Attacks on El Gamal

Malleability.

If Oscar (a man-in-the-middle) intercepts ciphertext (K, c) and replaces it with (K, sc). ← chooses s

then Bob decrypt it as

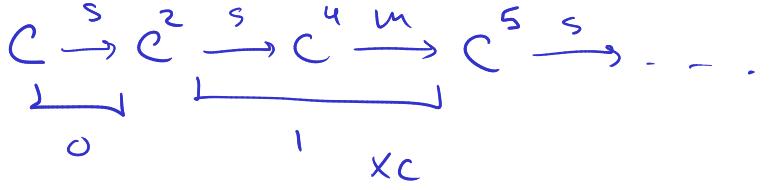
$$K^{-b} sc = s(K^{-b} c) = sm$$

So Oscar doesn't know m, but can manipulate it predictably.

This is an active attack. (interaction w/ protocol).

Timing attacks on RSA. (Side channel)

Server decrypting: $c^d \pmod n$.



Loops through bits of d:

① If 1, does a squaring & a multiplication

② If 0, does a squaring

Ex., $d = 1111\ldots1_2$ takes longer than $d = 10\ldots0_2$.

We can <u>time</u> the server.

<u>1st approach</u>:

Msr time t_i to decrypt ciphertexts c_i for $i = 1, \ldots, n$.

Msr time to do known operations on similar hardware in lab.

Then avg time $= m = \dfrac{(t_1 + \cdots + t_n)}{n}$ will approximate time for $\log d$ squarings + k mult's

where k = Hamming wt = # 1's in binary expansion of d.

Better (2nd approach). Bit-by-bit discovery of d.

Q: Is the 1st bit a 0 or 1?
(no mult) (mult.) ✡

Msr times: $t_1, \ldots, t_n$ for decrypting ciphertexts $c_1, \ldots, c_n$

Lab msr times to do 1st multiplication for $c_1, \ldots, c_n$ ✡
$t_1', \ldots, t_n'$

Compute variances: $\frac{(t_1 - m)^2 + \cdots + (t_n - m)^2}{n} =: \mathrm{Var}(t_i\text{'s})$.
f_{avg}

also compute $\mathrm{Var}(t_i - t_i')$.

If $\mathrm{Var}(t_i) > \mathrm{Var}(t_i - t_i')$, conclude 1st bit is 1.

If " $<$ " " 0.

General Ppl: For independent processes S and S',

variances add: $Var(s+s') \approx Var(s) + Var(s')$

Situation 1 bit is 1 (includes mult.)

$t_i - t_i' + t_i'$

$t_i - t_i'$: rest of comput. t_i': mult.

indep

$Var(t_i) \approx V(t_i - t_i') + V(t_i')$

$\Rightarrow Var(t_i) > Var(t_i - t_i')$

Situation 2. bit is 0. (no mult)

t_i t_i'

t_i: full comp t_i': mult. in lab (irrelavent)

indep

$Var(t_i - t_i') \approx Var(t_i) + Var(t_i')$

$\Rightarrow Var(t_i) < Var(t_i - t_i')$.

Countermeasures: fake multiplications ~~to take up time~~

Finding primes

How many primes are there?

$$\#(N) \sim \frac{N}{\log N} \quad \left(\sim \int_2^N \frac{dt}{\log t}.\right)$$

$\#(N)$ = # of primes $0 < p < N$

So for integers of size $\sim N$, the prob. of being prime is $\frac{1}{\log N}$ (not too bad).