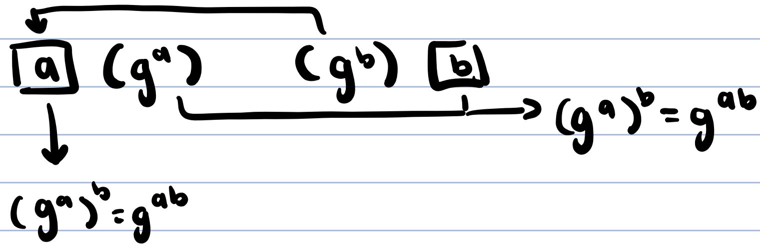
CS 55500: Lecture 10

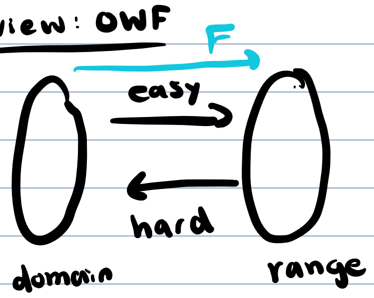
Recap

Diffie-Hellman Construction:

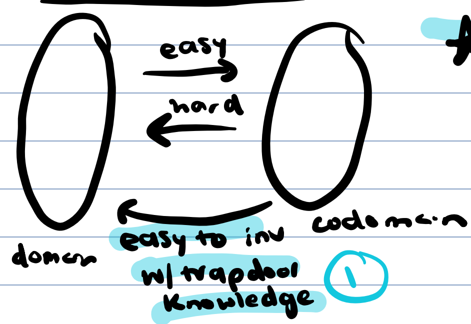


← Key exchange

Review: OWF



Trapdoor OWF



★ domain = range
2

Trapdoor Func: defn

A function (family) $F = \{F_n\}_{n \in \mathbb{N}}$ w/ F_n itself a collection of functions $F_n = \{F_i : \{0,1\}^n \rightarrow \{0,1\}^{m(n)}\}_{i \in I_n}$ is trapdoor OWF if:

a. easy to sample function index w/ a trapdoor
b. easy to compute $F_i(x)$ given i and x
c. easy to compute inverse of $F_i(x)$ given t_i ← trapdoor value
d. one-way for every PPT A when all but t_i revealed

Trapdoor Permutation to IND-Secure Pub. key encryption

Construction Try 1

Gen: rand. index i w/ t_i → release i

Enc: output $c = F_i(m)$

Dec: output $F_i^{-1}(c)$ which can be done using knowledge of t_i

★ having a OWF does not guarantee all bits are secret → can have an input bit that is not hardcore → in this case a OWF is not IND-CPA secure bc you could use that bit to dist b/w input messages

Not IND-CPA secure bc OWF of trapdool can leak some info from input m in c → breaks IND-CPA distinguishability

Try 2:

Gen: Same as above

BUT this only encrypts one bit

Enc: Pick random r : Output $c = (F_i(r), HCB(r) \oplus m)$

Dec: recover r from $F_i(r)$ using t_i → calc $HCB(r)$ + XOR w/ m

↳ This is IND-CPA → hybrid argument

Trapdoor Permutation: Candidates

2 candidates (both rely on factoring being hard)

a. RSA

b. Rabin / Blum-Williams

RSA...

num thry review:

- $N = pq$: prod. of two large primes
- $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid \gcd(a, N) = 1\}$ is a grp
  - group op is mult. mod N
  - inv exist + easy to compute
  - $\phi(n) = (p-1)(q-1) = \text{ord}(\mathbb{Z}_n^*)$

- given $(p,q) \rightarrow pq$ easy $pq \rightarrow (p,q)$ hard

Trapdoor const.

Let $e \in \mathbb{Z}$ w/ $\gcd(e, \phi(n)) = 1$. Then map $F_{N,e}(x) = x^e \pmod n$ is a trapdoor permutation.

↳ given $d \ni ed \equiv 1 \pmod{\phi(n)}$, it's easy to comput x given x^e

↓

Proof: $(x^e)^d \equiv x^{ed} \equiv x^{k\phi(n)+1} \equiv 1 \cdot x^1$

RSA trapdoor:

for $e \in \mathbb{Z}$ with $\gcd(e, \phi(n)) = 1$. Then, the map $F_{N,e}(x) = x^e \bmod N$ is a trapdoor permutation.

- given N, e and $x^e \bmod N$, hard to compute x

★ if factoring is easy, RSA is broken

Chinese Remainder thm

$x \equiv r_p r_q \pmod{pq}$ $\Rightarrow$ $x = k_p p + r_p \pmod q$, $x = k_q q + r_q \pmod p$

$x \equiv r_p \pmod p$ $\quad$ $x \equiv r_q \pmod q$

finding $e^{-1} \pmod{\phi(n)}$

$ed \equiv 1 \pmod{\phi(n)}$

$p-1$ $\quad$ $q-1$

$e \equiv r_{ep} \pmod p$ and $e \equiv r_{eq} \pmod q$
$d \equiv r_{dp} \pmod p$ and $e \equiv r_{dp} \pmod q$

only need to do one of these and we know finding inv w/ a prime modulus is easy $\rightarrow$ $\boxed{ed \equiv r_{ep} \cdot r_{dp} \equiv 1 \pmod p}$ OR $\boxed{e \equiv r_{eq} \cdot r_{dp} \pmod q}$

$\rightarrow$ break down until prime modulus

Need hard-core bit too, not just OWF:

hardcore-bit for RSA:

a. GL bit $GL(r; r') = \langle r, r' \rangle \bmod 2$

b. Least significant bit $LSB(r)$

c. Most significant bit : $HALF_N(r) = 1$ iff $r < N/2$

↳ here any bit of r is hardcore

RSA Enc

- $Gen(1^n)$: Let $N = pq$ and (e, d) $\ni$ $ed \equiv 1 \bmod (\phi(n))$

  $pk = (N, e)$ and $sk = d$

- $Enc(pk, b)$ w/ b being a bit: gen random $r \in \mathbb{Z}_n^*$ $\rightarrow$ hard to find r

  ↳ output $\underbrace{r^e \bmod N}_{OWF}$ and $\underbrace{LSB(r)}_{HCB} \oplus m$

  slightly biased

- $Dec(sk, c)$: recover r via RSA inversion

-Impractical bc can only encrypt one bit at a time...

Theoretic soln

GL: give one bit with $\langle r', r \rangle$

vs

use rand matrix $A = \begin{bmatrix} r_1' \\ r_2' \\ \vdots \\ r_d' \end{bmatrix}$ $\rightarrow$ Ar gives d-hardcore bits

↳ but always giving rand A is a lot

Practical soln

having function f that encodes $LSB(r)$ to many bits

then $f(LSB(r)) \oplus m$

Quadratic Residues Mod p

- Let p be prime
- exactly half of $\mathbb{Z}_p^*$ are squares
- define Legendre symbol $\left(\frac{x}{p}\right) = \begin{cases} 1 & \text{if } x \text{ is not a squre} \\ -1 & \text{if } x \text{ is not a square} \\ 0 & \text{if } x \equiv 0 \pmod{p} \end{cases}$

so... $\left(\frac{x}{p}\right) = x^{p-1/2}$

→ it is easy to compute square roots mod p

- for $p \equiv 3 \pmod 4$ → square roots of x mod p $= \pm x^{(p+1)/4}$

$$\left(\pm x^{(p+1)/4}\right)^2 \equiv x^{(p+1)/2} = x \cdot x^{p-1/2} \pmod p = x \pmod p$$

Quadratic Res mod N=pq

x is a square mod N iff x is square mod p and square mod q

↓

Jacobi symbol: $\left(\frac{x}{N}\right) = \underbrace{\left(\frac{x}{p}\right)\left(\frac{x}{Q}\right)}_{1 \text{ if both square or both non-square}}$

*we can find $\left(\frac{x}{N}\right)$ in poly time w/o knowing p or q

for all Jac_{+1} → square mod p and q → QR_N

→ non-square mod p and q → QNR_N

↓

Given N, no PPT can distinguish between QR_N and QNR_N

↓why?

suppose you know p, q:

find sqroot of y mod p and q

$x \equiv y_p^2 \pmod p$ $\quad x = y_q^2 \mod q$

$y = c_p y_p + c_q y_q$ → a. $c_p = 1 \mod p$ and $c_p = 0 \mod q$

b. $c_q = 0 \mod p$ and $c_q = 1 \mod q$

suppose you have an alg that can comp any $\sqrt{x}$

$$x \to \boxed{\sqrt{\ }} \to y \ni y^2 \equiv x \bmod N$$

feed box $x = z^2 \pmod{n}$ for a rand z

then w/ prob $1/2$, $\gcd(z+y, N)$ is a non-trivial factor of N