xgcd(a,b) gives integers x, y s.t.

$$xa + yb = \gcd(a,b).$$

if $\gcd(a,b) = 1$, then x is the modular inverse of a mod b:

$$xa + yb = 1$$
$$xa = 1 - yb$$

Now reduce mod b: $xa \equiv 1 \mod b$

Why did we care? Came up in RSA:

needed to find $d \equiv e^{-1} \mod \varphi(n)$. (Hence the requirement $\gcd(e, \varphi(n)) = 1$)

Hybrid encryption: idea/motivation:

Public key is convenient (easy to distribute keys).

Why not just use ElGamal? Too slow! (for long messages)

Symmetric key enc. can be really fast.

AES has hardware support (even on my Sandybridge from 2010).

Enc/dec @ 1 byte per clock cycle.

"AES-NI"

$$\text{Hybrid-Enc}(m) = \left(E_{PK}(k), AES_k(m)\right)$$

(where k is randomly chosen by sender.)

OWFs: $f: X \longrightarrow Y$

f easy to compute, but, for $x \in_R X$,
it is hard to find $x' \in X$ s.t.

$$f(x') = y = f(x) \quad \text{(given only } y\text{)}$$

Vanilla RSA does this (and allows a trapdoor to invert.)

So even though vanilla RSA isn't good for encryption by itself (Not IND-CPA even), it's fine for use in KEM / Hybrid encryption.

ElGamal looks sort of like this...

Say $\langle g \rangle = G$, $|G| = q$.

$SK = a \in_R \mathbb{Z}_q$, $PK = A = g^a$

$$E_{PK}(m) = (g^b, A^b \cdot m)$$

$(m \in G)$

g^b: $\approx$ key encapsulation of DH key g^{ab}

$A^b \cdot m$: $\approx$ OTP enc. w/ g^{ab}.

Choices for G? Subgroup of $\mathbb{Z}_p^*$ why not $(\mathbb{Z}_p, +)$? DLP not hard!

Requirement: DLP has to be hard for sure.

DDH should also be true

Recall: DDH: $(g^a, g^b, g^{ab}) \approx_{PPT} (g^a, g^b, g^c)$

$a, b, c \in_R \mathbb{Z}_q$

CDH: (computational) compute g^{ab} from g^a, g^b, where $a, b \in_R \mathbb{Z}_q$.

CDH conjectured to hold in $\mathbb{Z}_p^*$, but DDH is false there!

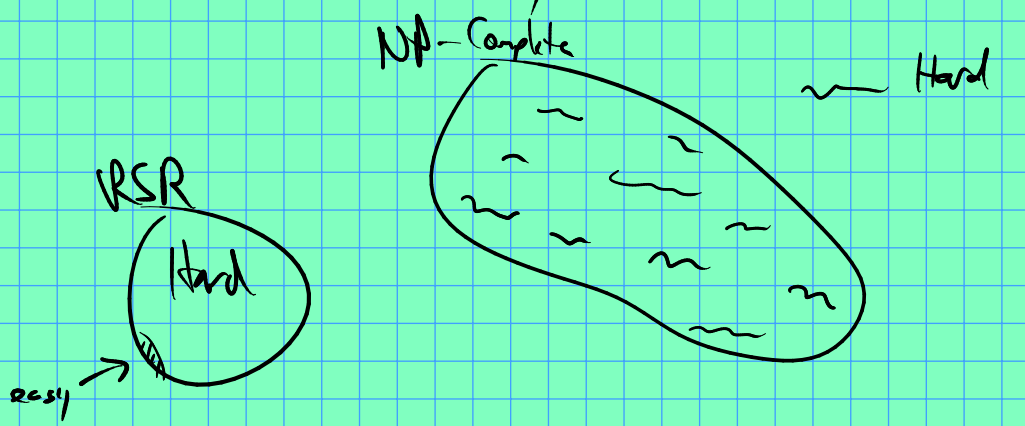
Note: why not use NP-hard problems for crypto assumptions?

For crypto, what you need / want is for random instances to be hard.

Not usually true for NP Hard. In fact, really unlikely according to Fortnow et al:

Random instances hard $\approx$ random self-reducible.

↑ gives a way to randomize instances



Example: DLP: Say instance is $y = g^x$.

No matter how x is chosen, instance can be

randomized! $r \in_R \mathbb{Z}_q$.

and compute $y' = y g^r \; (= g^{x+r \bmod q})$.

if r is random (unif. dist.) so is $x+r \bmod q$!

But solving y' is good enough!

$$\log_g(y') - r = x.$$

Takeaway: if DLP is hard, it's hard pretty much everywhere.

Even $q/poly$ fraction being easy would make all instances easy.

Feigenbaum + Fortnow's result: NP complete + RSR $\overset{\circledast}{\Rightarrow}$ poly hierarchy collapses!

$\circledast$ see "On the random-self-reducibility of complete sets"

formalizing #3:

$A \xrightarrow{m_0 = 000000,\; m_1 = 111111} C$

$A \xleftarrow{E(m_b)} C$

$= bbbb..b$

A wins w/ prob 1. ✓ $\qquad \frac{1}{2} + \frac{1}{poly(\ell)}$

#4 If D = decryption algo, how big must $|D^{-1}(x)|$ be (for eff. computable x)?

$D^{-1}(x)$ ≡ ciphertexts that decrypt to x

So it contains all possible encryptions of x.

If x is eff. computable, $D^{-1}(x)$ must be large! (not poly sized in ℓ = sec. param)

$$E_{pk}(0) \underset{PPT}{\approx} E_{pk}(1)$$

Note: Vanilla RSA (or anything w/ deterministic enc.) has $|D^{-1}(x)| = 1$