Katz, Lindell
Introduction to Modern Cryptrography
Slides Chapter 8

Markus Bläser, Saarland University

Weak factoring experiment

The weak factoring experiment

1. Choose two n-bit integers x_1, x_2 uniformly.
2. Compute $N := x_1 x_2$.
3. $\mathcal{A}$ is given N and outputs $x_1', x_2' > 1$.
4. The result of the experiment is 1 if $N = x_1' x_2'$ and 0 otherwise

Weak factoring experiment

The weak factoring experiment

1. Choose two n-bit integers x_1, x_2 uniformly.
2. Compute $N := x_1 x_2$.
3. $\mathcal{A}$ is given N and outputs $x_1', x_2' > 1$.
4. The result of the experiment is 1 if $N = x_1' x_2'$ and 0 otherwise

Easy to win $\longrightarrow$ x_1, x_2 prime

Generating primes

Algorithm 8.31/8.34

Input: length 1^n, parameter 1^t
Output: a uniform n-bit prime

1. for $i := 1$ to $t = 3n^2$
2. $\quad p' \leftarrow \{0,1\}^{n-1}$
3. $\quad p := 1 \| p'$
4. $\quad$ if p is prime then return p
5. return "failure"

Primality testing

Observation

N *prime* $\Longrightarrow a^{N-1} = 1 \mod N$ *for any* a

Algorithm 8.35

Input: integer N, parameter 1^t

1. for $i := 1$ to t do
2. $\quad a \leftarrow \{1, \ldots, N-1\}$
3. $\quad$ if $a^{N-1} \neq 1 \mod N$ then return "composite"
4. return "prime"

Primality testing (2)

Proposition (8.36)

Let G *be a finite group and* $H \subseteq G$*. If* H *is nonempty and for all* $a, b \in H$ *we have* $ab \in H$*, then* H *is a subgroup of* G*.*

Lemma (8.37)

Let H *be a proper subgroup of a finite group* G*. Then* $|H| \leq |G|/2$*.*

Theorem (8.38)

If there is a witness a *such that* $a^{N-1} \neq 1 \mod N$*, then this is true for a least half the elements in* $\mathbb{Z}_N^*$*.*

Miller-Rabin test

Write $N - 1 = 2^r u$ with u odd.

Definition

a is a strong witness that N is composite, if (1) $a^u \neq \pm 1 \mod N$ and (2) $a^{2^i u} \neq -1 \mod N$ for all $1 \leq i \leq r - 1$.

Lemma (8.39)

If N is an odd prime, then the only square roots of 1 modulo N are $\pm 1 \mod N$.

Lemma

If N is prime, then N has no strong witnesses.

Theorem (8.40)

If N is odd and not a prime power, then half of the elements of $\mathbb{Z}_N^*$ are strong witnesses for N.

Miller-Rabin test (2)

Algorithm 8.44

Input: $N > 2$, parameter 1^t

1. If N is even, return "composite"
2. If N is a perfect power, return "composite"
3. Write $N - 1 = 2^r u$ with u odd
4. for $i := 1$ to t do
5. $\quad a \leftarrow \{0, \ldots, N-1\}$
6. $\quad$ If a is a strong witness, return "composite"
7. return "prime"

Theorem (8.33)

If N is prime, then the Miller-Rabin test always outputs "prime". If N is composite, then it outputs "composite" with probability $\geq 1 - 2^{-t}$. The running time is polynomial in t and $|\log N|$.

The factoring experiment

GenModulus: On input 1^n, output (N, p, q) where

- $N = pq$,
- p, q are n-bit primes

(except with probability negligible in n).

The factoring experiment $\mathsf{Factor}_{\mathcal{A},\mathsf{GenModulus}}(n)$

1. Run $\mathsf{GenModulus}(1^n)$ to obtain (N, p, q).
2. $\mathcal{A}$ is given N and outputs $p', q' > 1$.
3. The outcome is 1 if $p'q' = N$ and 0 otherwise.

Definition (8.45)

Factoring is hard relative to GenModulus if for all ppt $\mathcal{A}$,

$$\Pr[\mathsf{Factor}_{\mathcal{A},\mathsf{GenModulus}}(n) = 1] \leq \mathsf{negl}(n).$$

RSA assumption

Factoring does not yield directly practical systems ...

- $\phi(N) := |\mathbb{Z}_N^*| = |\{a \in \{1, \ldots, N-1\} \mid \gcd(a, N) = 1\}|$
- $N = \prod_i p_i^{e_i} \longrightarrow \phi(N) = \prod_i p_i^{e_i - 1}(p_i - 1)$
- $a^{\phi(N)} = 1 \mod N$

GenRSA(1^n):

1. $(N, p, q) \leftarrow$ GenModulus(1^n)
2. $\phi(N) := (p-1)(q-1)$
3. Choose $e > 1$ such that $\gcd(e, \phi(N)) = 1$
4. Compute $d := e^{-1} \mod \phi(N)$.
5. return N, e, d.

RSA assumption (2)

RSA-experiment $\text{RSA-Inv}_{\mathcal{A},\text{GenRSA}}(n)$

1. Run $\text{GenRSA}(1^n)$ to obtain (N, e, d).
2. Choose $y \in \mathbb{Z}_N^*$ uniformly at random.
3. $\mathcal{A}$ is given N, e, y and outputs $x \in \mathbb{Z}_N^*$.
4. The outcome is 1 if $x^e = y \mod N$.

Definition (8.46)

The RSA problem is hard relative to GenRSA if for all ppt $\mathcal{A}$,

$$\Pr[\text{RSA-Inv}_{\mathcal{A},\text{GenRSA}}(n) = 1] \leq \text{negl}(n).$$

- If we can factor, we can win RSA-Inv.
- Given N, e, d, we can factor (Theorem 8.50).
  This is *not* the converse of the first statement.

Extended Euclidean Algorithm

Proposition (8.2)

For all natural numbers a, b *there are integer* X *and* Y *("cofactors") such that*

$$Xa + Yb = \gcd(a, b).$$

$\gcd(a, b)$ *is the smallest integer (in absolute value) that can be written like this.*

- Assume that $a > b$ and write $a = qb + r$ with $0 \leq r < b$.
- If $r = 0$, then $b = \gcd(a, b) = 0 \cdot a + 1 \cdot b$.
- Otherwise, recursively compute X, Y such that $Xb + Yr = \gcd(b, r) = \gcd(a, b)$. Then $Y \cdot a + (X - Yq)b = \gcd(a, b)$.

$\longrightarrow$ Extended Euclidean Algorithm (B.10)

Factoring and RSA assumption

- Let $\gcd(a, M) = 1$ and $Xa + YM = 1$ Then $Xa = 1 \mod M$, i.e., X is a modular inverse of a modulo M.
- If $de = 1 \mod \phi(N)$, then $de = 1 + Y\phi(N)$ for some Y, hence

$$(x^d)^e = x^{de} = x^{1+Y\phi(N)} = x \mod N,$$

  since $z^{\phi(N)} = 1 \mod N$ for any z.
- So x^d is an eth root of x modulo N.
- If we can factor N, then we can compute $\phi(N)$.

Cyclic groups

Let G be a *finite* group of size m and $g \in G$.We set

$$\langle g \rangle := \{g^0, g^1, \dots\}.$$

We have $g^m = 1$ and let $i \leq m$ be the smallest index with $g^i = 1$. Then

$$\langle g \rangle = \{g^0, \dots, g^{i-1}\}.$$

Proposition

$\langle g \rangle$ *is a subgroup of* G.

Definition (8.51)

The order of g is the smallest i with $g^i = 1$.

Cyclic groups (2)

Proposition (8.53)

Let g *be an element of order* i *in a finite group* G*. Then* $g^x = g^y$ *iff* $x = y \mod i$*.*

Definition

Let G be a finite group. G is *cyclic* if there is a $g \in G$ with $\langle g \rangle = G$. Such a g is called a *generator*.

Proposition (8.54)

Let g *be an element of order* i *in a finite group* G *of size* m*. Then* $i|m$*.*

Corollary (8.55)

If G *is a finite group of prime order, then* G *is cyclic. All elements except* 1 *are generators.*

Discrete logarithm assumption

Group generation algorithm on input 1^n outputs:

- a description of a cyclic group G
- its order q (of bit length n)
- a generator g.

Description means

- representation of elements as bit strings such that
- we can efficiently ($=$ polynomial in n) check whether a string is a group element and
- we can efficiently perform the group operation.

Discrete logarithm assumption (2)

G finite cyclic group with generator g, $G = \{g^0, \ldots, g^{q-1}\}$.

Definition

Let $y \in G$. The unique $0 \leq i \leq q-1$ such that $g^i = y$ is the *discrete logarithm* of y (with respect to g).

Notation: $i = \log_g y$

Some rules:

- $\log_g 1 = 0$
- $\log_g y^r = r \cdot \log_g y \mod q$
- $\log_g(xy) = \log_g(x) + \log_q(y) \mod q$.

Discrete logarithm assumption (3)

The discrete logarithm experiment $\mathrm{Dlog}_{\mathcal{A},\mathrm{gen}}(n)$**:**

1. Generate an instance (G, q, g) with q having n bits.
2. Choose $h \in G$ uniformly at random.
3. $\mathcal{A}$ is given (G, q, g) and h and outputs $x \in \mathbb{Z}_q$.
4. The results of the experiment is 1 if $g^x = h$ and 0 otherwise

Definition

We say that the discrete logarithm problem is hard with respect to Gen if for all ppt $\mathcal{A}$,

$$\Pr[\mathrm{Dlog}_{\mathcal{A},\mathrm{Gen}}(n) = 1] \leq \mathrm{negl}(n).$$

Computational Diffie–Hellman assumption

- Let G be a cyclic group with generator g.
- Let $h_1 = g^{x_1}$ and $h_2 = g^{x_2}$. We define

$$\mathrm{DH}_g(h_1, h_2) := g^{x_1 \cdot x_2} = h_1^{x_2} = h_2^{x_1}.$$

- Computational Diffie-Hellman experiment: Compute $\mathrm{DH}(h_1, h_2)$ from uniform h_1 and h_2 (without knowing x_1 or x_2).
- If discrete logarithm is easy, so is computational Diffie-Hellman: Compute $x_1 = \log_g(h_1)$ and output $h_2^{x_1}$.
- Converse is not known.

Decisional Diffie-Hellman assumption

Definition (8.63)

The decisional Diffie-Hellman problem DHH is hard relative to Gen if for all ppt $\mathcal{A}$

$$|\Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy}) = 1]| \leq \mathsf{negl}(n)$$

where the probability is taken over the randomness of the generation algorithm $\mathsf{Gen}(1^n)$ outputting (G, q, g) and x, y, and z in $\mathbb{Z}_q$ are chosen uniformly at random.

"distinguish g^{xy} from a random element g^z"

Choosing a group

The order of the group should be a prime:

- Discrete logarithm is supposed to be harder in such groups.
- Finding a generator is trivial
- $\mathrm{DH}_g(h_1, h_2)$ is close to uniform for such groups.

Theorem

Let $p = rq + 1$ *with* p *and* q *being prime. Then*

$$G := \{h^r \mod p \mid h \in \mathbb{Z}_p^*\}$$

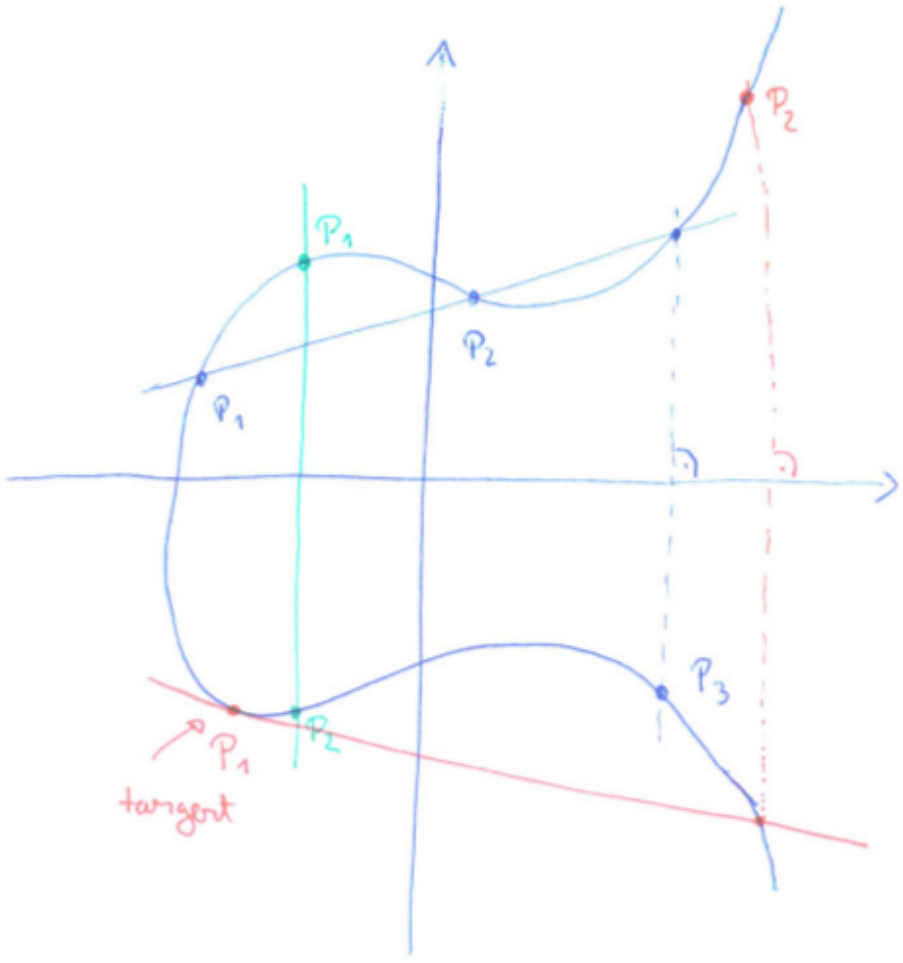
is a subgroup of $\mathbb{Z}_p^*$ *of order* q.

Group generation

Input: Security parameter 1^n, parameter $\ell = \ell(n)$

1. generate an n-bit prime q uniformly at random
2. generate an ℓ-bit prime p such that $q|(p-1)$
3. choose $h \in \mathbb{Z}_q$ with $h \neq 1$
4. set $g = h^{(p-1)/q} \mod p$
5. return p, q, g

Elliptic curves



$P_1 + P_2 = P_3$

$P_1 + P_2 = \mathcal{O}$

$2P_1 = P_1 + P_1 = P_2$