rsa encryption

LECTURE 11 // WED 3/13

diffie-hellman model

RSA encryption follows this general security model

$\forall$ party A, they have a public key PK_A that others use to encrypt messages M to A.

$$C = PK_A(M)$$

party A has a secret key SK_A that they can use to decrypt ciphertexts they receive.

$$M = SK_A(C)$$

the cinch is that even publishing PK_A, nothing is revealed about SK_A

RSA implementation

RSA relies on the difficulty of factoring the product n of two very large primes p, q to be secure

the private and public keys are pairs of integers:

$PK = (n, e)$ where $n = p \cdot q$ and $\gcd(e, \varphi(n)) = 1$

$SK = (n, d)$ where $de \equiv 1 \mod \varphi(n)$

eulers totient counts pos. int. up to n that are rel. prime to n

to encrypt and decrypt, keys are used as follows:

$C \equiv Enc(PK, m) \equiv m^e \pmod{n}$

$m \equiv Dec(SK, C) \equiv c^d \pmod{n}$

how are our <u>keys generated</u>?

1. two distinct large primes p, q are chosen
2. compute $n = pq$
3. compute $\lambda(n) = \text{lcm}(\varphi(p), \varphi(q)) = \text{lcm}(p-1, q-1)$ where λ is Carmichael's totient function. $\lambda(n)$ is private

   $\varphi(n)$ can be used instead

4. choose int. e s.t. $1 < e < \lambda(n)$ and $\gcd(e, \lambda(n))$
also: $e \leftarrow \mathbb{Z}^*_{\lambda(n)}$, $\varphi(n) = |\mathbb{Z}_n^*| = (p-1)(q-1)$ ↖ this means e & $\lambda(n)$ are relatively prime to another

5. compute $d \equiv e^{-1} \pmod{\lambda(n)}$
d is the modular multiplicative inverse of e mod $\lambda(n)$

let's briefly show correctness

$$
\begin{aligned}
& \mathrm{Dec}(SK, \mathrm{Enc}(PK, m)) \\
&= \mathrm{Dec}(SK, m^e \bmod n) \\
&= (m^e)^d \bmod n \\
&= m^{e \cdot e^{-1} \bmod \lambda(n)} \bmod n \\
&= m^1 \bmod n = m
\end{aligned}
$$

fun fact

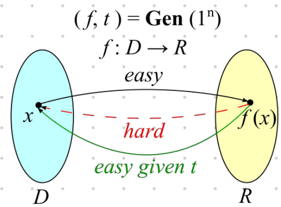
practical implement. use chinese remainder theorem

RSA IS ...

→ not semantically secure

→ not even randomized

→ a trapdoor permutation
↳ easy to compute, hard to invert (RSA assumption)
↳ easy to invert given a trapdoor d

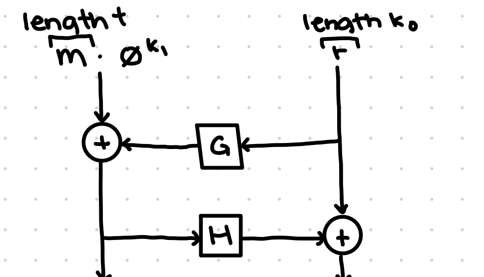


wikipedia

making RSA CCA2 secure

idea: apply RSA encryption to an encoding of the message

we call this OAEP: optimal asymmetric encryption padding



$x_0 = G(r) \oplus (m \cdot 0^{k_1})$ $x_1 = r \oplus H(x_0)$

$\mathrm{Enc}_{n,e}(x_0, x_1)$

where G, H are random oracles

OAEP is randomized $\forall m \rightarrow Enc(m) = (x_0, x_1)$ is randomized.
without revealing the encoding (x_0, x_1) entirely, nothing can be learned about m

any trapdoor with OAEP encoding is CPA secure
RSA with OAEP is CCA2 secure