(m))^d \bmod n \quad (H \text{ is a hash function})$$

s is the signature of message m .

- Signature verification:

$$s^e \bmod n \stackrel{?}{=} H(m)$$

Hash function must be used in digital signature so that we can sign messages with arbitrary length.

For a strong hash function, computationally each message digest represents only one message.

(In history, cryptographic hash functions were designed for the application of digital signature.)

RSA Signature Scheme

$$s = (H(m))^d \bmod n$$

- *Insecure against multiplicative forgery for some message digest sizes (say, 160-bit)
 - Vulnerable to chosen-message attack
 - a 160-bit random integer is 2^{10} -smooth with prob. about 2^{-40}
 - An attacker generates some messages (M_i) with “smooth” message digests, then request for the signatures of these messages
 - The attacker can forge the signature of some message M if

$$H(M) = H(M_i)^{\alpha_i} \times H(M_j)^{\alpha_j} \times \cdots \times H(M_x)^{\alpha_x}$$

- *For secure RSA signature, pad the message digest before signing
 - To learn later

ElGamal Signature Scheme

Key Generation of ElGamal Signature Scheme

- Public key: (p, g, y) $(y = g^x \bmod p)$
- Private key: x
- Key generation of ElGamal Signature Scheme is the same as that of ElGamal Encryption

ElGamal Signature Generation

- Signature generation
 - Choose a random secret k with $\gcd(k, p-1) = 1$
 - Compute $r = g^k \bmod p$
 - Compute $s = (H(m) - xr) k^{-1} \pmod{p-1}$
 - If $s = 0$, start over again

The signature of message m is: (r, s)

ElGamal Signature Verification

$$r = g^k \bmod p$$

$$s = (H(m) - xr) k^{-1} \pmod{p-1}$$

- Signature verification
 - $0 < r < p, 0 < s < p-1$
 - $g^{H(m)} \bmod p \neq y^r r^s \bmod p$

ElGamal Signature Scheme

$$r = g^k \bmod p$$

$$s = (H(m) - xr) k^{-1} \pmod{p-1}$$

- Security

- use each k only once

- Otherwise, x can be recovered

Digital Signature Standards

Digital Signature Standard (DSS)

- Digital Signature Standard (FIPS 186-4, 2013)
 - **Digital Signature Algorithm (DSA)**
 - Based on discrete logarithm
 - A variant of ElGamal signature
 - The size of the signature of DSA is smaller than that of ElGamal Signature
 - **RSA Digital Signature Algorithm**
 - **Elliptic Curve Digital Signature Algorithm (ECDSA)**
 - These digital signature standards are widely used in applications

Digital Signature Algorithm (DSA)

Key generation (Phase 1):

- Generate an N -bit prime q (example: $N = 256, L = 3072$)
- Generate an L -bit prime modulus p such that $p-1$ is a multiple of q
- Generate g which is a generator of a subgroup of order q in the multiplicative group Z_p^* . (g 's multiplicative order modulo p is q , *i.e.*, q is the smallest positive integer such that $g^q \bmod p = 1$)
 - Set $g = h^{(p-1)/q} \bmod p$ for some integer h , g should not be 1.
- The algorithm parameters (p, q, g) can be used by all the users of the digital signature algorithm
- key length L and N
 - ~~(1024,160)~~
 - (2048,224)
 - (2048,256)
 - (3072,256)

Digital Signature Algorithm (DSA)

Key generation (Phase 2):

- Compute private and public keys for a single user:
 - Choose a random secret integer x , where $0 < x < q$
 - Calculate $y = g^x \bmod p$

Public key: (p, q, g, y)

Private key: x

Digital Signature Algorithm (DSA)

Signature Generation:

- Generate a random per-message secret integer k where $0 < k < q$
- Calculate $r = (g^k \bmod p) \bmod q$
- Calculate $s = (k^{-1}(H(m) + xr)) \bmod q$
- Recalculate the signature if $r = 0$ or $s = 0$

The signature is: (r, s)

(The signature size of DSA is much smaller than that of ElGamal Signature)

Digital Signature Algorithm (DSA)

$$r = (g^k \bmod p) \bmod q$$

$$s = (k^{-1}(H(m) + xr)) \bmod q$$

Signature Verification:

- $0 < r < q$ and $0 < s < q$
- Calculate $u1 = H(m)s^{-1} \bmod q$
- Calculate $u2 = rs^{-1} \bmod q$
- Calculate $v = ((g^{u1} y^{u2}) \bmod p) \bmod q$
- $v \neq r$

Digital Signature Algorithm (DSA)

- Hash function being used in DSA
 - ~~SHA-1~~
 - SHA-2

Digital Signature Algorithm (DSA)

- Security of DSA
 - The one-time per-message k should be secret, and should never be reused
 - But it is not easy to generate random numbers in application
 - In Sony's PlayStation, Sony uses ECDSA to sign the codes. Only the codes signed by Sony can run on PlayStation3. In 2010, it was revealed that Sony used a constant as k , so the private key of Sony was leaked, then any code can be signed and run on PlayStation3.

Digital Signature Algorithm (DSA)

- Security of DSA (cont.)
 - **Deterministic signature**: In order to eliminate the risk of using weak k in DSA, it was proposed in RFC 6769 to generate k in a deterministic way by hashing the private key of DSA together with the message digest of the message to be signed
 - The above approach improves the security of DSA and ECDSA since a secure k can always be generated.
 - The deterministic DSA and ECDSA now become popular. For example, the Bitcoin crypto currency.

RSA Digital Signature Algorithm

- Message padding is used to resist multiplicative forgery
- Two versions of RSA Digital Signature Algorithm
 - One version specified in ANSI X9.31
 - ANSI: American National Standard Institute (private organization)
 - Another version specified in PKCS#1 v2.1
 - PKCS: Public-Key Cryptography Standard
 - Published by RSA lab

RSA Digital Signature Algorithm

ANSI X9.31

- Message padding:

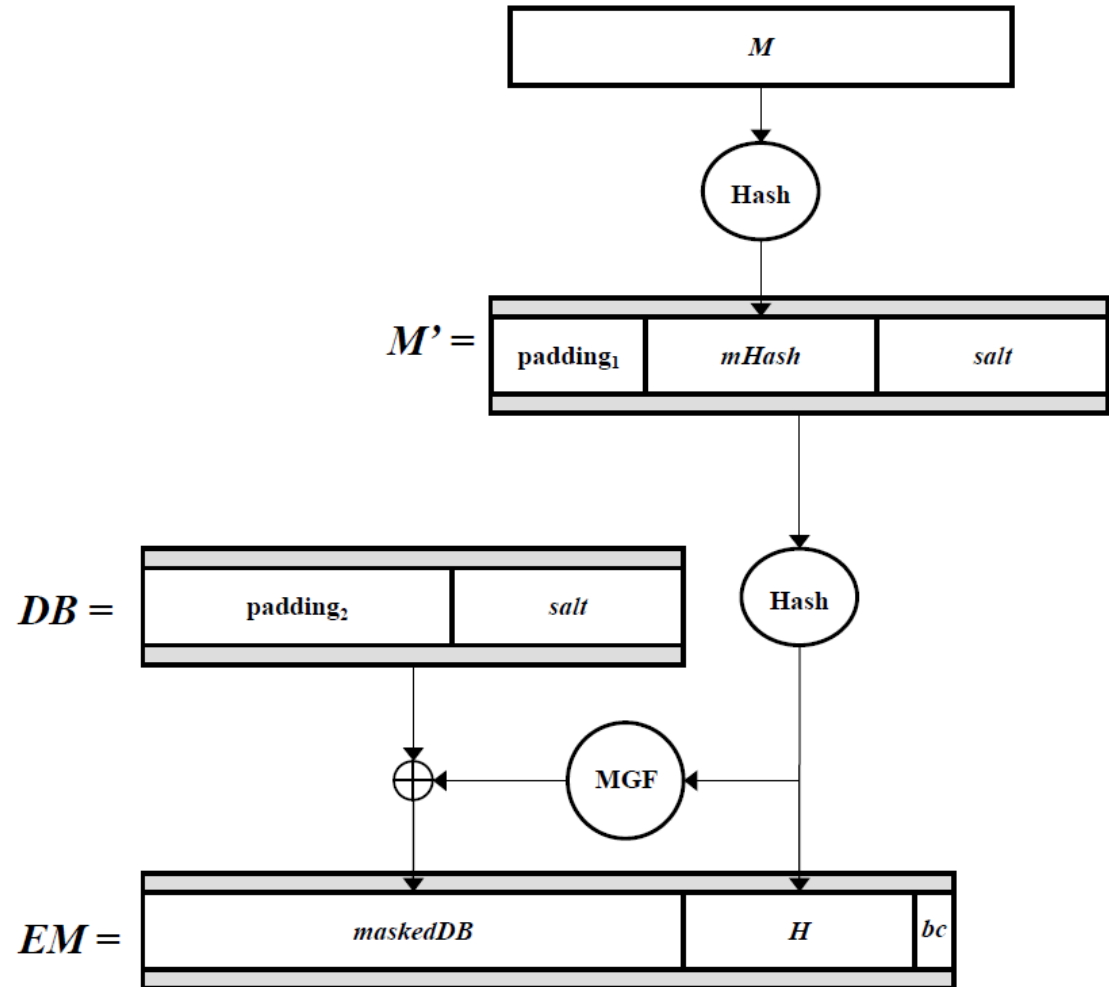
6b bb ... bb ba || Hash(M) || 3 x cc

x is a small integer, indicating which hash function is used.

- Resistant to multiplicative forgery since the message after padding becomes very large
- Standardized
 - IEEE P1363, ISO/IEC 14888-3
 - FIPS 186-3

RSA Digital Signature Algorithm

- Padding in RSASSA-PSS in PKCS#1 v2.1 (will not be tested)



Digital Signature & Hash Function

- Hash function must be used for digital signature
 - Main Reason:
 - To compress arbitrary messages into message digests with fixed length for signing
 - Another reason:
 - Randomizing the message before signing
 - Resist forgery attack

Digital Signature & Hash Function

- The security of hash function & the security of digital signature
 - Hash function for digital signature must be collision resistant
 - Important for the non-repudiation property of digital signature
 - Suppose that the hash function is not collision resistant, then Alice can find two different messages m and m' so that $H(m) = H(m')$. Now both Alice and Bob sign the contract m . Later, Alice can insist that what she signed is message m' since m' and the signature of m can pass the verification.

Applications of Digital Signature

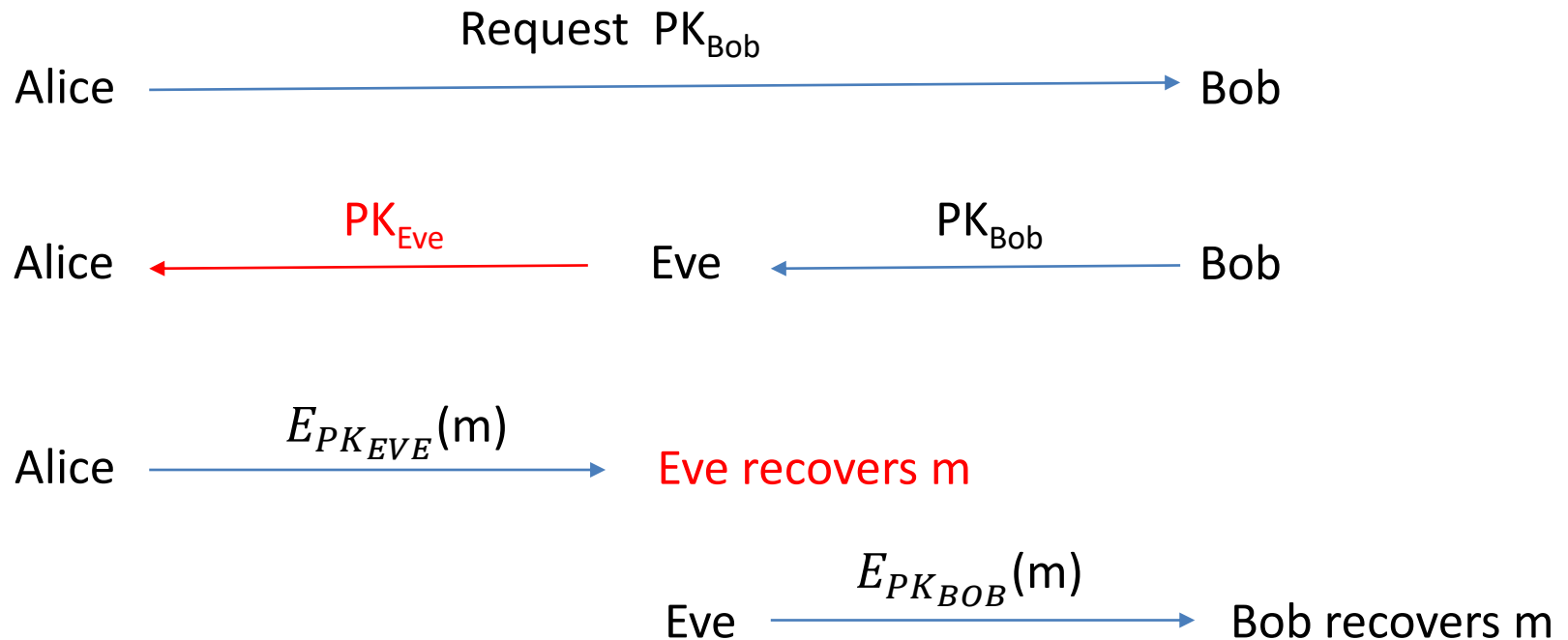
Authenticate Public Keys with Digital Signature

- Authenticating public keys
 - The most widely used application of digital signature
 - Important for resisting impersonation attack

Authenticate Public Keys with Digital Signature

- What is the risk if public keys are not authenticated?
 - Suppose that Alice wants to send a message m to Bob. Alice wants to encrypt the message using Bob's public key.
 - Alice requests the public key from Bob. During the transmission, Eve replaces Bob's public key with Eve's public key, so Alice receives Eve's public key instead of Bob's public key.
 - Alice then encrypts message m using the public key received, then try to send the ciphertext to Bob. During the transmission, Eve decrypts Alice's ciphertext to obtain m , then encrypts m using Bob's public key, then sends the new ciphertext to Bob. Bob can recover the message m .
 - In the above **man-in-the-middle attack**, Eve is able to recover the message m without being detected by Alice and Bob.

Man-in-the-middle attack against un-authenticated public key



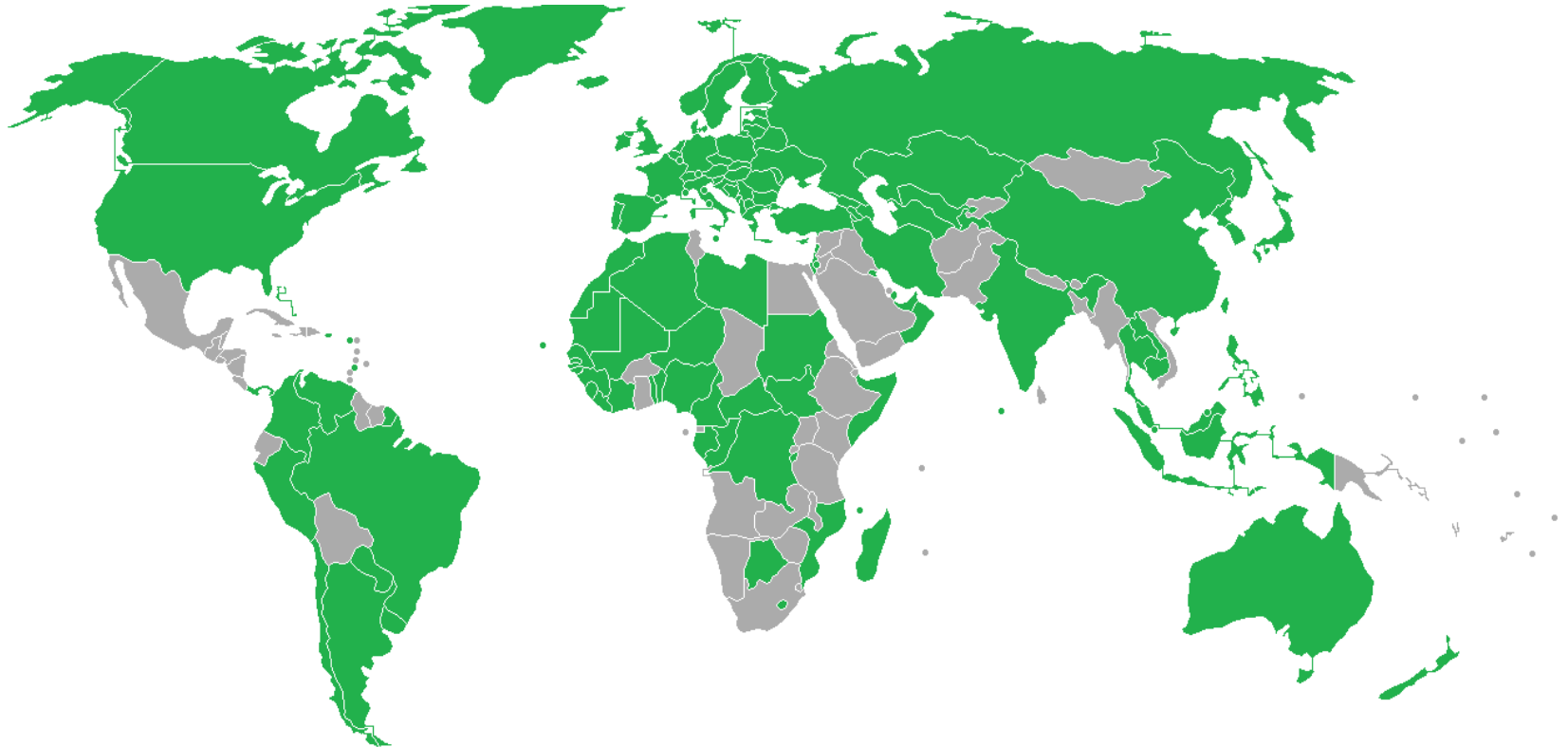
Authenticate Public Keys with Digital Signature

- To authenticate a public key over internet, normally a Certificate Authority (CA) is needed
 - The public key of CA is known to every computer
 - Such as in every web browser (IE, Firefox ...)
 - CA signs a user's public key + the user's information
 - After receiving a public key of a user (together with the signature generated by CA), use CA's public key to verify the signature to check whether the public key belongs to that user

Electronic passport

- Digital Signature is used to protect electronic passport (also called biometric passport)
 - The digital data (passport number, validity period, photo, fingerprint ...) stored in an e-passport is signed by a government
 - The signature is also stored in the e-passport
 - The public key of the government is given to the customs around the world for passport verification
 - Forgery of electronic passport is virtually impossible

Map of countries with electronic passports



Signing Contract

- Two or more parties can sign a contract using their private keys
- Expected to be widely used in the future when many people have the keys for digital signature

Cryptography Currency

- Cryptography currency is based on digital signature (for example, ECDSA is used in Bitcoin)
 - Simply, Alice's digital signature's public key is her wallet address
 - To spend the crypto currency in her wallet, she can use her private key to sign a transaction which contains Alice's wallet address, the receiver's wallet address, and the amount of currency being sent. Later the receiver can use Alice's signature to show that he/she has the currency.
 - For Alice to receive crypto currency from Bob, Bob can use his private key to sign the transaction which contains Bob's wallet address, Alice's wallet address, and the amount of currency being sent from Bob to Alice.
 - To prevent the double spending of crypto currency, all the transactions are recorded on a hash chain (called block chain in crypto currency). Every user is supposed to have a copy of the hash chain. Invalid transactions (invalid signature, or insufficient amount of currency in the account) cannot be recorded into the hash chain.

Cryptography Currency

- Hash chain
 - Suppose that there are many messages $m_1, m_2, m_3, m_4, m_5, m_6, \dots$
 - Apply a hash function to hash these messages:
$$\begin{aligned}MD_1 &= H(m_1) \\MD_2 &= H(MD_1 \parallel m_2) \\MD_3 &= H(MD_2 \parallel m_3) \\&\dots\dots \\MD_i &= H(MD_{i-1} \parallel m_i) \\&\dots\dots\end{aligned}$$
 - The above hash is a hash chain
 - Property of a hash chain: once MD_n is fixed, all the previous messages (m_1, m_2, \dots, m_n) cannot be modified without being detected.

Summary

- Digital Signature
 - Authentication
 - Everyone can verify
 - Schemes
 - RSA signature scheme
 - padding is needed for message digest
 - ElGamal signature scheme
 - Digital Signature Standards
 - Digital Signature Algorithm
 - RSA digital signature algorithm
 - Elliptic curve digital signature algorithm
- Should use different keys for digital signature and public key encryption
- Applications
 - Authenticate digital information: public key, e-passport, contract, crypto currency...