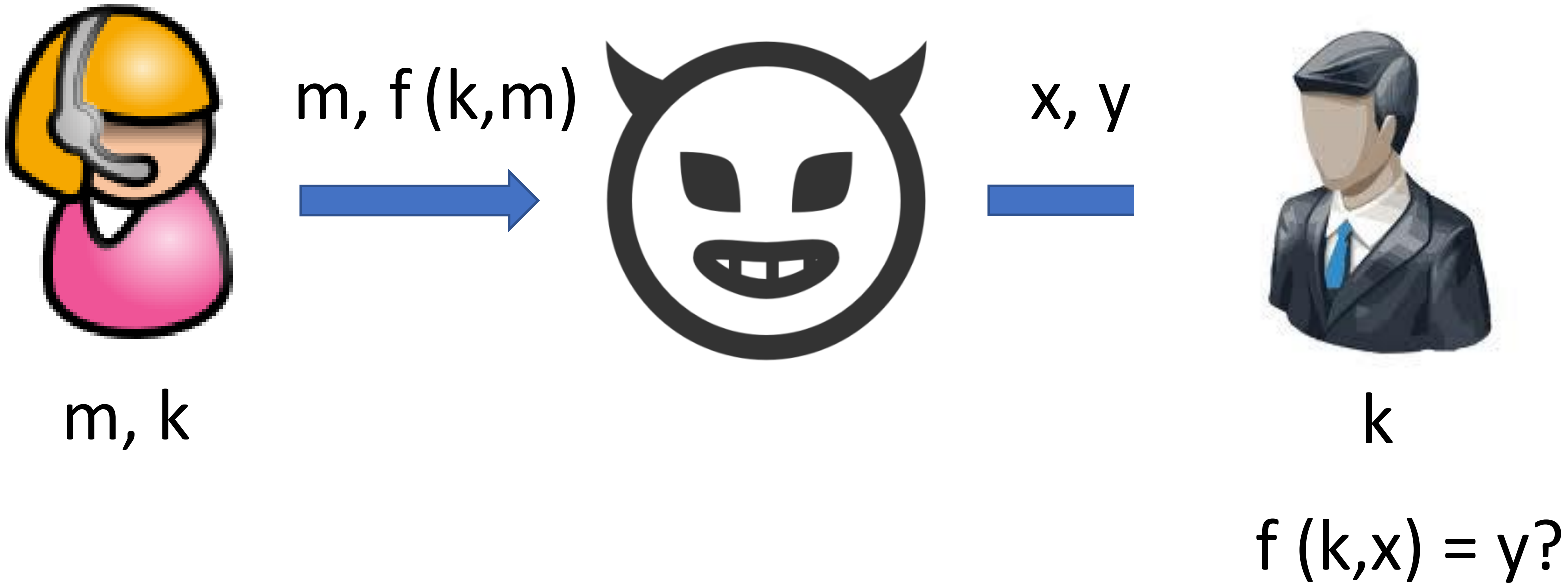
Ten Minute Review of Crypto Primitives

CS 628A

Pramod Subramanyan

What is a MAC?



m, f (k,m)

x, y

m, k

k

f (k,x) = y?

MAC: message authentication code

Hash function:
The Swiss army knife of crypto

Popular examples:

- MD5 (has weaknesses, shouldn’t be used)
- SHA-1, SHA-256

Typical construction:

“Merkle-Damgård”

Ralph Merkle

Hash function:

takes any string as input

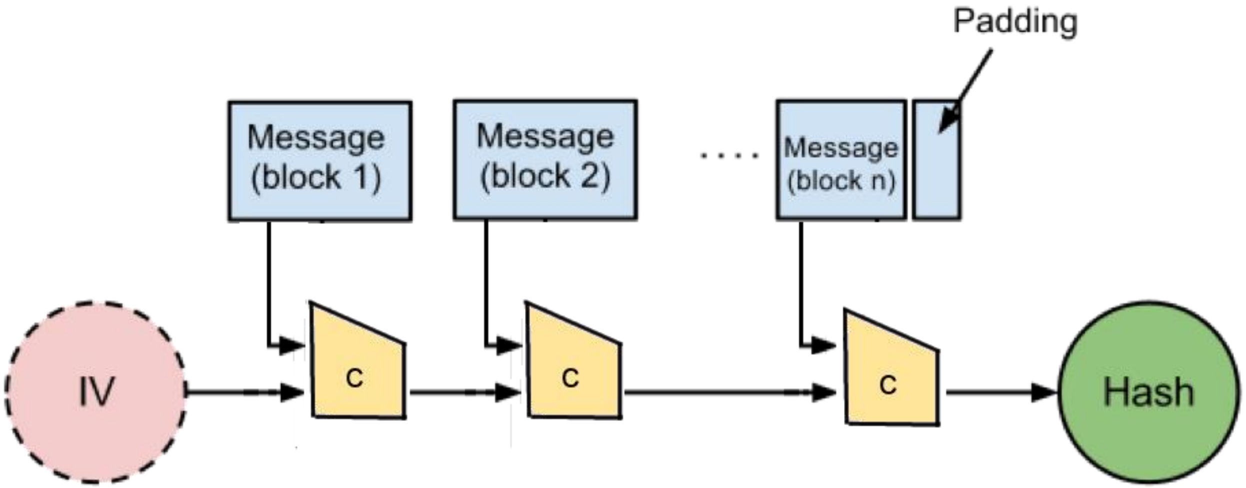
fixed-size output (we’ll use 256 bits)

efficiently computable

Security properties:

collision-free
hiding (preimage resistance)
puzzle-friendly

Merkle-Damgård construction



- Break input into blocks (say 512 bits)
  Pad the last block
- Apply “compression function” to message block together with output of previous stage
- Compression function designed to look really hairy
- IV = initialization vector

Hash-based MAC

Q. Is a Hash(k ‖ msg) a secure MAC?

A. No! “Length-extension attack”

Knowing $f_k(msg)$ *(i.e., f(k || msg))* lets adversary compute $f_k(msg \| app)$ without knowing the key

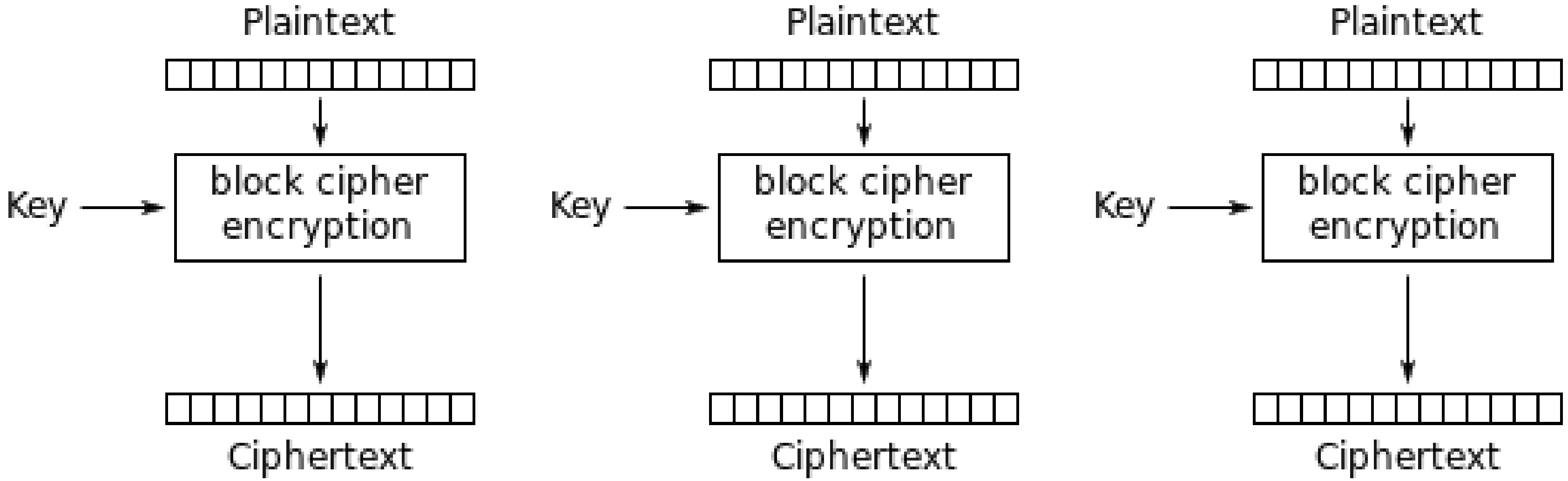
Homework: verify this

How to fix: HMAC

$HMAC(k,m) = H(\underline{k \oplus z_1} \| \underline{H(k \oplus z_2 \| m)})$

z_1 and z_2 are constants

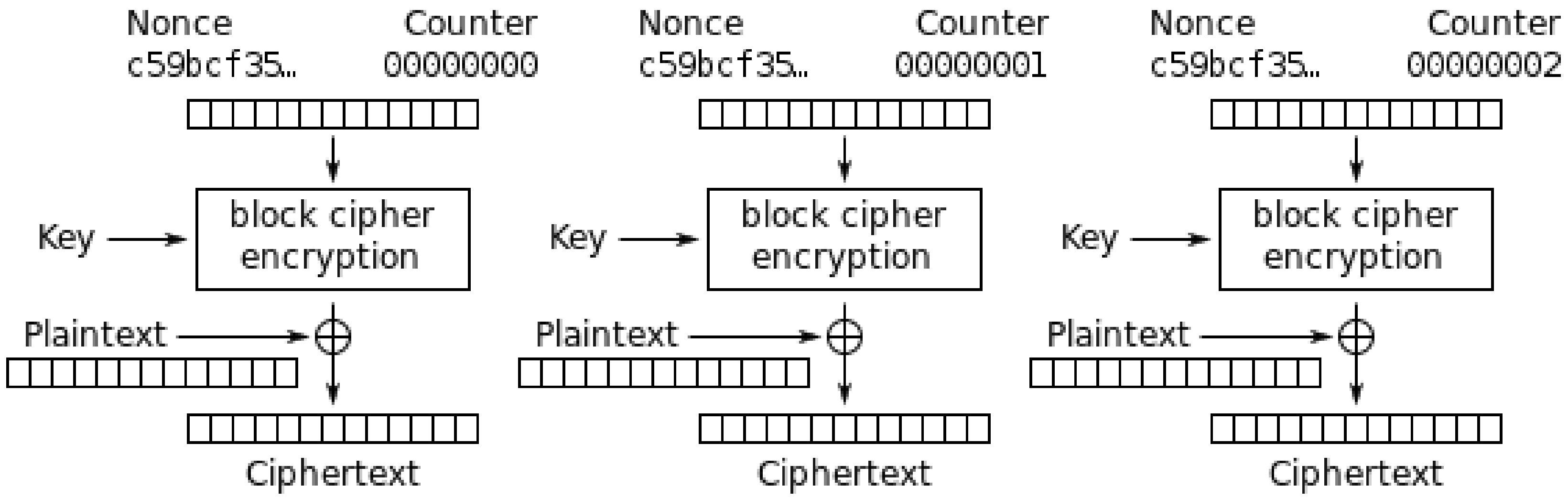
Block ciphers



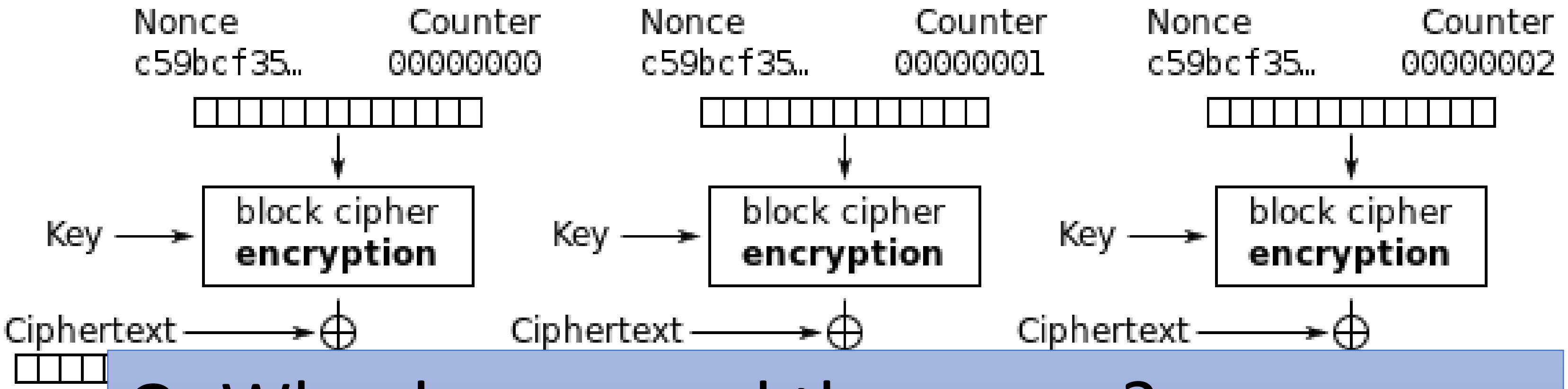
Electronic Codebook (ECB) mode encryption

Question: What is the problem with ECB?

Same input block results in the same output block



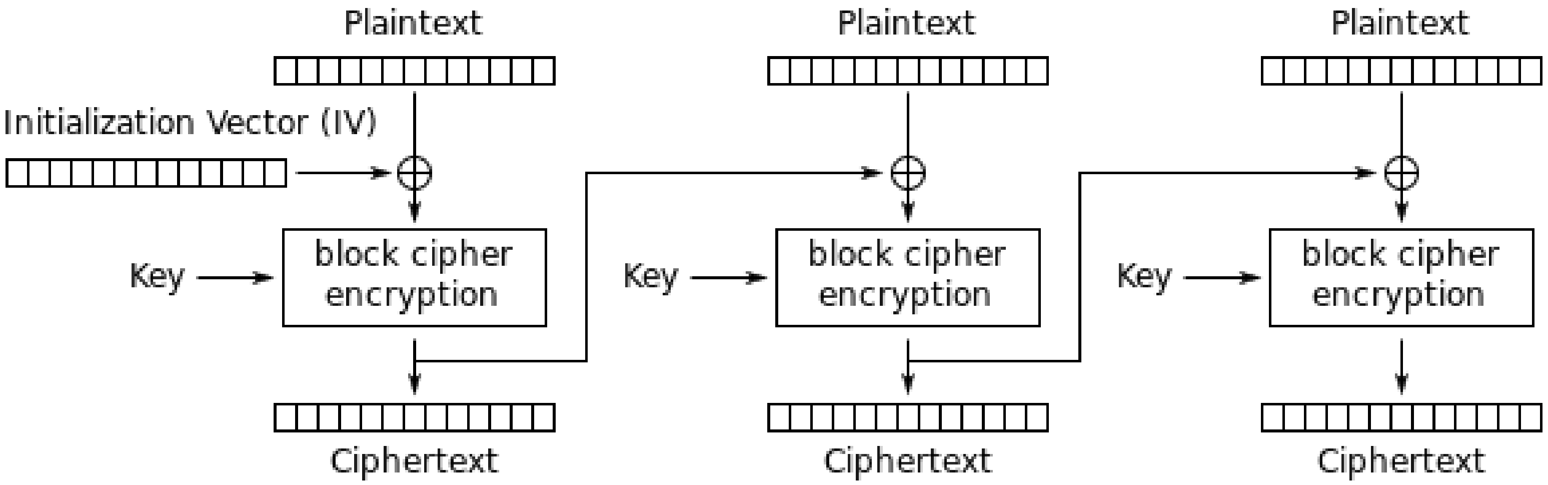
Counter (CTR) mode encryption



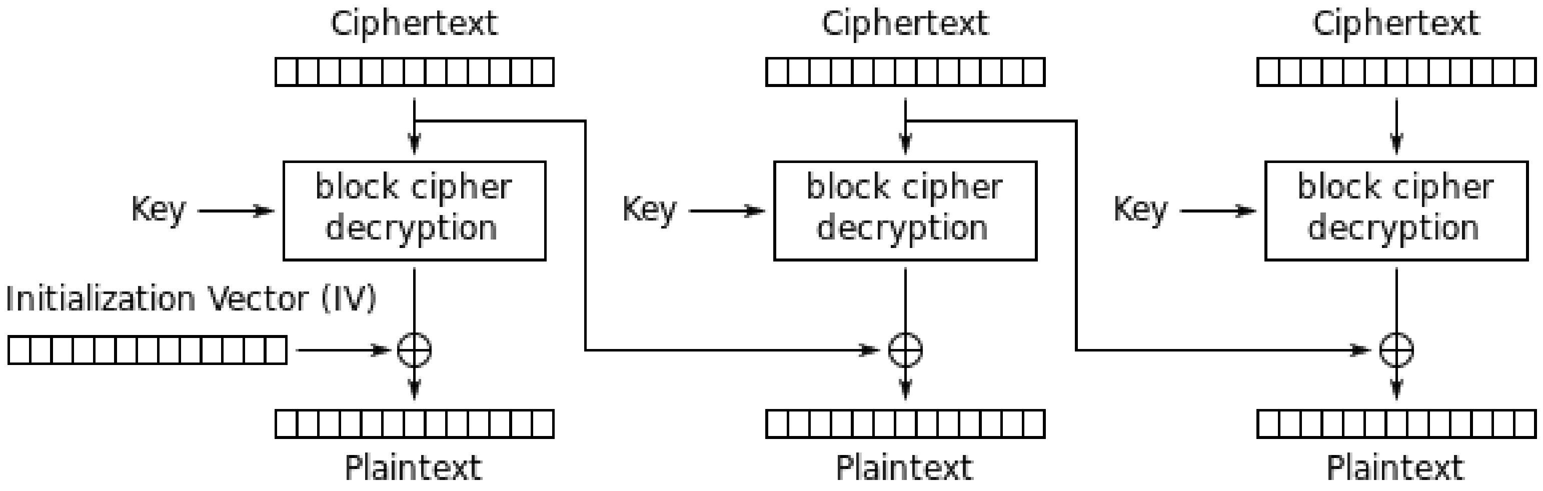
Q: Why do we need the nonce?

A: Almost as bad as ECB without the nonce

Counter (CTR) mode decryption



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

RSA function

Large random primes

- Alice generates N = pq and
  e relatively prime to (p-1)(q-1)
- Euclid's algo to find d s.t.
  ed % (p-1)(q-1) = 1
- Publishes (N, e). Keeps (d, p, q) secret

- RSA(N, e, x) = x^e % N
  RSA(N, d, y) = y^d % N

Inverses

Trapdoor permutation

- Permutation
  Easy to compute

- Hard to invert
  Except if trapdoor is known

RSA Encryption – OAEP encoding

n: RSA modulus length

m: message

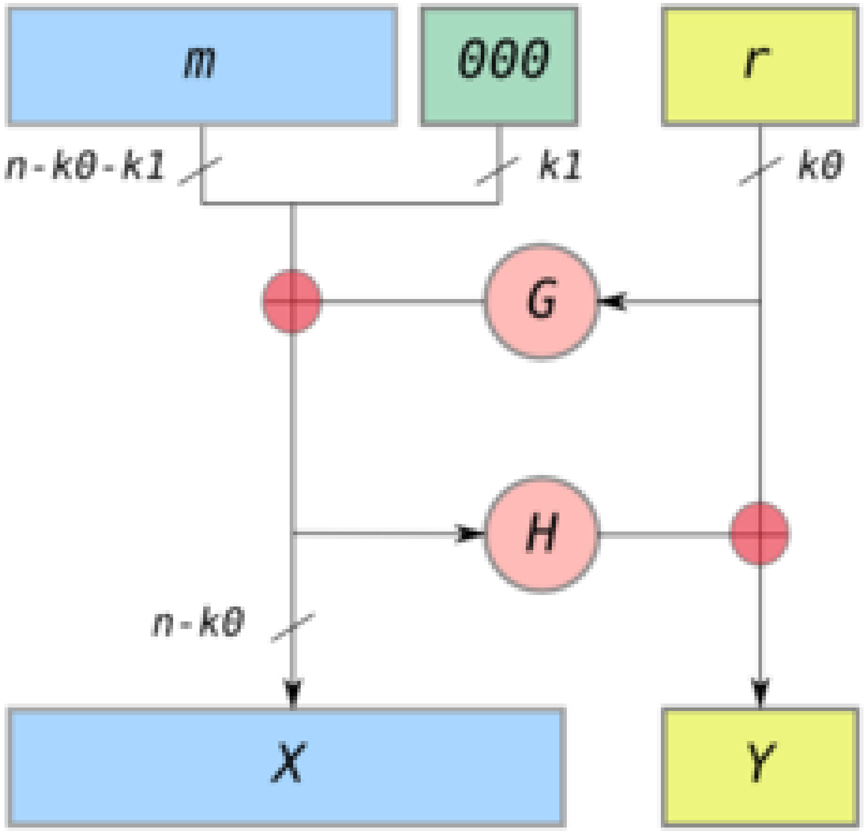
000: padding

r: random nonce

G: PRG

H: hash function

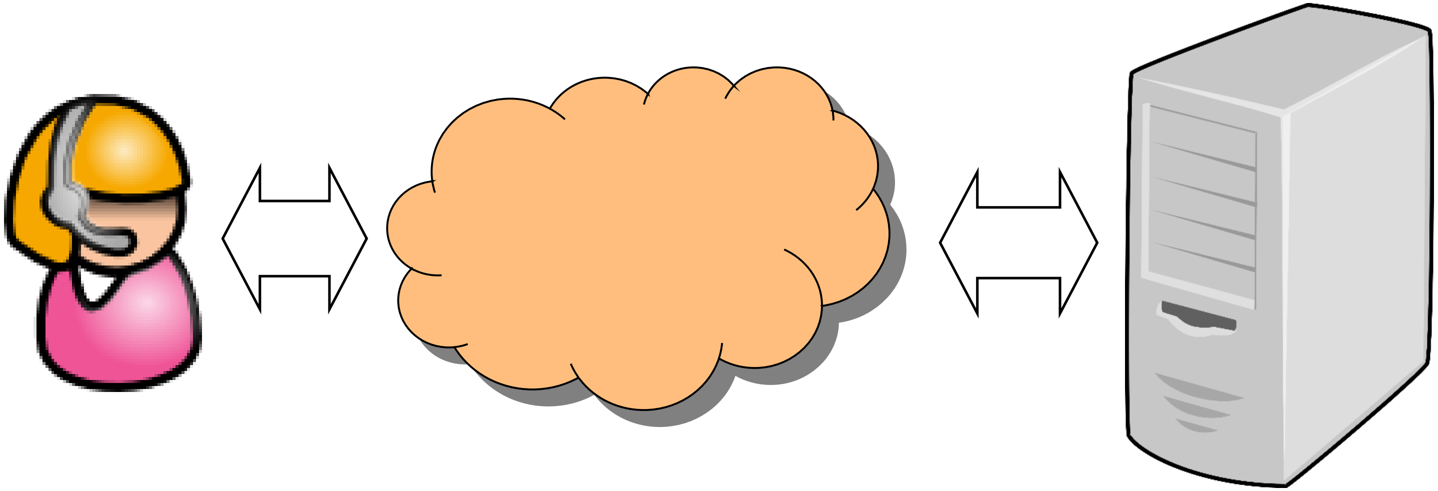
k_0, k_1: 128 bits



RSA input

PKCS #1
https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

Strawman SSL

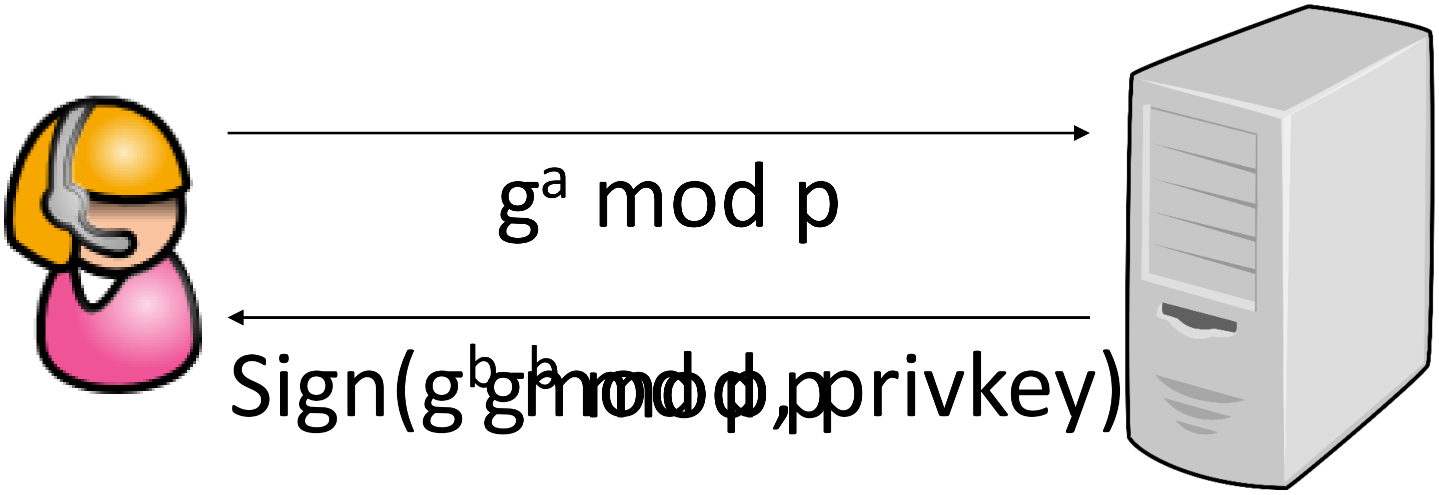


- Alice gets public key of webserver from CA
- Sends session key encrypted using this pubkey
- Server and alice communicate using this key

Problems with this protocol?

- What if server private key is compromised?

Diffie-Hellman Key Exchange



g^a mod p

Sign(g^b mod p, privkey)

g^b mod p

- p is a prime, g is called a generator
- After exchange, both parties know g^{ab} mod p
- More importantly, nobody else knows g^{ab}
- This holds even if privkey is compromised **in future**
- Satisfies property of forward secrecy