Date:

#public-keys

#Modular-Exponentiation
↳ All RSA operation involves #.

$2^{10} \bmod 17$,
$1024 \bmod 17 = 4$.
pow(base, exponent, modulus)

↳ Trapdoor function
↳ Easy to compute in one direction, but hard to do in reverse unless right information.

#RSA
↳ Rivest, Shamir, Adleman

$N = p \cdot q$
↓
Modulus

↳ Together exponent & Modulus form RSA public key (N, e)
↳ Most common $e = 65537$.

Date:

Su Mo Tu We Th Fr Sa

Euler's Totient. ϕ

↳ RSA relies on difficulty of factorization of Modulus N. If prime numbers can be factor can be deduced, then we can calculate euler totient of N & decrypt ciphertext.

$$N = p \cdot q$$

$$\boxed{\phi(N) = (p-1) \cdot (q-1)}$$

Private key (a) is used to decrypt CT. made with its Public key. & also used to sign.

↳ In RSA Private key is Modular Multiplicative inverse of e modulo $\phi(N)$, Euler's totient of N.

$$d \equiv e^{-1} \bmod \phi(N)$$

Date: Su Mo Tu We Th Fr Sa

RSA Signature.

- How can u ensure person receiving ur message knows u wrote it?

Imagine message u wrote "m"
Encrypt with friend public key

$$c = m^{e_0} \bmod N_0.$$

To sign message, calculate Hash of messge: H(m) & encrypt with your private key.

$$S = H(m)^{d_1} \bmod N_1.$$

- Friend can decrypt this msg using their ~~public~~ private key.

$$m = c^{d_0} \bmod N_0$$

- U sing ur public key they calculate

$$S = S^{e_1} \bmod N_1.$$

Date:

Now by computing H(m) & comparing to 'S'. If its same we can ensure message is sent by them.

Factoring

↳ Using primes atleast 1024 ~~to 2048~~ bits long is recomended.

↳ Multipying two such 1024 primes gives you 2048 bit large Modulus

↳ ~~With~~

If 'n' is prime numbes just 1 prime not p·q, anyone could compute private key.

↳ Eulers Totent would be

$$\phi(n) = n - 1$$

So it's easy to compute

↑ Easy to factor

78 Many Primes (chall)

- Your Modulo is product of over 30 primes.

for decryption:

- We need to factor 'n' into all its prime factor.
- We use optimized algorithm for this many primes like Elliptic Curve Method (ECM).
- Use factordb.com to get all primes if we have 'n'.

Salty (chall)

- e = 1 means no encryption.

$$ct = pt^1 \bmod n$$
$$= pt \bmod n$$

Date:

Modulus Inutils (chall)

↳ e = 3, 2 plaintext is short

↳ If pt^e ($(p \cdot T)^e$) is less than n means its not safe.

↳ In our case:

$$CT = (PT)^3 \bmod n$$
$$= (PT)^3$$

↳ So its just cube root of p C·T.

Multiplying or addition ~~forms~~ any two elements in set returns another element in set of integers modulo N.

↳ When Modulo is prime, N = p,

More than a College

CamScanner

Date:

Diffie-Hellman

- Ring Z/NZ is set $\{0,1,2 \ldots N-1\}$ where addition & multiplication are both in that set.
- when $N = p$ is prime, every non-zero element has multiplicative inverse - meaning for any $g \neq 0$, there exists d such that

$$g \cdot d \equiv 1 \pmod{p}$$

- This makes it field F_p. where g = generator (or just element) you are given in infinite field F_p.
- 'd' is multiplicative inverse of g.

To find $d = g^{-1}$ where $g = 209$ & $p = 991$, we can use E.E $_{\text{algo}}$.

$$g \cdot x + p \cdot y = \gcd(g, p) = 1$$

Date

→ Diffie Hellman protocol works with element of some finite field Fp where prime modulus is typically very large.

#Generator of Groups.

↳ In finite field Fp. if ya take any element 'g' & keep multiplying by itself.

$$g, g^2, g^3, g^4 \ldots$$

↳ Eventually sequence cycles back to 'g'. The set of all distinct values in cycle forms subgroup $H = (g)$.

↳ The no of distinct element in (g) is called order of g. ord(g). By Lagrange's theorem, order of any element must divide p-1 #

↳ ~~Positive~~ Primitive element is one whose generated subgroup equals entire multiplicative group. $Fp = \{1, 2, 3, \ldots, p-1\}$.

More than a College

That means,

$$ord(g) = p - 1$$

4) Every non-0 ~~group~~ element of F_p can be written as

$g^n \bmod p$ for some n.

4) This is why they are called generators (they generate whole group).

4) $p = 28151$ & find smallest element g which is primve element of F_p.

Sol

1) find $p-1 = 28150$
2) Break 28150 into prime factors $(2 \times 5^2 \times 563)$
   $= (2, 5, 563)$
3) g is primitive element if & only if

$$g^{(p-1)/q} \neq 1 \pmod p$$

4) Then find all & get smallest.

Date:

Su Mo Tu We Th Fr Sa

L) D-H protocol is used coz the discrete logarithm is hard computation

Steps for DH

(p)

① Establish prime & generator (g) of finite field.
   L) Must be chosen carefully.
   L) Eg. $p = 2 \cdot q + 1$ is usually picked such that only factors of p-1 are (2/q) where q is some other large prime. It protects D-H from Pohlig-Hellman algo.

② User picks secret ~~+g~~ int a < p-1 & calculates,

$g^a \bmod p$.

L) The value a is known as secret value. while

$A = g^a \bmod p$ is public value.

More than a College

CamScanner

Computing shared secrets in DH

- You generate your secret integer b & calculate your public value

$B = g^b \bmod p$ which u send to Alice

Now to get shared secrets.

shared secret $= A^b \bmod p$

Deriving Symetric keys (hall)

= We have DH parameters g|p.

4) Alice public value (A)
4) our secret (b)
4) Alice encrypted flag with IV & encrypted flag (C.T)

4) AES is in CBC

source = How encryption works

decrypt.py = Just helper

Steps

① Compute shared secret

≠

② Derive AES key via SHA1.

key = Sha1 [:16]

③ Decrypt with AES CBC

Parameter Injection (Challenge).

- we can intercept Alice & Bob DH key but also rewrite their messages. Recover shared secret.

- we get Alice Intercept(p, g, A)
- we can replace her A with p before forwarding to Bob & compute.

Shared secret = $A^b \bmod p = p^b \bmod A$

$= 0$

- If we inject A: p bob computes.

$p^b \bmod p = 0$

- Alice also now gets 0 whenever she computes.

↳ Now, we already know shared secret is 0. So we derive AES key & Decrypt the flag like before.

Export-grade (Chall).

↳ Based on real Logjam attack (2015).

↳ Steps

① Parameter Downgrade:

↳ MITM between A & B.

A offers list of DH groups (DH1536, DH1024).

You intercept A message & replace with weakest option & send to B. B now has DH64 so he sends DH64 parameter which is easy to crack.

② Crack.

↳ DH64 = Brute force.

↳ Bob Public value is

$B = g^b \bmod p$

So simply brute force until b value is find till:
$g^b \bmod p = B$ #

↳ Now with b, compute shared secret =
↳ $A^b \bmod p$.

↳ The same as prev chall.