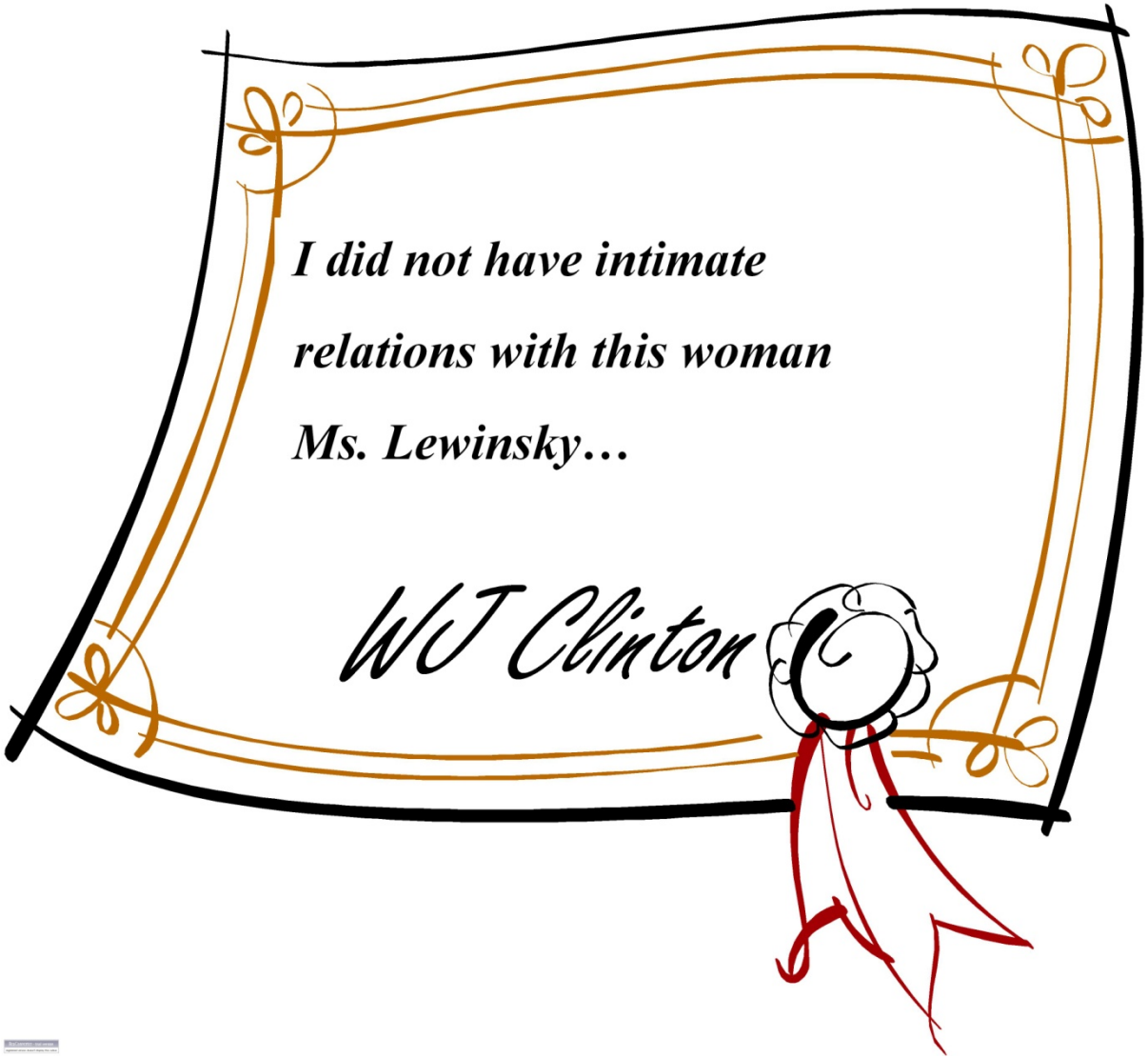
COM 5336
Lecture 9
Digital Signatures

Scott CH Huang

Outline

- Introduction
- RSA signature scheme
- ElGamal signature scheme
- Rabin public-key signature scheme
- DSA signature scheme (optional)
- Rabin one-time signature
- Arbitrated signature schemes

Digital Signature



Definitions

- Digital signature - a data string which associates a message with some originating entity
- Digital signature generation algorithm – a method for producing a digital signature
- Digital signature verification algorithm – a method for verifying whether a digital signature is authentic
- Digital signature scheme - consists of a signature generation algorithm and an associated verification algorithm

Digital Signature

Digital Signatures can provide

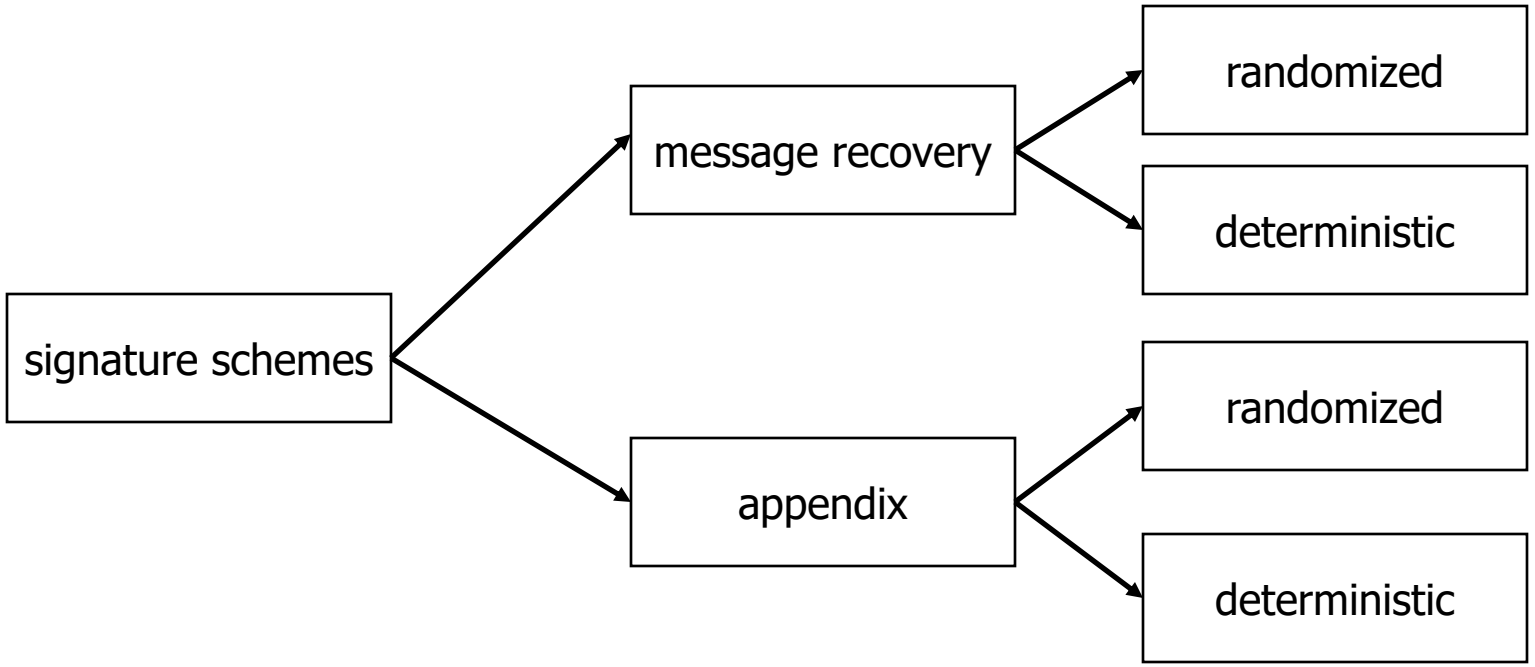
- Authentication
- Data Integrity
- Non-Repudiation
- Timestamping

Desirable Properties

- It should be efficient to compute by the signer
- It should be easy to verify by everybody
- It should be computationally infeasible for an attacker to forge the signature

Types of Digital Signatures

- Taxonomy of digital signatures



Schemes w/ Appendix

- Requires the message as input to verification algorithm
- Rely on cryptographic hash functions rather than customized redundancy functions
- Schemes of this type
  - ElGamal, DSA, Schnorr etc.

Schemes w/ Message Recovery

- The original message is not required as input to the verification algorithm
- The original message can be recovered from the signature itself
- Schemes of this type
  - RSA, Rabin, Nyberg-Rueppel

Breaking a Signature Scheme

- *Total Break*: private key is comprimised
- *Selective forgery*: adversary can create a valid signature on a preselected message
- *Existential forgery*: adversary can create a valid signature with no control over the message

Types of Attacks

- Key-only
  - adversary knows only the public key
- Message attacks
  - *Known-message attack*: adversary has signatures for a set of messages which are known to the adversary but not chosen by him
  - *Chosen-message attack*: adversary obtains valid signatures from a chosen list of his choice (non adaptive)
  - *Adaptive chosen-message attack*: adversary can use the signer as an oracle

RSA Signature Scheme

- Key generation *n, p, q, e, d*
- Sign
  - Compute $m_r = R(m)$, where R is an invertable redundancy function that maps a message into the signature space
  - Compute $s = m_r^d \bmod n$
  - The signature for m is s
- Verify
  - Obtain authentic public key (n, e)
  - Compute $m_r = s^e \bmod n$
  - Verify that $m_r \in M_r$
  - Recover $m = R^{-1}(m_r)$

RSA Signature (cont'd)

- Attacks
  - Integer factorization
  - Homomorphic property
- Reblocking problem
  - If signatures are encrypted different modulus sizes can render the message unrecoverable
- Importance of the redundancy function
  - ISO/IEC 9796

RSA Signature (cont'd)

- Performance (p, q are k-bit primes)
  - Signature $O(k^3)$
  - Verification $O(k^2)$
- Bandwidth
  - Bandwidth is determined by *R*. For example, ISO/IEC 9796 maps *k*-bit messages to 2*k*-bit elements in M_S for a 2*k*-bit signature (efficiency of ½)

ElGamal Signature Scheme

- Key generation: p, α, a, $y=\alpha^a \bmod p$
- Signature Generation
  - Select random k, $1 \leq k \leq p-1$, gcd(k, p-1)=1
  - Compute $r = \alpha^k \bmod p$
  - Compute $k^{-1} \bmod (p-1)$
  - Compute $s = k^{-1} * (h(m) - ar) \bmod (p-1)$, where h is a one-way hash function.
  - Signature is (r,s)

ElGamal Signature (cont'd)

- Signature Verification
  - Verify $1 \leq r \leq p-1$
  - Compute $v_1 = y^r r^s \bmod p$
  - Compute h(m) and $v_2 = \alpha^{h(m)} \bmod p$
  - Accept iff $v_1 = v_2$

$$s \equiv k^{-1}\{h(m) - ar\} \pmod{p-1}$$

$$ks \equiv h(m) - ar \pmod{p-1}$$

$$\alpha^{h(m)} \equiv \alpha^{ar+ks} \equiv (\alpha^a)^r r^s \pmod{p}$$

ElGamal Signature (cont'd)

- Security (based on DLP)
  - Index-calculus attack: p should be large
  - Pohlig-Hellman attack: p-1 should not be smooth
  - *Weak generators*: If $p \equiv 1 \bmod 4$, $\alpha | p-1$, DL can be broken for subgroup S of order α. Forgeries are then possible

ElGamal Signature (cont'd)

- In addition…
  - k must be unique for each message signed
    - $(s_1-s_2)k=(h(m_1)-h(m_2))\bmod (p-1)$
  - An existential forgery attack can be mounted if a hash function is not used

ElGamal Signature (cont'd)

- Performance
  - Signature Generation
    - One modular exponentiation
    - One Euclidean Algorithm
    - Both can be done offline
  - Verification
    - Three modular exponentiations
- Generalized ElGamal Signatures

Rabin Public-Key Signature Scheme

- Signature: $s = m^{1/2} \bmod n$
  - Using any sqrt
- Verification: $m = s^2 \bmod n$
- About ¼ of the messages have sqrts.
- If a message doesn't have a sqrt, then it has to be slightly modified by doing proper padding.

Rabin Signature (cont'd)

- In practice, one might try to append a small amount of random bits and *hope* that it has a sqrt.
- On average, two such attempts would be enough.
- Such method has no guarantee of success, so a deterministic method would be preferable.

Modified Rabin Signature Scheme

- To overcome this problem, a modified Rabin signature scheme is designed.
- It provides a deterministic method for associating messages with elements in the signing space such that computing s square root (or something close to it) is always possible.
- Details of modified Rabin signature scheme is beyond the scope of this course.

Rabin Signature Pros & Cons

- Advantage:
  - Verification is extremely fast
  - Cracking the signature scheme is provable as hard as doing factorization.
- Disadvantage: like Rabin encryption, one system can only generate signature for a single user.

DSA Signature

- DSA Algorithm : key generation
  1. select a prime q of 160 bits
  2. Choose $0 \le t \le 8$, select $2^{511+64t} < p < 2^{512+64t}$ with $q \mid p-1$
  3. Select g in Z_p^*, and $\alpha = g^{(p-1)/q} \bmod p$, $\alpha \neq 1$ ($ord(\alpha)=q$)
  4. Select $1 \le a \le q-1$, compute $y = \alpha^a \bmod p$
  5. public key (p, q, α, y), private key a

DSA Signature (cont'd)

- DSA signature generation
  - Select a random integer k, $0 < k < q$
  - Compute $r=(\alpha^k \bmod p) \bmod q$
  - Compute $k^{-1} \bmod q$
  - Compute $s=k^{-1} * (h(m) + ar) \bmod q$
  - Signature = (r, s)

DSA Signature (cont'd)

- DSA signature verification
  - Verify 0<r<q and 0<s<q, if not, invalid
  - Compute w= s^{-1}mod q
  - Compute u_1=w∗h(m) mod q, u_2=r∗w mod q
  - Compute v = ($\alpha^{u_1} y^{u_2}$ mod p) mod q
  - Valid iff v=r

$$h(m) \equiv -ar + ks \pmod{q}$$
$$wh(m) + arw \equiv k \pmod{q}$$
$$u_1 + au_2 \equiv k \pmod{q}$$
$$\alpha^{u_1} y^{u_2} \bmod p \pmod{q} = \alpha^{k} \bmod p \pmod{q}$$

DSA Signature (cont'd)

- Security of DSA
  - two distinct DL problems: Z_P^*, cyclic subgroup order q
- Parameters:
  - q~160bits, p 768~1Kb, p,q, α can be system wide
- Probability of failure
  - Pr[s=0]= $(1/2)^{160}$

DSA Signature (cont'd)

- Performance
  - Signature Generation
    - One modular exponentiation
    - Several 160-bit operations (if p is 768 bits)
    - The exponentiation can be pre-computed
  - Verification
    - Two modular exponentiations

Nyberg-Rueppel Signature Scheme

- Can be regarded as an ElGamal signature scheme w/ message recovery
- Key generation:
  - Same as DSA but no constraints on the sizes of *p,q*
  - Select g in Z_p^*, and $\alpha = g^{(p-1)/q} \bmod p$, $\alpha \neq 1$ ($ord(\alpha)=q$)
  - Select, p,q w/ $q|(p-1\}$, $\alpha \in Z_p^*$, (private key) a, $y= \alpha^a$.

Nyberg-Rueppel Signature Scheme

- Sign message m
  - Compute $m_r = R(m)$, where R is an invertable redundancy function
  - Select a random secret integer k s.t. $1 \leq k \leq p-1$
  - Compute $r \equiv \alpha^{-k} \pmod{p}$, $e \equiv m_r\, r \pmod{p}$, and $s \equiv ae + k \pmod{q}$
  - Signature is (e,s)
- Verify signature (e,s)
  - Obtain authentic public key (p,q,α,y)
  - Verify that $0 < e < p$ and $0 \leq s \leq q$. If not, reject.
  - Compute $v \equiv \alpha^{s} y^{-e} \pmod{p}$ and $m \equiv ve \pmod{p}$
  - Recover $m = R^{-1}(m_r)$