ElGamal encryption

p, g don't need to be random

Bob chooses

- large prime p, primitive root $g \pmod p$
- random x, $h := g^x \pmod p$

public key:	private key:
p, g, h	x

DH: p, g
A: $y \rightsquigarrow g^y$
B: $x \rightsquigarrow g^x$
shared secret $g^{xy} \pmod p$

Alice encrypt: $c = (c_1, c_2)$

$c_1 = g^y \pmod p$ where she selects y at random

$c_2 = h^y m \pmod p$ ($h^y = g^{xy}$)

Bob decrypts: $m \equiv c_2 c_1^{-x} \pmod p$ ($c_1^{-x} = g^{-xy}$)

EG Bob's public ElGamal key $(p = 31,\ g = 11,\ h = 6)$

- encrypt $m = 3$ ("random" choice $y = 4$)

$c_1 = 11^4 \pmod{31} \equiv 9$

$c_2 = 6^4 \cdot 3 \pmod{31} \equiv 13$

$\Rightarrow c = (9, 13)$

- determine private key x

$h = g^x \pmod p$

$6 = 11^x \pmod{31}$ discrete log! hard!

$\rightsquigarrow x = 5$

- decrypt $c = (9, 13)$

$m = c_2 c_1^{-x} \pmod p$

$13 \cdot 9^{-5} \pmod{31}$

$\equiv 3$

Format!

$\equiv 9^{25}$