Week 9

Lecture 1

Polynomial Rings

Lecture 2

ElGamal Encryption

Workshop 9: Workshop based on Lectures in Week 8

Quiz 9

Udaya Parampalli

Additional Material

COMP90043
Lecture 2

Udaya Parampalli

Schnorr Digital Signatures

- Uses exponentiation in a finite (Galois)
  - Security based on discrete logarithms, as in D-H
- Minimizes message dependent computation
  - multiplying a 2*n-bit* integer with an *n-bit* integer
- Main work can be done in idle time
- Have using a prime modulus *p*
  - $p-1$ 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 (number): $0 < s_A < q$
  - compute their **public key**: $v_A = a^{-s_A} \bmod q$

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 (number): $0 < s < q$
  - compute their **public key**: $v = a^{-s} \bmod p$

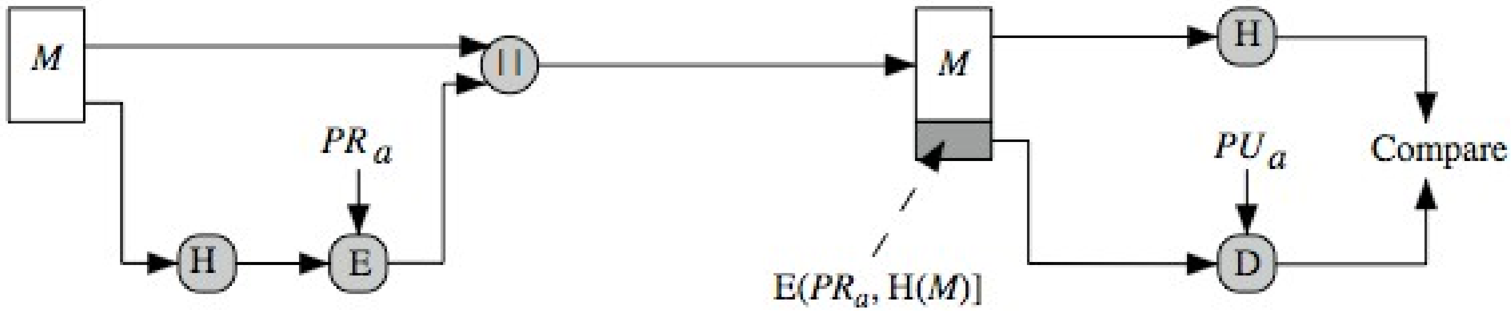
Schnorr Signature

- 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 computing: $e = H(M \| x)$
  - computing: $y = (r + se) \bmod q$
  - signature is pair (e, y)
- any other user can verify the signature as follows:
  - computing: $x' = a^y v^e \bmod p$
  - verifying that: $e = H(M \| x')$

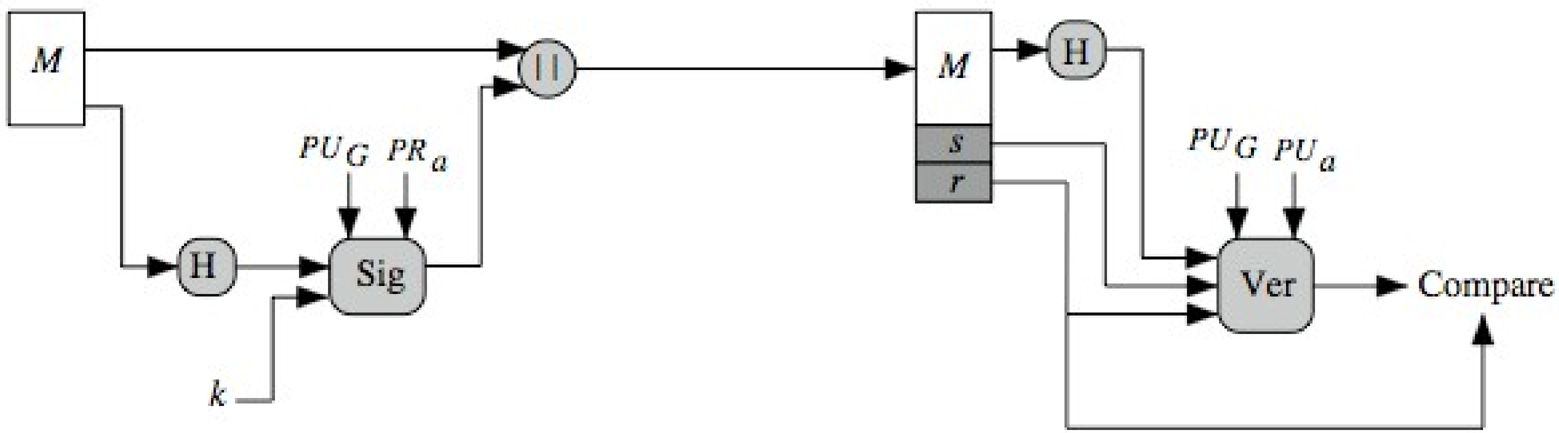
Digital Signature Standard (DSS)

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

DSS vs RSA Signatures



(a) RSA Approach



(b) DSS Approach

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
- It is a variant of ElGamal & Schnorr schemes

Main Idea

- Works in subgroup of a larger finite field.
- Works over a large finite field Z_p. p: 1000 bits long.
- Maximum size of the cyclic group = p-1.
- We will ensure that p-1 has a large prime factor q (160 bit long). Hence q divides (p-1).
- We will choose a generator of the subgroup (g).
- Then $g^{(q)} = 1 \bmod p$.
- Now we can redefine ElGamal idea over the subgroup:
  - Signing equations involve modulo q
  - Verifications are over mod p;
- DSA follows a similar strategy with some modifications.

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 to 1024 bits and is a multiple of 64
    - such that q is a 160 bit prime divisor of $(p-1)$
  - choose $g = h^{(p-1)/q}$
    - where $1<h<p-1$ and $h^{(p-1)/q} \bmod p > 1$
- users choose private & compute public key:
  - 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
- to **sign** 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 & signature `(r,s)`
- to **verify** a signature, recipient computes:

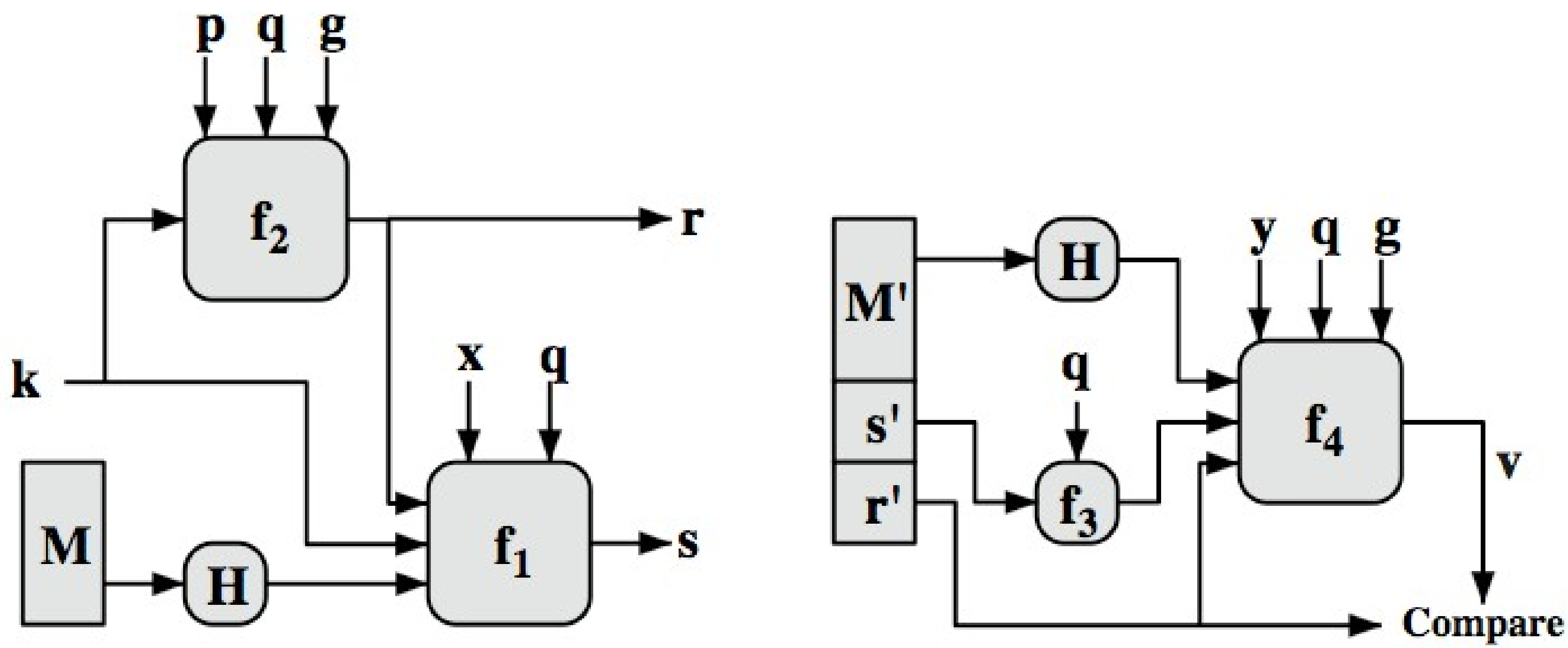
  $w = s^{-1} \bmod q$

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

  $u2= (rw)\bmod q$

  $v = [(g^{u1} \, y^{u2})\bmod p \,]\bmod q$
- if `v=r` then signature is verified
- Appendix A of Chapter 13 for details of proof why

DSS Overview



$$s = f_1(H(M), k, x, r, q) = (k^{-1}\,(H(M) + xr)) \bmod q$$

$$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$$

(a) Signing

$$w = f_3(s', q) = (s')^{-1} \bmod q$$

$$v = f_4(y, q, g, H(M'), w, r')$$
$$= ((g^{(H(M')w) \bmod q}\, y^{r'w \bmod q}) \bmod p) \bmod q$$

(b) Verifying