

Public Key

Table of Contents:

1. Public key cryptography
 2. Diffie-Hellman key exchange
 3. RSA
 4. Provably secure cryptosystems
-

One-Way Functions

A) Multiplication (Integer Factorization)

| Its inverse is FACTORING [hard]

trapdoor: if you know one of the two numbers, finding the other one is easy.

RSA

B) Modular Exponentiation (Discrete Logarithm)

| Its inverse is DISCRETE LOGARITHM [hard]

trapdoor: In RSA, if you know $\phi(N)$, you can compute the inverse exponent efficiently.

Diffie-Hellman/DSA/ElGamal

What is $\phi(N)$?

counts how many numbers from 1 to N are co-prime with N (share no common factors except 1)

EXAMPLE:

```
Numbers 1 to 10: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Co_prime with 10: {1, 3, 7, 9} (don't share factors 2 or 5)
 $\phi(10) = 4$ 
```

RSA

Deterministic algorithm [partial information leaks]

Type of Possible Attacks:

1. Brute Force
2. Mathematical: factoring the product of two primes
3. Timing: ciphertext-only attacks, depends on the running time of the decryption
4. Choose Cipher Attacks: exposes properties of the RSA algorithm
5. Factor N: given an RSA modulus N and the value $\phi(N)$, from that you can get p, q

How to stop Deterministic?

- Add Nonce/Salt
- Add random padding

Textbook RSA is not CPA secure Why?

- Same message always produces same ciphertext [Deterministic]
- No randomness involved

Essentially RSA is malleable due to its homomorphic property [$c = 2^e c \pmod{N}$]

| This means the scheme is not secure against **Adaptive Chosen Ciphertext CCA2**.