Elgamal encryption algorithm

Prime p and generator g are public keys of Bob

Alice:

Bob:

+ public key

$Y = g^x \bmod p$

Chooses random k

Chooses private x

Calculates

$K = Y^k \bmod p$

calculates

$C_1 = g^k \bmod p$

$C_2 = M\,K \bmod p$

(C_1, C_2)

Calculates $C_1^x \bmod p$

$= K$

and recovers message

$M = K^{-1}C_2 \bmod p$

K^{-1} = the inverse of K mod p

Elgamal = Diffie Hellman key exhange + encryption by multiplying mod p

Elgamal example

Alice sends a message M = 100 to Bob

Prime p = 139 and g =3

Alice:

public key

$44 = 3^{12} \bmod 139$

Bob:

Chooses k = 52

Chooses private x = 12

Calculates

$K = 44^{52} \bmod 139 = 112$

Calculates

$C_1 = 3^{52} \bmod 139 = 38$

$C_2 = 100*112 \bmod 139 = 80$

$(C_1, C_2)=(39,80)$

Calculates $K = 38^{12} \bmod 139 = 112$

$K^{-1} = 112^{-1} \bmod 139 = 36$

and recovers message

$M = K^{-1}C_2 \bmod p =$

$36*80 \bmod 139 = 100$

Elgamal = Diffie Hellman key exhange + encryption by multiplying mod p

Elgamal security

- Each user has a private key x
- Each user has three public keys: prime modulus p, generator g and public $Y = g^x \bmod p$
- Security is based on the difficulty of DLP
- Secure key size > 1024 bits (today even 2048 bits)
- Elgamal is quite slow, it is used mainly for key authentication protocols
- Now widely used, but Elliptic Curve variant is increasingly popular