

# Cryptography

## Course 3: discrete-log based cryptography

Jean-Sébastien Coron

Université du Luxembourg

March 17, 2010

# Discrete-log based cryptography

- Discrete-log based group
  - The multiplicative group  $\mathbb{Z}_p$
- Discrete-log based encryption
  - ElGamal
  - Cramer-Shoup
- Discrete-log based key-exchange
  - Diffie-Hellman
  - MQV, HMQV
- Discrete-log based signature
  - Schnorr signatures.

# The multiplicative group $\mathbb{Z}_p^*$

- Let  $p$  be a prime integer.
  - The set  $\mathbb{Z}_p^*$  is the set of integers modulo  $p$  which are invertible modulo  $p$ .
  - The set  $\mathbb{Z}_p^*$  is a cyclic group of order  $p - 1$  for the operation of multiplication modulo  $p$ .
- Generators of  $\mathbb{Z}_p^*$  :
  - There exists  $g \in \mathbb{Z}_p^*$  such that any  $h \in \mathbb{Z}_p^*$  can be uniquely written as  $h = g^x \pmod p$  with  $0 \leq x < p - 1$ .
  - The integer  $x$  is called the *discrete logarithm* of  $h$  to the base  $g$ , and denoted  $\log_g h$ .

# Finding a generator of $\mathbb{Z}_p^*$

- Finding a generator of  $\mathbb{Z}_p^*$  for prime  $p$ .
  - The factorization of  $p - 1$  is needed. Otherwise, no efficient algorithm is known.
  - Factoring is hard, but it is possible to generate  $p$  such that the factorization of  $p - 1$  is known.
- Generator of  $\mathbb{Z}_p^*$ 
  - $g \in \mathbb{Z}_p^*$  is a generator of  $\mathbb{Z}_p^*$  if and only if  $g^{(p-1)/q} \neq 1 \pmod p$  for each prime factor  $q$  of  $p - 1$ .
  - There are  $\phi(p - 1)$  generators of  $\mathbb{Z}_p^*$

# Generating $p$ and $q$

- Generate  $p$  such that  $p - 1 = 2 \cdot q$  for some prime  $q$ .
  - Generate a random prime  $p$ .
  - Test if  $q = (p - 1)/2$  is prime. Otherwise, generate another  $p$ .
- Finding a generator  $g$  for  $\mathbb{Z}_p^*$ 
  - Generate a random  $g \in \mathbb{Z}_p^*$  with  $g \neq \pm 1$
  - Check that  $g^q \neq 1 \pmod p$ . Otherwise, generate another  $g$ .
  - Complexity :
    - There are  $\phi(p - 1) = q - 1$  generators.
    - $g$  is a generator with probability  $\simeq 1/2$ .

- Discrete logarithm problem :
  - Given  $g, h$  such that  $h = g^x$  for  $x \stackrel{R}{\leftarrow} \mathbb{Z}_{p-1}$ , find  $x$ .
- Computing discrete logarithms in  $\mathbb{Z}_p^*$ 
  - Hard problem: no efficient algorithm is known for large  $p$ .
  - Brute force: enumerate all possible  $x$ . Complexity  $\mathcal{O}(p)$ .
  - Baby step/giant step method: complexity  $\mathcal{O}(\sqrt{p})$ .

- We want to work in a prime-order subgroup of  $\mathbb{Z}_p^*$ 
  - Generate  $p, q$  such that  $p - 1 = 2 \cdot q$  and  $p, q$  are prime
  - Find a generator  $g$  of  $\mathbb{Z}_p^*$
  - Then  $g' = g^2 \pmod p$  is a generator of a subgroup  $G$  of  $\mathbb{Z}_p^*$  of prime order  $q$ .

- Key generation
  - Let  $G$  be a subgroup of  $\mathbb{Z}_p^*$  of prime order  $q$  and  $g$  a generator of  $G$ .
  - Let  $x \xleftarrow{R} \mathbb{Z}_q$ . Let  $h = g^x \pmod p$ .
  - Public-key :  $(g, h)$ . Private-key :  $x$
- Encryption of  $m \in G$  :
  - Let  $r \xleftarrow{R} \mathbb{Z}_q$
  - Output  $c = (g^r, h^r \cdot m)$
- Decryption of  $c = (c_1, c_2)$ 
  - Output  $m = c_2 / (c_1^x) \pmod p$

- To recover  $m$  from  $(g^r, h^r \cdot m)$ 
  - One must find  $h^r$  from  $(g, g^r, h = g^x)$
- Computational Diffie-Hellmann problem (CDH) :
  - Given  $(g, g^a, g^b)$ , find  $g^{ab}$
  - No efficient algorithm is known.
  - Best algorithm is finding the discrete-log

# Chosen-ciphertext attack

- El-Gamal is not chosen-ciphertext secure
  - Given  $c = (g^r, h^r \cdot m)$  where  $pk = (g, h)$
  - Ask for the decryption of  $c' = (g^{r+1}, h^{r+1} \cdot m)$  and recover  $m$ .
- The Cramer-Shoup encryption scheme (1998)
  - Can be seen as extension of El-Gamal.
  - Chosen-ciphertext secure (IND-CCA) without random oracle.

# The Cramer-Shoup cryptosystem

- Key generation
  - Let  $G$  a group of prime order  $q$
  - Generate random  $g_1, g_2 \in G$  and randoms  $x_1, x_2, y_1, y_2, z \in \mathbb{Z}_q$
  - Let  $c = g_1^{x_1} g_2^{x_2}$ ,  $d = g_1^{y_1} g_2^{y_2}$ ,  $h = g_1^z$
  - Let  $H$  be a hash function
  - $pk = (g_1, g_2, c, d, h, H)$  and  $sk = (x_1, x_2, y_1, y_2, z)$
- Encryption of  $m \in G$ 
  - Generate a random  $r \in \mathbb{Z}_q$
  - $C = (g_1^r, g_2^r, h^r m, c^r d^{r\alpha})$
  - where  $\alpha = H(g_1^r, g_2^r, h^r m)$

# The Cramer-Shoup cryptosystem

- Decryption of  $C = (u_1, u_2, e, v)$ 
  - Compute  $\alpha = H(u_1, u_2, v)$  and test if :

$$u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} = v$$

- Output “reject” if the condition does not hold.
- Otherwise, output :

$$m = e/(u_1)^z$$

- INC-CCA security
  - Cramer-Shoup is secure secure against adaptive chosen ciphertext attack
  - without the random oracle model assumption

# Diffie-Hellman protocol

- Diffie-Hellman key exchange
  - Enables Alice and Bob to establish a shared secret key
  - without having talked to each other before.
- Key generation
  - Let  $p$  a prime integer and  $G$  a subgroup of  $\mathbb{Z}_p^*$  of order  $q$  and generator  $g$ .
  - Alice generates  $x \xleftarrow{R} G$  and publishes  $X = g^x \pmod p$ . She keeps  $x$  secret.
  - Bob generates a random  $y \xleftarrow{R} G$  and publishes  $Y = g^y \pmod p$ . He keeps  $y$  secret.

- Key establishment

- Alice sends  $X$  to Bob. Bob sends  $Y$  to Alice.
- Alice computes  $K_a = Y^x \pmod p$
- Bob computes  $K_b = X^y \pmod p$

$$K_a = Y^x = (g^y)^x = g^{xy} = (g^x)^y = X^y = K_b$$

- Alice and Bob now share the same key

- $K = K_a = K_b$
- $K$  can be used as a session key to symmetrically encrypt data.

# Security of Diffie-Hellman

- Computational Diffie-Hellman problem (CDH) :
  - Given  $(g, g^a, g^b)$ , find  $g^{ab}$
  - No efficient algorithm is known.
  - Best algorithm is finding the discrete-log.
- Man in the middle attack
  - An attacker in the middle can impersonate Alice or Bob and establish a shared key with Alice and Bob.
  - The parties must be authenticated
    - With a PKI, the parties may sign  $g^a$  and  $g^b$

- The MQV protocol
  - Designed by Menezes, Qu and Vanstone in 1995.
  - Efficient authenticated Diffie-Hellman protocol.
  - Requires a PKI.
  - Standardized in the public-key standard IEEE P1363.
- The HMQV protocol (2005)
  - Improvement of MQV with formal security analysis.

# The HMQV protocol

- Setup:
  - Alice has public-key  $g^a$  and sk  $a$
  - Bob's has public-key  $g^b$  and sk  $b$
- The HMQV protocol:
  - Alice and Bob run a basic DH key exchange
    - Alice sends  $X = g^x$  to Bob
    - Bob sends  $Y = g^y$  to Alice
  - Alice computes  $\sigma_A = (YB^e)^{x+da}$
  - Bob computes  $\sigma_B = (XA^d)^{y+eb}$
  - Alice and Bob set  $K = H(\sigma_A) = H(\sigma_B)$
  - where  $d = H_2(X, ID_{Bob})$  and  $e = H_2(Y, ID_{Alice})$

# Security properties of HMQV

- HMQV is proven secure in the Canetti-Krawczyk model
  - in the random oracle model
  - under the CDH assumption
- The model covers:
  - Impersonation attacks
    - An attacker impersonates Alice and establishes a session key with Alice and Bob.
  - Known-key attacks
    - If a session key is leaked, this does not affect the security of other session keys.

# The Schnorr signature scheme

- Key generation:
  - Let  $G$  be a group of order  $q$  and let  $g$  be a generator.  
Generate a private key  $x \leftarrow \mathbb{Z}_q$
  - The public key is  $y = g^x \pmod p$
- Signature generation of  $m$ 
  - Generate a random  $k$  in  $\mathbb{Z}_q$
  - Let  $r = g^k$ ,  $e = H(m||r)$  and  $s = (k - xe) \pmod q$
  - Signature is  $(s, e)$ .
- Signature verification of  $(s, e)$ 
  - Let  $r_v = g^s y^e \pmod p$  and  $e_v = H(M||r_v)$
  - Check that  $e_v = e$ .

- Security of Schnorr signatures
  - Provably secure against existential forgery in a chosen message attack
  - in the random oracle model under the discrete-log assumption
  - using the “Forking lemma” (Pointcheval and Stern, 1996)