Requirements of Digital Signature

- Efficiency
- Unforgeability : only signer can generate
- Authentication of a signer:
- Not reusable : not to use for other message
- Unalterable : No modification of signed message
- Non-repudiation : not denying the act of signing

Security of Digital Signature(I)

Weak

Strong

- Key only or no message attack : Ad access only to public parameters and public keys
- Message attack : Ad has access to pairs of message texts and corresponding signatures. Depending on Ad' s power of selecting messages signed by Si
  - Known-messages : Ad doesn' t choose message signed by Si.
  - Generic chosen-messages : Ad choose a set of messages to be signed before knowing the actual Si targeted for attack.
  - Directed chosen-message : Ad choose a set of messages to be signed after selecting a specific Si but the actual attack.
  - Adaptive chosen-message : Ad choose message for signing dynamically after inspecting signatures he obtained for previous messages.
- Ad: Adversary, Si: legitimate signer

Security of Digital Signature(II)

Hard

- **Total break : Ad recovers the secret key of Si under attack.**
- **Universal forgery : Ad doesn' t obtain the secret key of Si, but gains the ability to generate valid signatures for <u>any message</u>.**
- **Selective forgery : Ad doesn' t obtain the secret key of Si, but gains the ability to generate valid signatures for any set of <u>preselected</u> messages.**
- **Existential forgery : Ad can create at least one new message and signature pair without knowing the secret key. The messages are only arbitrary bit strings and Ad doesn' t have any power over their composition.**

Easy

Construction of Digital Signature

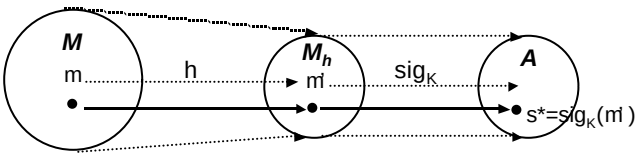
- **Consists of 6 elements *(M,Mh,A,K,S,V)***
  - ✓ ***M* : message space**
  - ✓ ***Mh* (or *Ms)* : signing space**
  - ✓ ***A* : signature space**
  - ✓ ***K* : key space**
  - ✓ **For K Î *K*, $ signing alg. sig_K Î *S* and its corresponding verification alg. ver_K Î *V*.**
  - ✓ **Each sig_K : *M*® *A* and ver_K : *M* x *A* ® {t,f} are fts s.t., $ver_K(x,y)$= t if y = $sig_K(x)$ or $ver_K(x,y)$=f if y ¹ $sig_K(x)$**

Digital signature with appendix(I)

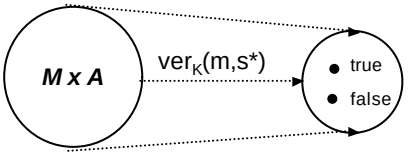
(1) Signature generation
(a) get secret key, K_s
(b) m' =h(m) : hash algorithm and $s^*=sig_{Ks}(m')$
(c) m, s^* : signature
(2) Signature verification
(a) obtain public key, Kp
(b) compute m' =h(m) and $u=ver_{Kp}(m',s^*)$
(c) accept signature iff u=true.
(Ex.) DSA, ElGamal, Schnorr

Digital signature with appendix(II)

(a) signing



(b) verification

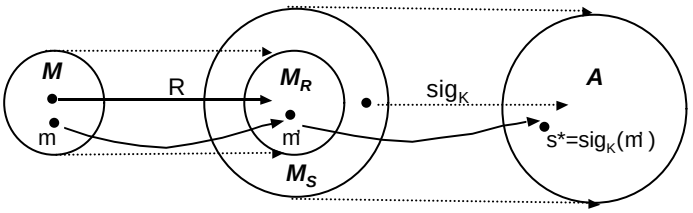


Digital signature with message recovery(I)

(1) Signature generation
(a) get secret key, Ks
(b) m' =R(m) : redundancy ft and $s^*=sig_{Ks}(m')$
(c) s^* : signature
(2) Signature verification
(a) obtain public key K_P
(b) compute $m' = ver_{Kp}(s^*)$
(c) verify that m' Î M_R (if m' Ï M_R, then reject)
(d) recover m from m' by computing $R^{-1}(m')$
(Ex.) RSA, Rabin, Nyberg-Rueppel
*** R() and R^{-1}() are easy to compute.**

Digital signature with message recovery(II)

(a) signing



(b) verification

DIY

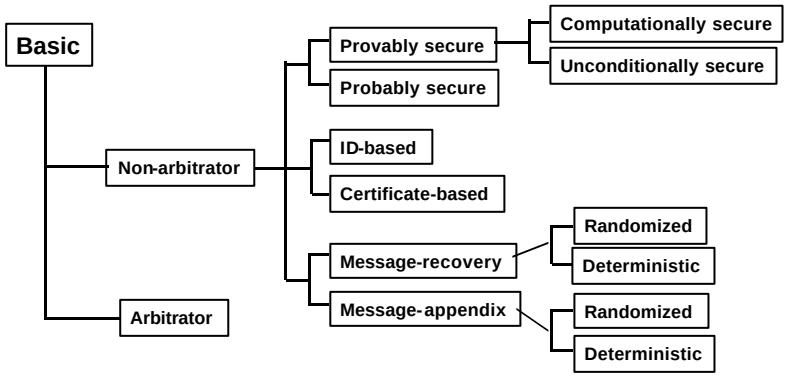
R: redundancy ft
e.g., 1:1 ft
M_R : image of R

*****This scheme can be easily changed to digital signature with appendix s.t., hashing before signing**.**

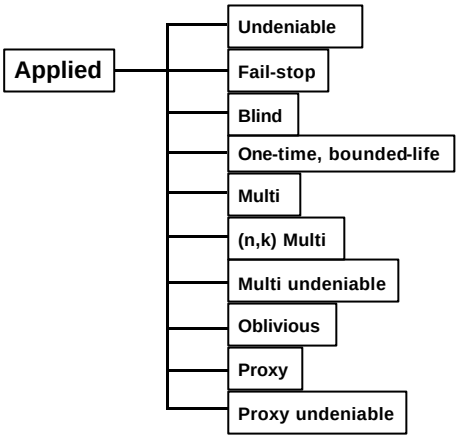
Comparison

Item	Handwritten	Digital
Result of Signature	Fixed	Variable
Digital Copy	Difficult	Easy
Operation	Simple	Mathematical
Legality	Yes	Yes
Forgeability	Possible	Impossible
Tool	Pen	Computer
Auxiliary Tool	Not Necessary	Necessary(Hash ft)

Classification of Digital Signature (I)



Classification of Digital Signature (II)



RSA Signature

- (preparation) n=pq, $M=A=Z_n$
  K ={(n,p,q,a,b) : n=pq, ab=1 mod ϕ(n)}
  public key : {b,n}, secret key : {a,p,q}
- (signing) K=(n,p,q,a,b), $sig_K(x)=x^a \bmod n$
- (verification) $ver_K(x,y)$=true $\Leftrightarrow$ $x=y^b \bmod n$, $x,y \in Z_n$
- (Problem) if an adversary know signature of x_1 and x_2 to be s_1 and s_2, he can create signature of $x_3=x_1x_2$, i.e., $s_3\{=s_1s_2=x_1^d x_2^d=(x_1x_2)^d\}$ without knowing secret key, d. To prevent this, hash(x) must be used.
  * $D(m_1)D(m_2)=D(m_1m_2)$ in RSA

ElGamal Signature(I)

- **p : prime,** $a \in Z_p^*$**: primitive element,**
  $M = Z_p^*$, $A = Z_p^* \times Z_{p-1}^*$
  $K = \{(p, a, a, b) : b = a^a \bmod p\}$
  public: (p, a, b)**, secret :** a
- **(signing)**$K = (p, a, a, b)$**, secret random** $k \in Z_{p-1}^*$
  $sig_K(x,k) = (g,d)$ **where** $g = a^k \bmod p$,
  $d = (x - ag)k^{-1} \bmod (p-1)$
- **(verification)** x, $g \in Z_p^*$ **and** $d \in Z_{p-1}^*$
  $ver_K(x,g,d) =$ **true** $\Leftrightarrow b^g g^d = a^x \bmod p$
  * $b^g g^d = a^{ag} a^{kd} = a^x \bmod p$

ElGamal Signature(II)

- **(Preparation) p=467,** a**=2, a=127**
  $b = a^a \bmod p = 2^{127} \bmod 467 = 132$, $x = 100$
  $k = 213$ $(\gcd(213, 466) = 1,\ 213^{-1} \bmod 466 = 431)$
- **signing**
  $g = 2^{213} \bmod 467 = 29$
  $d = (100 - 127 \times 29)\ 431 \bmod 466 = 51$
- **verification**
  $132^{29}\ 29^{51} = 189 \bmod 467$, $2^{100} = 189 \bmod 467$

ElGamal Signature(III)

- Security : without knowing a, forgery of x' s signature is reducible to DLP of finding d (g) chosen g (d).
- Note
  - Keep k to be secret
  - Not to use k two times.
- Generalization from Z_p^* to any finite Abelian group is possible

DSS (I)

- After 1991 August, for 3-year public debate, NIST announced DSS(Digital Signature Standard, FIPS186) 1994 December.
- Introduce efficient operation under subgroup in ElGamal signature scheme
- Used with DHA.

DSS(II)

- p:512 bit prime, q:160 bit prime, q|p-1,
  g Î Z_p^*, a =$g^{(p-1)/q}$ mod p (q-th root of 1 mod p), *M* =Z_p^*,*A*= Z_q x Z_q,*K*={(p,q,a ,a,b):b=a^a mod p}, public : (p,q,a ,b), secret : a.
- (signing) K=(p,q,a ,a,b), secret random k (1£k£ q-1,gcd(k,q)=1), $sig_K(x,k)$ = (g,d) where g = (a^k mod p) mod q, d=(x+ag)k^{-1} mod q.
- (verification) x Î Z_p^* and g, d Î Z_q
  $ver_K(x,g,d)$ =true Û ($a^{e_1}b^{e_2}$ mod p)mod q = g.
  e_1= xd^{-1} mod q, e_2= gd^{-1} mod q.

DSS(III)

(Ex.) q=101, p=78q+1=7879, g=3, a = 3^{78} mod 7879 =170, a=75, b =a^amod 7879= 4567.

(signing) x=1234, random k=50, k^{-1} mod 101=99.

g=(170^{50}mod 7879) mod101=2518mod101=94.

d=(1234 +75x94) 99 mod101 = 97.

(verification) sig_K(x,k) =(94,97), x=1234,

d^{-1}=97^{-1} mod 101=25, e_1=1234 x 25 mod 101 =45, e_2=94 x 25 mod 101 = 27,

(170^{45} 4567^{27} mod 7879) mod 101 = 2518 mod 101=94 Þ g =94 (valid)

DSS(IV)

- Security: DLP over Z_p^*
- design criteria : secret by NSA
- criticism on fixed p=512 bit Þ multiples of 64 (512 ~1024 bit)

Other Signature Scheme

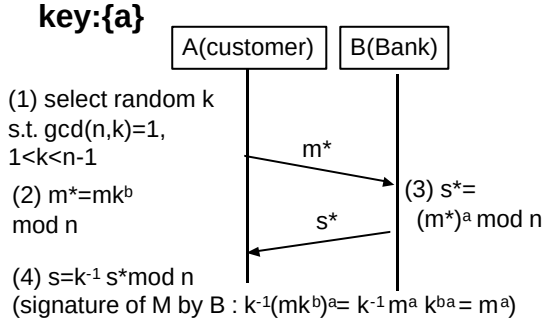
- Schnorr
- ESIGN
- Ong-Schnorr-Shamir
- Nyberg-Rueppel
- Rabin
- KCDSA(Korean Certificate-based Digital Signature Algorithm)

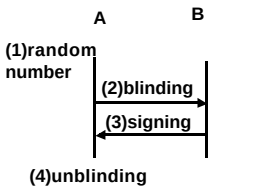
Applied Digital Signature

- **Blind signature**
- **One-time signature**
  - **Lamport scheme**
  - **Bos-Chaum scheme**
- **Undeniable signature**
  - **Chaum-van Antwerpen scheme**
- **Fail-stop signature**
  - **van Heyst-Peterson scheme**
- **Group Signature : group member can generate signature if dispute occurs, identify member.**

Chaum' s Blind Signature(I)

- **Without B' s knowing message M itself, A can get a signature of M from B.**
- **RSA scheme, B' s public key :{n,b}, secret key:{a}**





$g(S_B f(m))=S_B(m)$
f:blinding ft
g:unblinding ft only A knows
f(m) : blinded message

Chaum' s Blind Signature(II)

(Preparation) p=11, q=3, n=33,ϕ(n)= 10 * 2=20
gcd(a, ϕ(n))=1 => a=3, ab =1 mod ϕ(n) => 3 b = 1 mod 20 => b=7
B의 public key :{n,b}={33,7}, secret key ={a}={3}

(1) A' s blinding of m=5
select k s.t. gcd(k,n)=1 => gcd(k,33)=1 => k=2
m* = m k^b mod n= 5 2^7 mod 33=640= 13 mod 33
(2) B' s signing
s*= $(m^*)^a$ mod n = 13^3 mod 33 =2197=19 mod 33
(3) A' s unblinding
s=k^{-1} s* mod n (2 k^{-1}=1 mod 33 => k=17)
= 17 19 mod 33 =323=26 mod 33
* Original Signature : m^a mod n = 5^3 mod 33 =125 =26 mod 33