

MAS 433: 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
 - 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
 - Public key being used:
 - 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**
 - 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 practice
 - Digital Signature Algorithm (DSA)
 - RSA Digital Signature Algorithm
 - Elliptic Curve Digital Signature Algorithm (ECDSA)

RSA Signature Scheme

- Public key: (n, e)
- Private key: d
- Signature generation:

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

H : hash function

- Signature verification:

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

Avoid using the same public key and private key
for both signature & encryption

RSA Signature Scheme

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

- *Insecure against multiplicative forgery for typical hash 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
$$H(M) = H(M_i)^{\alpha_i} \times H(M_j)^{\alpha_j} \times \dots \times H(M_x)^{\alpha_x}$$
- *Padding the message digest before signing
 - To learn later

ElGamal Signature Scheme

$$y = g^x \bmod p$$

- Public key: (p, g, y)
- Private key: x
- 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 is: (r, s)

ElGamal Signature Scheme

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

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

- Signature verification
 - $0 < r < p, 0 < s < p-1$
 - $g^{H(m)} \stackrel{?}{=} y^r r^s \pmod{p}$

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 Standard (DSS)

- Digital Signature Standard (FIPS 186-3, 2009)
 - **Digital Signature Algorithm (DSA)**
 - Based on discrete logarithm
 - A variant of ElGamal signature algorithm
 - The size of signature of DSA is smaller than that of ElGamal Signature Scheme
 - RSA Digital Signature Algorithm
 - Elliptic Curve Digital Signature Algorithm (ECDSA)

DSS: Digital Signature Algorithm (DSA)

Key generation (Phase 1):

- Choose an N -bit prime q (example: $N = 256$, $L = 3072$)
- Choose an L -bit prime modulus p such that $p-1$ is a multiple of q
- Choose g whose multiplicative order modulo p is q
 - Set $g = h^{(p-1)/q} \bmod p$ for some integer h , then test whether g is 1 or not.
- The algorithm parameters (p, q, g) may be shared between different users of the system
- key length L and N
 - (1024,160)
 - (2048,224)
 - (2048,256)
 - (3072,256)

DSS: Digital Signature Algorithm (DSA)

Key generation (Phase 2):

- Computes 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

.

DSS: 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)

DSS: Digital Signature Algorithm (DSA)

Signature Verification:

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

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

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

DSS: Digital Signature Algorithm (DSA)

- Hash function being used in DSS
 - SHA-1
 - SHA-2

DSS: *RSA Digital Signature Algorithm

- Message padding is used to resist multiplicative forgery
- Two versions
 - 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

DSS: *RSA Digital Signature Algorithm

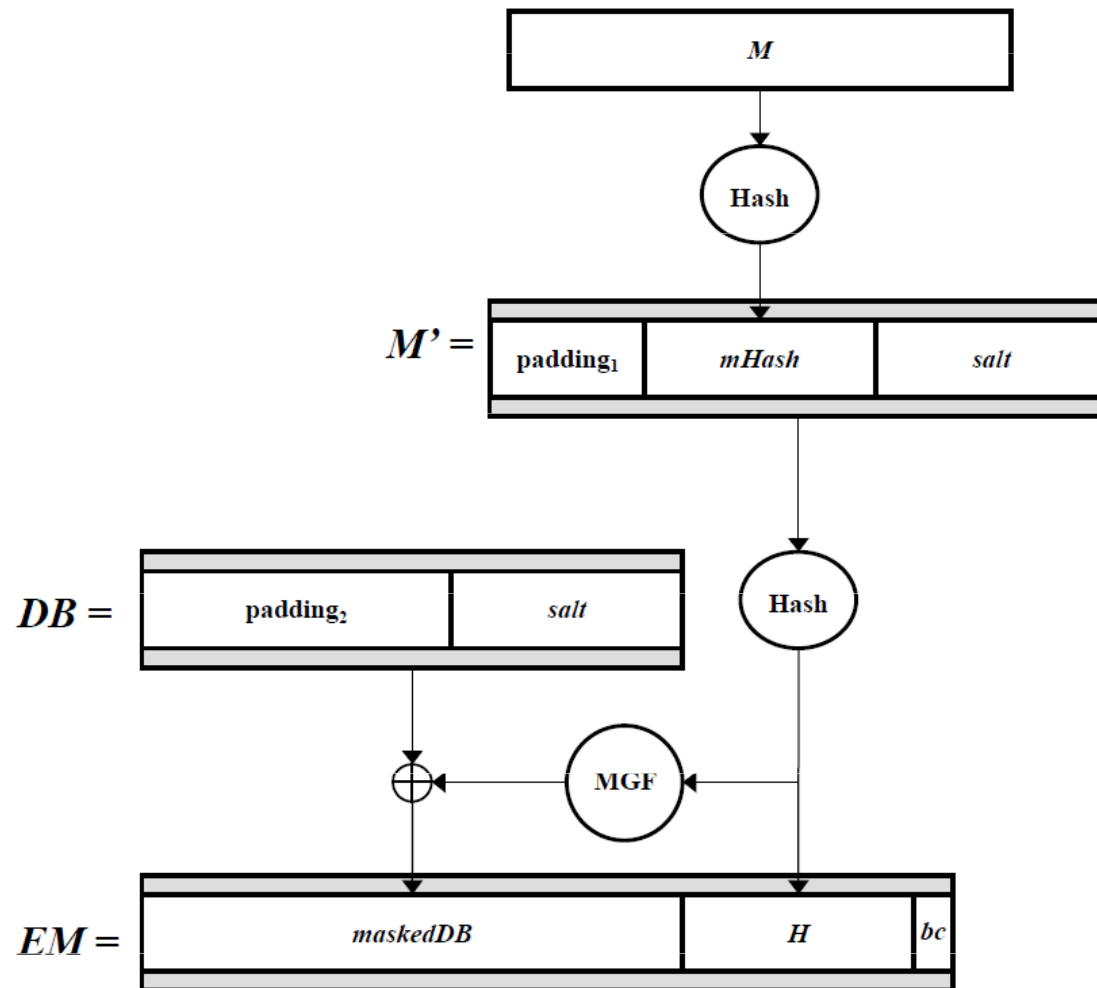
ANSI X9.31

- Message padding:
$$6b \text{ } bb \dots bb \text{ } ba \parallel \text{Hash}(M) \parallel 3x \text{ } cc$$

where $x = 3$ for SHA-1, 1 for RIPEMD-160
- Resistant to multiplicative forgery
- Some moduli are more at risk, but still out of range
- Standardized
 - IEEE P1363, ISO/IEC 14888-3
 - FIPS 186-3

DSS: *RSA Digital Signature Algorithm

- Padding in RSASSA-PSS in PKCS#1 v2.1 (not required for exam)



Digital Signature & Hash Function

- Hash function is necessary for digital signature
 - Main Reason:
 - Fixed message digest size 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
 - Second-preimage resistance
 - Hash function being second-preimage resistance
 - => important for using digital signature for authentication
 - Collision resistance
 - Hash function being collision resistance
 - => important for the non-repudiation property of digital signature

Applications of Digital Signature

- Authenticate public keys
 - The most widely used application of digital signature
 - Important for resisting impersonation attack
- How 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 the web browser (IE, Firefox ...)
 - CA sign 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

Applications of Digital Signature

- Authenticate email
 - PGP

```
-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA1
```

(content)

```
-----BEGIN PGP SIGNATURE-----  
Version: GnuPG/MacGPG2 v2.0.14 (Darwin)  
  
iEYEARECAAYFAkzbeO0ACgkQPntiHUUJ4tFn66wCfZq+6Kj4  
oXjPQK3fSp+NmV1m9YLIAnR60gstcXldFNSmVDhAiA00QnG  
o8=o8bb  
-----END PGP SIGNATURE-----
```

Applications of Digital Signature

- Digital time stamping
 - A trusted party sign a user's document + current time
 - The signature provides the evidence that the document is available at that time

Applications of Digital Signature

- Protect electronic passport
 - The digital data (passport information + personal particulars: photo, fingerprint) stored in an e-passport is signed by the government.
The signature is also stored in the e-passport.
 - The public key of the government is given to (local/foreign) customs for passport verification
 - Forgery of electronic passport is virtually impossible
- Sign contract
 - Expected to be widely used in the future

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
- Use different keys for digital signature and public key encryption
 - RSA
 - ElGamal
- Application
 - Authenticate digital documents (public key, e-passport ...)
 - Signing contract ...