Cryptography and Network Security Chapter 13

Fifth Edition
by William Stallings
Lecture slides by Lawrie Brown
(with edits by RHB)

Chapter 13 – Digital Signatures

To guard against the baneful influence exerted by strangers is therefore an elementary dictate of savage prudence. Hence before strangers are allowed to enter a district, or at least before they are permitted to mingle freely with the inhabitants, certain ceremonies are often performed by the natives of the country for the purpose of disarming the strangers of their magical powers, or of disinfecting, so to speak, the tainted atmosphere by which they are supposed to be surrounded.

—*The Golden Bough*, Sir James George Frazer

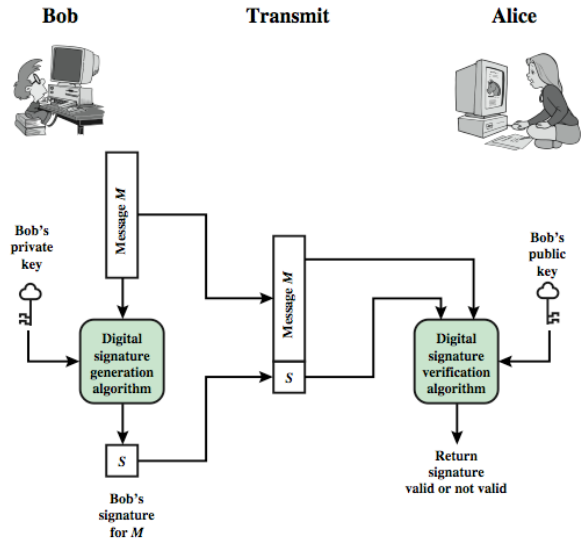
Outline

- will discuss:
  - digital signatures
  - ElGamal and Schnorr signature schemes
  - digital signature algorithm and standard

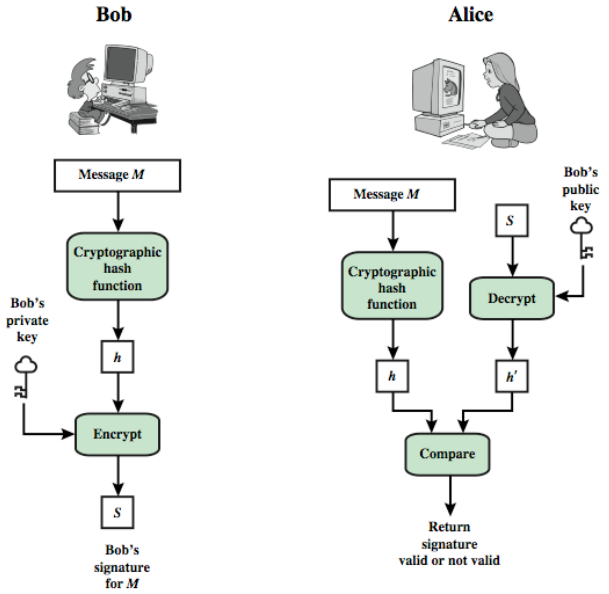
Digital Signatures

- have looked at message authentication
  - but does not address issues of lack of trust
- digital signatures provide the ability to:
  - verify author, date and time of signature
  - authenticate message contents
  - be verified by third parties to resolve disputes
- hence include authentication function with additional capabilities

Digital Signature Model



Digital Signature Model



Attacks and Forgeries

- attacks
  - (public) key-only attack
  - known message attack
  - generic chosen message attack
  - directed chosen message attack
  - adaptive chosen message attack
- break success levels
  - total break
  - selective forgery
  - existential forgery

Digital Signature Requirements

- must depend on the message signed
- must use information unique to sender
  - to prevent both forgery and denial
- must be relatively easy to produce
- must be relatively easy to recognize and verify
- be computationally infeasible to forge
  - with new message for existing digital signature
  - with fraudulent digital signature for given message
- be practical to save digital signature in storage

Direct Digital Signatures

- involves only sender and receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can encrypt using receivers public-key
- important that sign first then encrypt message and signature
- security depends on sender's private-key
- RSA is an obvious candidate … efficiency

Digital Signature Characteristics

- a public key scheme … TWO key pairs:
- a (long-time, permanent) durable private/public key pair
- a (nonce-like, one-time, per-message) disposable private/public key pair
- both key pairs generated by SENDER
- signature is two numbers, depending on message hash and secret info
- a verification calculation succeeds iff the two numbers correctly depend on the secret info
- disposable private/public key pair makes a collection of signatures of the sender uncorrelated, so hard to break, analytically or statistically

ElGamal Digital Signatures

- signature variant of ElGamal, related to D-H
  - so uses exponentiation in a finite Galois field
  - security based difficulty of computing discrete logarithms, as in D-H
- use private key for encryption (signing)
- uses public key for decryption (verification)
- each user (eg. A) generates their key
  - chooses a secret key: $1 < x_A < q-1$
  - compute their **public key**: $y_A = a^{x_A} \bmod q$

ElGamal Digital Signature

- Alice signs a message M to Bob by computing
  - the hash: $m = H(M)$, $0 <= m <= (q-1)$
  - choose one-time secret random integer K with $1 < K < (q-1)$ and $GCD(K, q-1) = 1$
  - compute temporary key: $S_1 = a^K \bmod q$
  - compute K^{-1} the inverse of $K \bmod (q-1)$
  - compute the value: $S_2 = K^{-1}(m - x_A S_1) \bmod (q-1)$
  - signature is: (S_1, S_2)
- any user B can verify the signature by computing
  - $V_1 = a^m \bmod q$
  - $V_2 = y_A^{S_1} S_1^{S_2} \bmod q$
  - signature is valid if $V_1 = V_2$

ElGamal Digital Signature ... verification.

Public info: $q,\ \ a,\ \ H(M) = m,\ \ y_A = a^{x_A} \bmod q,$
$S_1 = a^K \bmod q,\ \ S_2 = K^{-1}(m - x_A S_1) \bmod (q-1)$

Secret info: $x_A,\ \ K,\ \ K^{-1} \bmod (q-1)$
(N.B. K^{-1} exists since $K, q-1$ coprime)

Verify the signature by testing $V_1 \stackrel{?}{=} V_2$ where

$V_1 = a^m \bmod q$

$V_2 = y_A{}^{S_1} S_1{}^{S_2} \bmod q$

$$
\begin{aligned}
V_2 &= y_A{}^{S_1} S_1{}^{S_2} \bmod q \\
&= (a^{x_A} + Z_1 q)^{S_1} S_1{}^{S_2} \bmod q \\
&= a^{x_A S_1} S_1{}^{S_2} \bmod q \\
&= a^{x_A S_1} (a^K + Z_2 q)^{S_2} \bmod q \\
&= a^{x_A S_1} a^{K S_2} \bmod q \\
&= a^{x_A S_1} a^{K[K^{-1}(m - x_A S_1) + Z_3(q-1)]} \bmod q \\
&= a^{x_A S_1} a^{(m - x_A S_1) + Z_4(q-1)} \bmod q \\
&= a^m a^{Z_5(q-1)} \bmod q \\
&= a^m \bmod q \\
&= V_1
\end{aligned}
$$

ElGamal Signature Example

- use field GF(19) $q = 19$ and $a = 10$
- Alice computes her key:
  - A chooses $x_A = 16$ and computes $y_A = 10^{16} \bmod 19 = 4$
- Alice signs message with hash $m = 14$ as $(3, 4)$
  - choosing random $K = 5$ which has $\gcd(18, 5) = 1$
  - computing $S_1 = 10^5 \bmod 19 = 3$
  - finding $K^{-1} \bmod (q-1) = 5^{-1} \bmod 18 = 11$
  - computing $S_2 = 11(14 - 16.3) \bmod 18 = 4$
- any user B can verify the signature by computing
  - $V_1 = 10^{14} \bmod 19 = 16$
  - $V_2 = 4^3.3^4 = 5184 = 16 \bmod 19$
  - since $16 = 16$ signature is valid

Schnorr Digital Signatures

- uses exponentiation in a finite Galois field
  - security based on discrete logarithms, as in D-H
- minimizes message dependent computation
  - multiplying a $2n$-bit integer with an n-bit integer
- main work can be done in idle time
- use a prime modulus p such that
  - $p - 1$ has a prime factor q of appropriate size
  - typically p 1024-bit and q 160-bit numbers

Schnorr Key Setup

- choose suitable primes p, q
- choose a such that $a^q = 1 \bmod p$
- (a, p, q) are global parameters for all
- each user (eg. A) generates a key
  - chooses a secret key: $0 < s_A < q$
  - computes their **public key**: $v_A = a^{-s_A} \bmod p$

Schnorr Signature

- user signs message by
  - choosing random r with $0 < r < q$ and computing $x = a^r \bmod p$
  - concatenate message with x and hash result to compute: $e = H(M \,||\, x)$
  - compute: $y = (r + s_A e) \bmod q$
  - signature is pair (e, y)
- any other user can verify the signature as follows:
  - compute: $x' = a^y v_A^e \bmod p$
  - verify that: $e = H(M \,||\, x')$

Schnorr Digital Signature ... verification.

Public info: p, q, a ... (with $a^q = 1 \bmod p$), H, $v_A = a^{-s_A} \bmod p$, $e = H(M \,||\, x)$, $x = a^r \bmod p$, $y = (r + s_A e) \bmod q$

Secret info: s_A, r

Verify the signature by testing $e \stackrel{?}{=} H(M \,||\, x')$ where

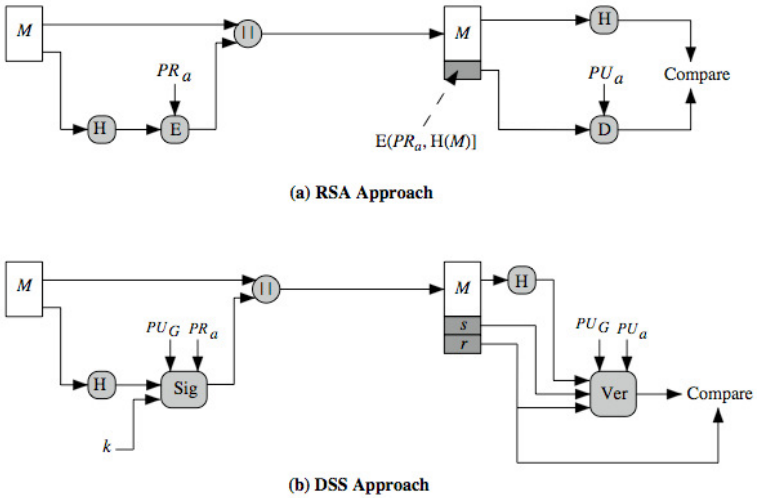
$x' = a^y v_A^e \bmod p$

$$
\begin{aligned}
x' &= a^y v_A^e \bmod p \\
&= a^{(r + s_A e + Z_1 q)} (a^{-s_A} + Z_2 p)^e \bmod p \\
&= a^{(r + s_A e + Z_1 q)} a^{-s_A e} \bmod p \\
&= a^{r + Z_3 q} \bmod p \\
&= a^r \bmod p \\
&= x
\end{aligned}
$$

Digital Signature Standard (DSS)

- US Govt approved signature scheme
- designed by NIST & NSA in early 90's
- published as FIPS-186 in 1991
- revised in 1993, 1996, 2000
- uses the SHA hash algorithm
- DSS is the standard, DSA is the algorithm
- FIPS 186-2 (2000) includes alternative RSA and elliptic curve signature variants
- DSA is a digital signature only, unlike RSA
- is a public-key technique

DSS vs RSA Signatures



Digital Signature Algorithm (DSA)

- creates a 320 bit signature
- with 512-1024 bit security
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms
- variant of ElGamal and Schnorr schemes

DSA Key Generation

- have shared global public key values (p,q,g):
  - choose 160-bit prime number q
  - choose a large prime p with $2^{L-1} < p < 2^L$
    - where $L = 512 \ldots 1024$ bits and is a multiple of 64
    - such that q is a 160 bit prime divisor of $(p - 1)$
  - choose h and find $g = h^{(p-1)/q} \bmod p$
    - where $1 < h < p - 1$ and $h^{(p-1)/q} \bmod p > 1$
- users choose private, and compute public keys:
  - choose random private key: $x < q$
  - compute public key: $y = g^x \bmod p$

DSA Signature Creation

- to **sign** a message `M` the sender:
  - generates a random signature key `k , k < q`
  - nb. `k` must be random, be destroyed after use, and never be reused
- then computes signature pair:

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

  $s = [k^{-1}(H(M) + xr)] \bmod q$
- sends signature `(r,s)` with message `M`

DSA Signature Verification

- having received `M` and signature `(r,s)`
- to **verify** a signature, recipient computes:

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

  $u1 = [H(M)w] \bmod q$

  $u2 = (rw) \bmod q$

  $v = [(g^{u1} y^{u2}) \bmod p] \bmod q$
- if $v = r$ then signature is confirmed

DSA ... verification.

Public info: p, q ... (with $q \mid p-1$),
g ... (with $g = h^{(p-1)/q} > 1 \bmod p$ and $1 < h < p-1$),
H, $y = g^x \bmod p$,
$r = (g^k \bmod p) \bmod q$, $s = k^{-1}(H(M) + xr) \bmod q$

Secret info: x ... (with $0 < x < q$), k ... (with $0 < k < q$), $k^{-1} \bmod q$
(N.B. k^{-1} exists since k, q coprime)

Verify the signature by testing $v \stackrel{?}{=} r$ where

$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$ $\quad$ $w = s^{-1} \bmod q$

$u_1 = [H(M)w] \bmod q$ $\quad$ $u_2 = rw \bmod q$

$$
\begin{aligned}
v &= [(g^{u_1} y^{u_2}) \bmod p] \bmod q \\
&= [(g^{(H(M)w + Z_1 q)} y^{(rw + Z_2 q)}) \bmod p] \bmod q \\
&= [(g^{(H(M)w + Z_1 q)} (g^x + Z_3 p)^{(rw + Z_2 q)}) \bmod p] \bmod q \\
&= [(g^{(H(M)w + Z_1 q)} g^{x(rw + Z_2 q)}) \bmod p] \bmod q \\
&= [(g^{((H(M) + xr)w + Z_4 q)}) \bmod p] \bmod q \\
&= [(g^{((H(M) + xr)s^{-1} + Z_5 q)}) \bmod p] \bmod q \\
\\
&= [(g^{((H(M) + xr)[k^{-1}(H(M) + xr)]^{-1} + Z_6 q)}) \bmod p] \bmod q \\
&= [(g^{((H(M) + xr)k(H(M) + xr)^{-1} + Z_7 q)}) \bmod p] \bmod q \\
&= [(g^{k + Z_8 q}) \bmod p] \bmod q \\
&= [g^k g^{Z_9 q} \bmod p] \bmod q \\
&= [g^k (h^{(p-1)/q} + Z_{10} p)^{Z_9 q} \bmod p] \bmod q
\end{aligned}
$$

$$
\begin{aligned}
&= [g^k h^{[(p-1)/q]Z_{11}q} \bmod p] \bmod q \\
&= [g^k h^{[(p-1)]Z_{11}} \bmod p] \bmod q \\
&= [g^k 1^{Z_{11}} \bmod p] \bmod q \\
&= [g^k \bmod p] \bmod q \\
&= r
\end{aligned}
$$

Global Public Key Components

p prime number where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L a multiple of 64 i.e., bit length of between 512 and 1024 bits in increments of 64 bits

q prime divisor of $(p-1)$, where $2^{159} < q < 2^{160}$ i.e., bit length of 160 bits

g $= h^{(p-1)/q} \bmod p$ where h is any integer with $1 < h < (p-1)$ such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

y $= g^x \bmod p$

User's Per-Message Secret Number

k = random or pseudorandom integer with $0 < k < q$

Signing

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

$s = \left[k^{-1}(\mathrm{H}(M) + xr)\right] \bmod q$

Signature = (r, s)

Verifying

$w = (s')^{-1} \bmod q$

$u1 = \left[\mathrm{H}(M')w\right] \bmod q$

$u_2 = (r')w \bmod q$

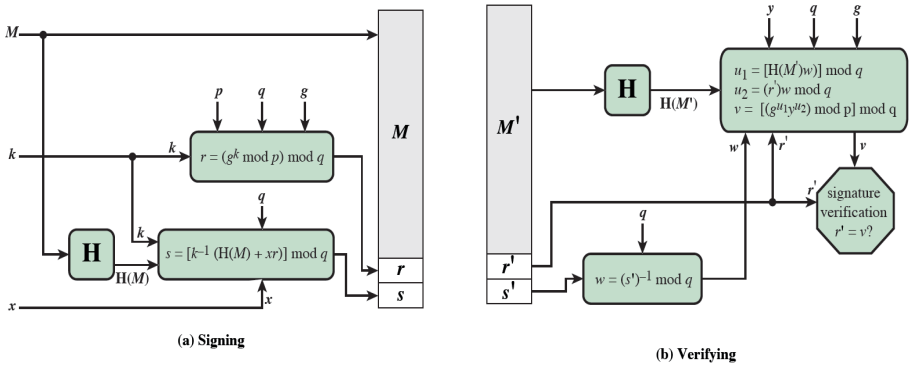
$v = \left[\left(g^{u1} y^{u2}\right) \bmod p\right] \bmod q$

TEST: $v = r'$

M = message to be signed
$\mathrm{H}(M)$ = hash of M using SHA-1
M', r', s' = received versions of M, r, s

Figure 13.4 The Digital Signature Algorithm (DSS)

DSS Overview



(a) Signing

(b) Verifying