

# **650.445/600.454: Practical Cryptographic Systems**

## **Symmetric Cryptography II & Asymmetric Cryptography**

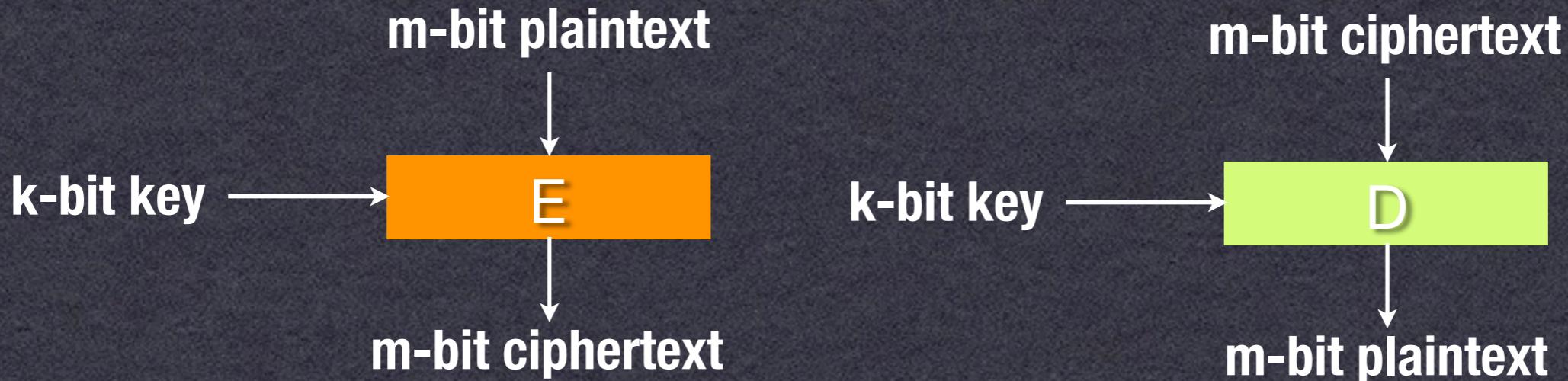
**Instructor: Matthew Green**

# Review

- Last time:
  - Historical tour of Cryptography
  - Classical Crypto, Mechanical ciphers
  - One Time Pads
  - Block ciphers
- Modes of Operation (CBC & CTR)

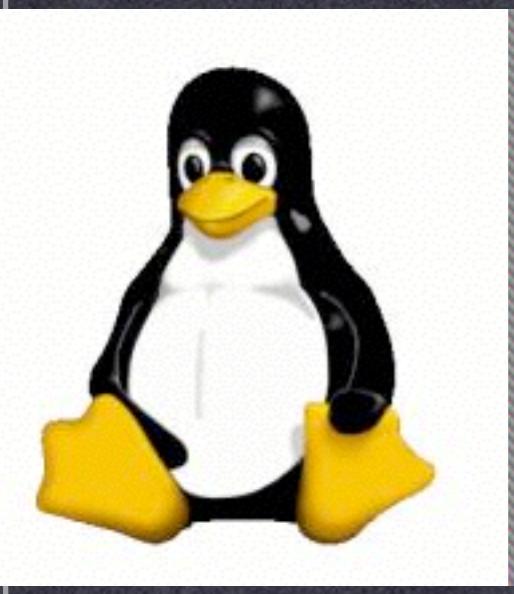
# Review

- Review: Block Ciphers
  - Keyed primitive
  - Operate on fixed-size input blocks
  - Invertible (pseudo-random permutation)
  - Examples: DES, AES...



# Defining Security

- What we'd like from our encryption scheme?
  - Ciphertext doesn't leak any info about plaintext
  - Even if Adversary knows a lot about the plaintext distribution
  - Even if Adversary can choose the plaintext distribution
  - Even if Adversary can obtain encryptions of chosen plaintexts



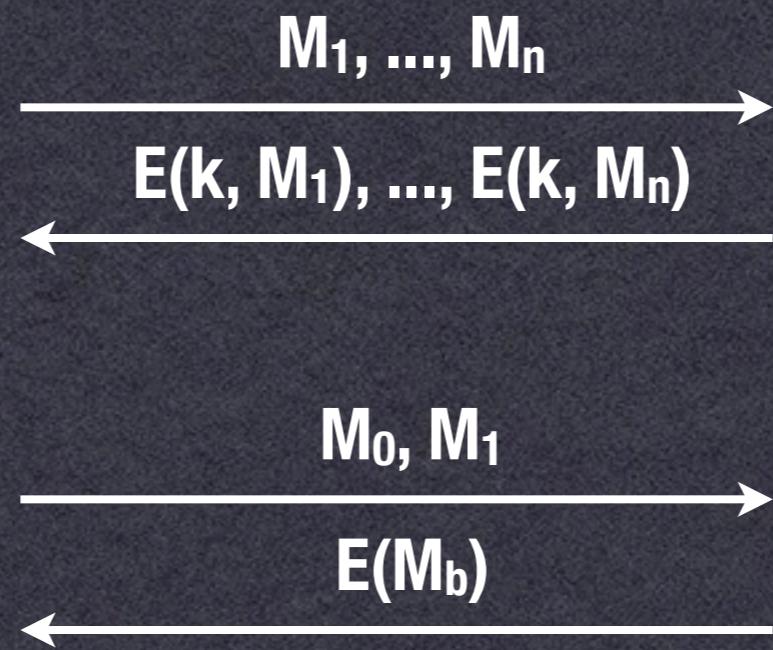
# Review

- Semantic Security (IND-CPA)



Adversary

b?



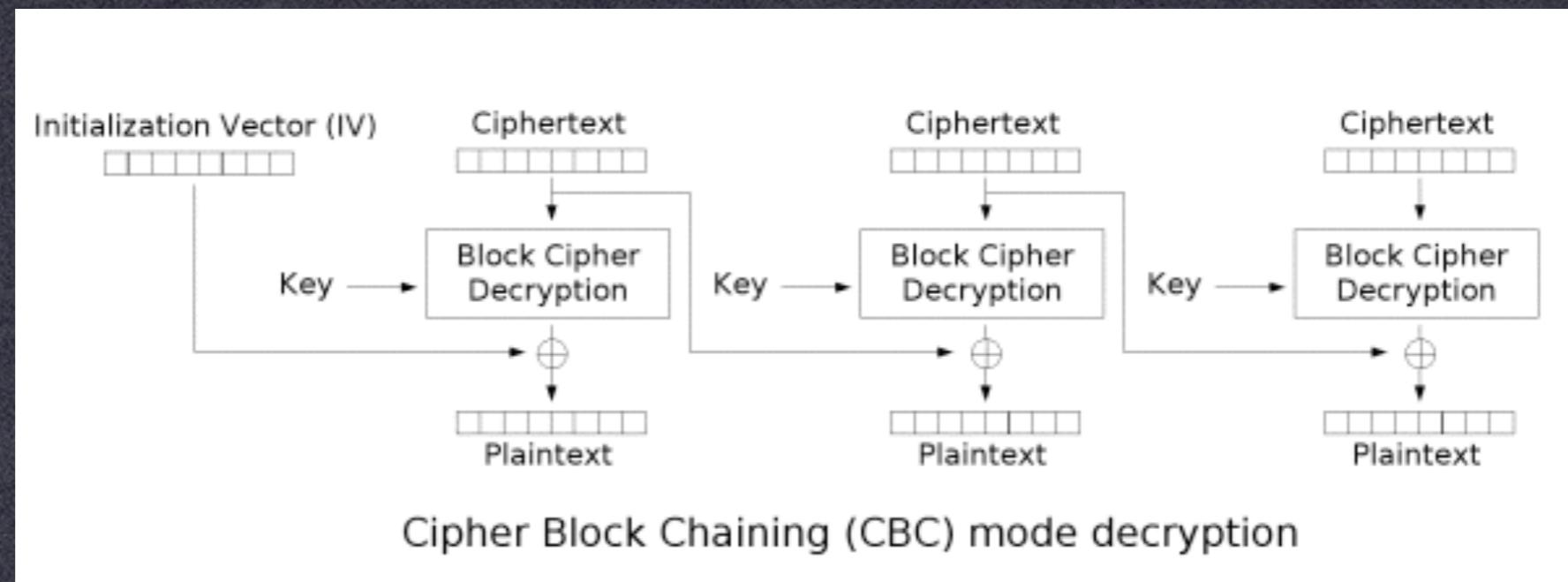
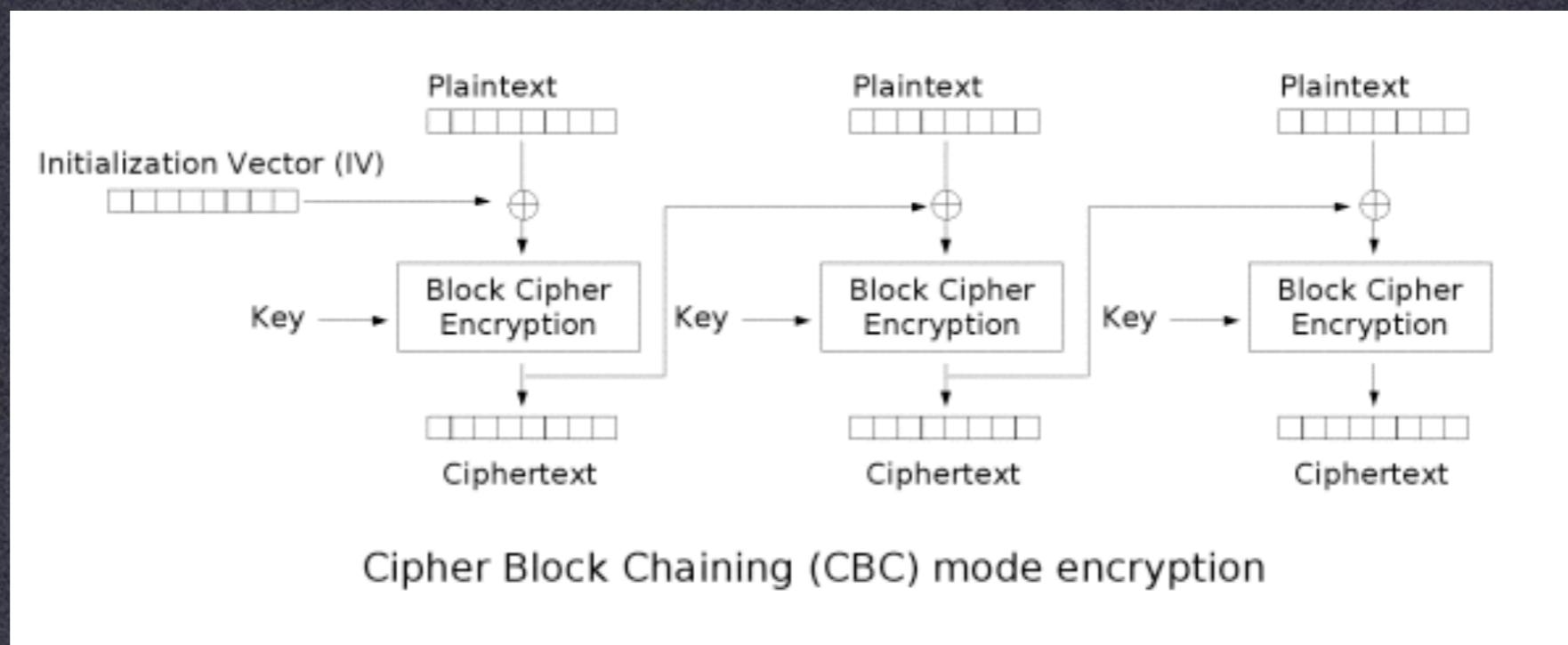
k

$$b \xleftarrow{\$} \{0, 1\}$$

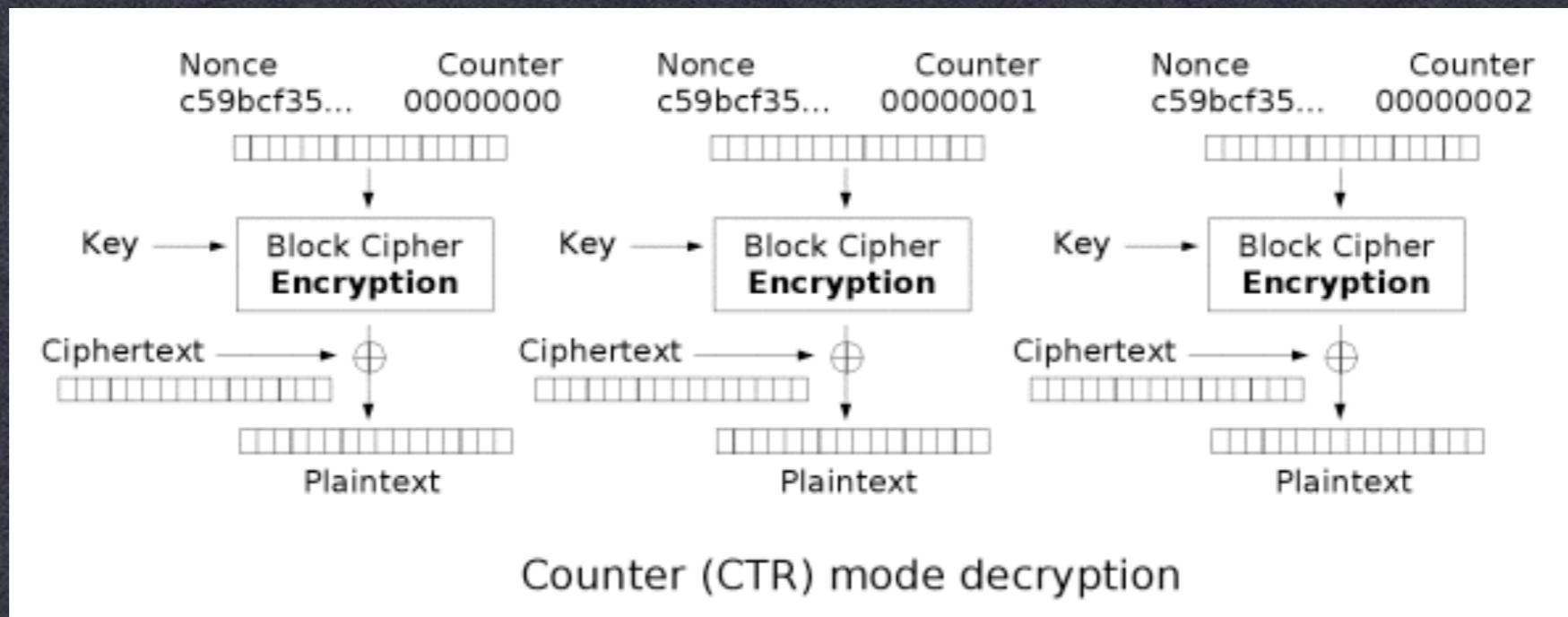
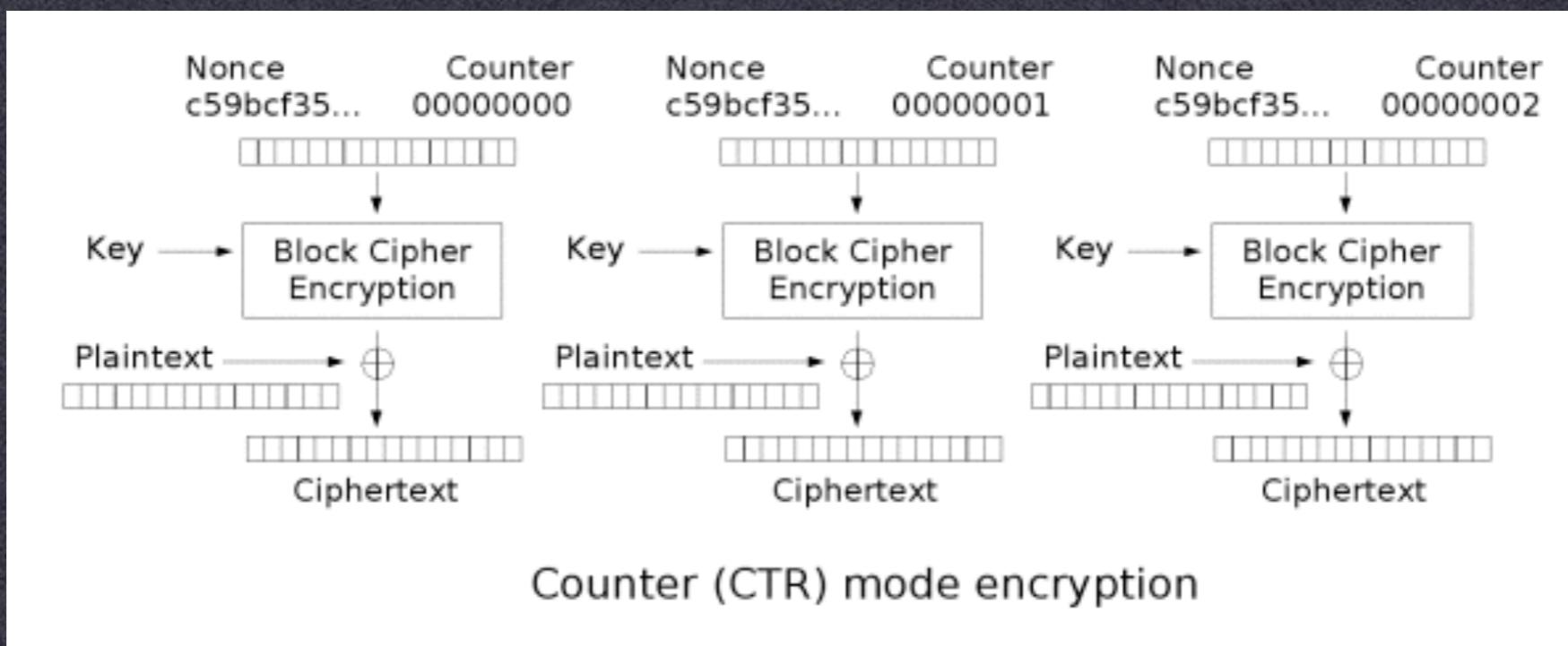
# Using Block Ciphers

- ECB is not semantically secure, hence we use a “mode of operation”
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an Initialization Vector)

# CBC Mode



# CTR Mode



# CTR Mode

- Intuition:
  - We're using the block cipher to generate a stream of “randomish” bits
  - XOR that stream of bits with the message
  - This should look familiar

# CTR Mode

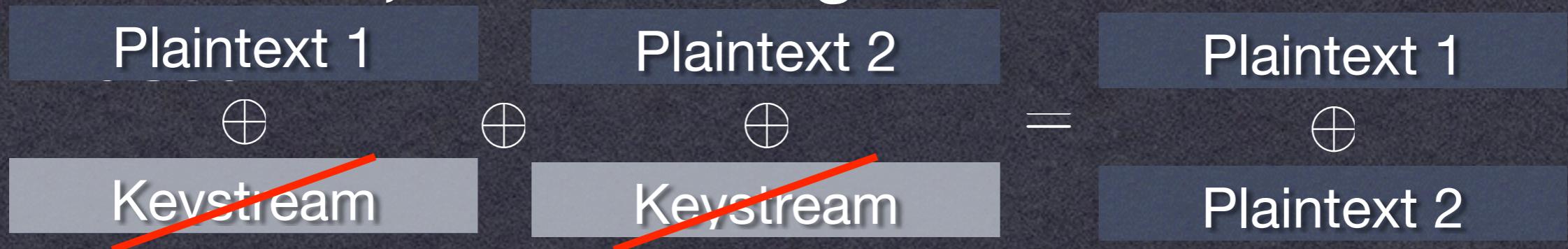
- CTR vs CBC
  - Unlike CBC, CTR can be parallelized (each block of the message encrypted separately by a different processor)
  - CTR keystream can be pre-computed
  - IV is much simpler

# Security of CTR

- IND-CPA assuming secure block cipher (PRP)

# Security of CTR

- IND-CPA assuming secure block cipher (PRP)
- However, counter range must never be re-



- Similar example: MS Word 2003

# “Two-time pad”

- Re-using CTR IV range gives “two time pad”

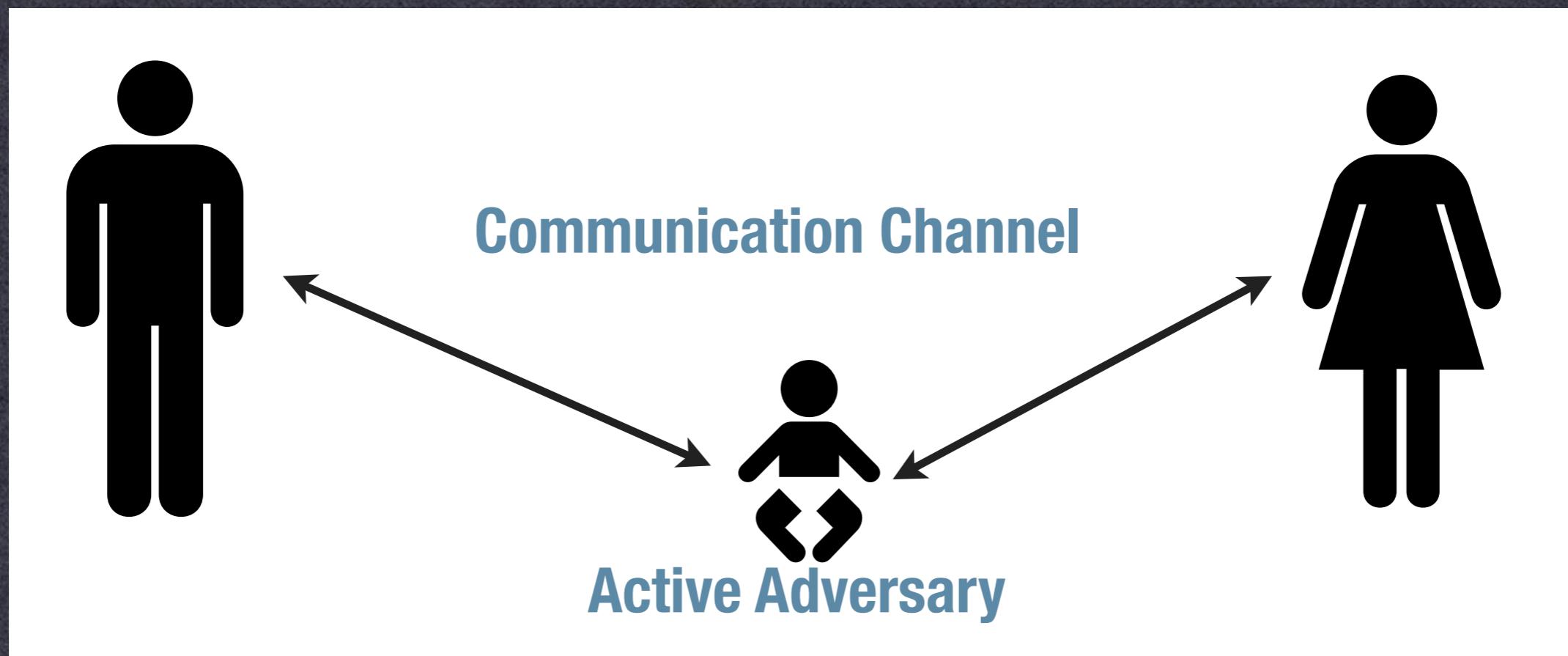
$$\text{Plaintext 1} = \oplus \text{Plaintext 2}$$

- But can we tear these messages apart?
  - Answer: depends on the plaintexts
  - Venona project (hand-decode)
  - Modern NLP techniques (auto-decode)

# Malleability

- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode (bad)
  - CBC Mode (somewhat bad)
- The solution:
  - Authenticated Encryption

# Authenticated Encryption



# MACs

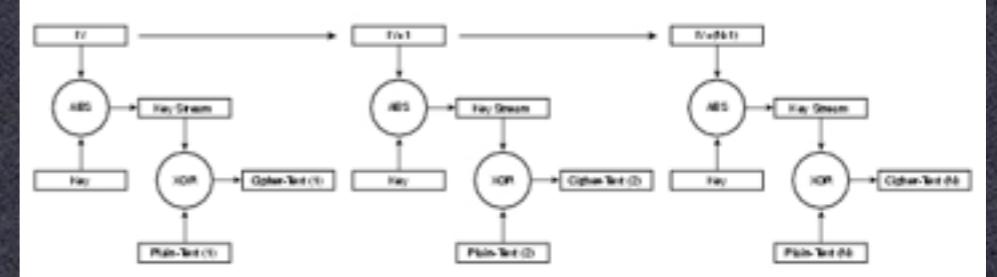
- Symmetric-key primitive
  - Given a key and a message, compute a “tag”
  - Tag can be verified using the same key
  - Any changes to the message detectable
- To prevent malleability:
  - Encrypt then MAC
  - Under separate keys

# MACs

- Definitions of Security
  - **Existential Unforgeability under Chosen Message Attack (EU-CMA)**
- Examples:
  - **HMAC (based on hash functions)**
  - **CMAC/CBC-MAC (block ciphers)**

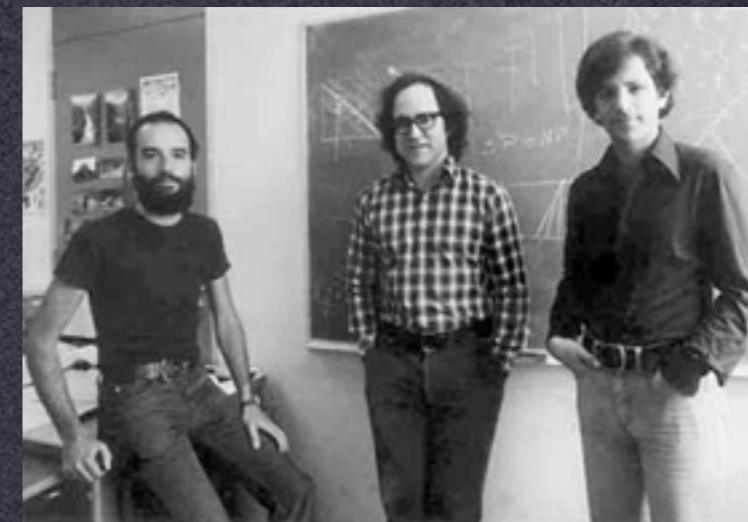
# Authenticated Encryption

- Two ways to get there:
  - Generic composition  
Encrypt (e.g., CBC mode) then MAC
  - two different keys, multiple primitives
  - Authenticated mode of operation
  - Integrates both encryption & authentication
  - Single key, typically uses only one primitive (e.g., block cipher)
  - Ex: CCM, OCB, GCM modes



# Asymmetric Crypto

- So far we've discussed symmetric crypto
  - Requires both parties to share a key
  - Key distribution is a hard problem!



# Key Agreement

- Establish a shared key in the presence of a passive adversary



# D-H Protocol

Malcolm Williamson in 72

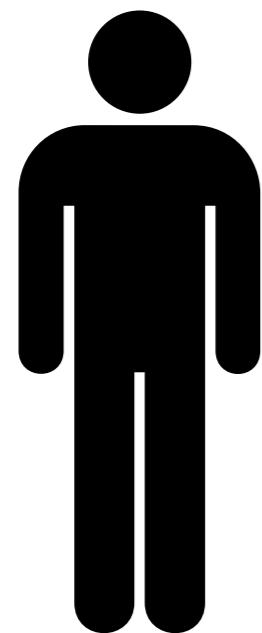
Diffie-Hellman in 76



$$b \in \mathbb{Z}_q$$

$$p, q : p = 2q + 1$$

$$a \in \mathbb{Z}_q$$



$$\xleftarrow{g^b \text{ mod } p}$$

$$g^{ab}$$

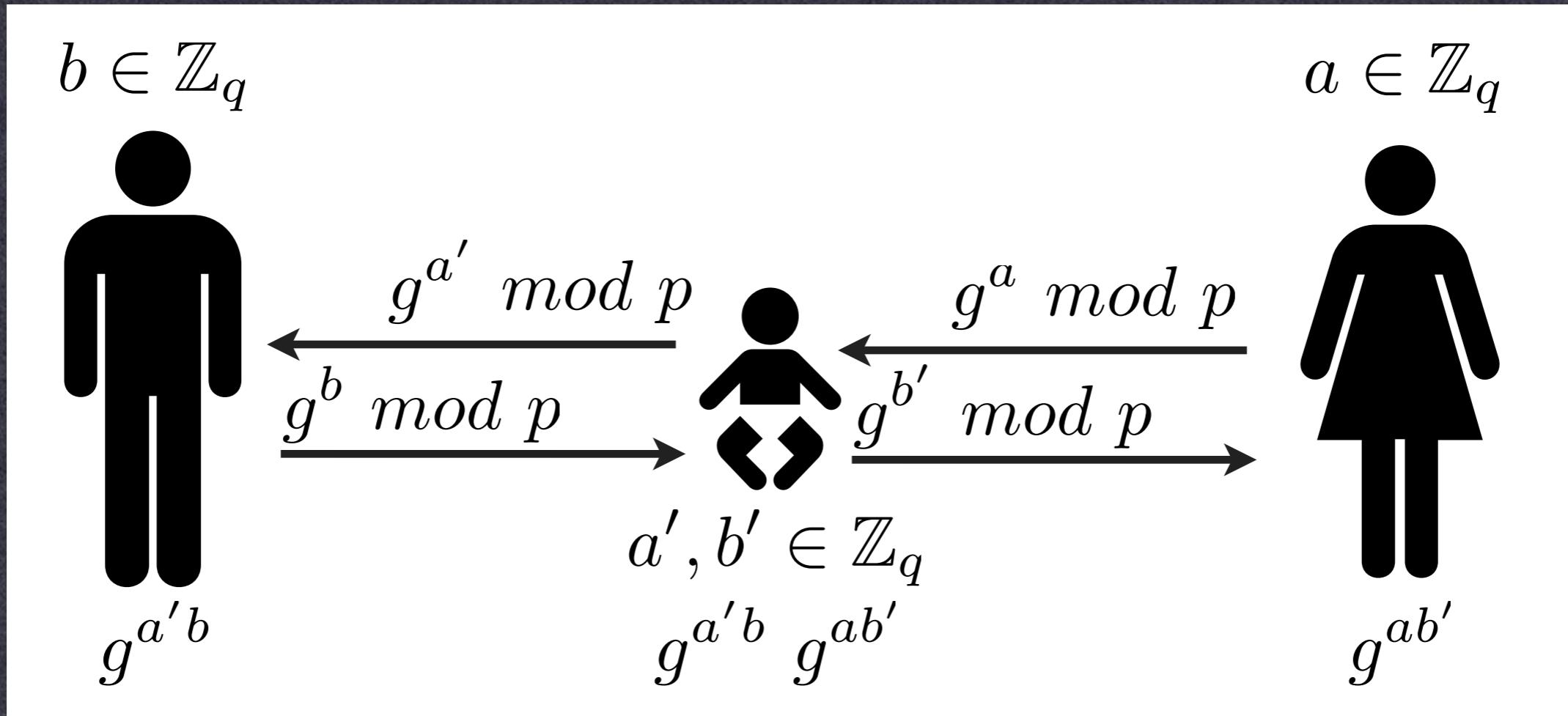
$$\xrightarrow{g^a \text{ mod } p}$$



$$g^{ab}$$

# Man in the Middle

- Assume an active adversary:

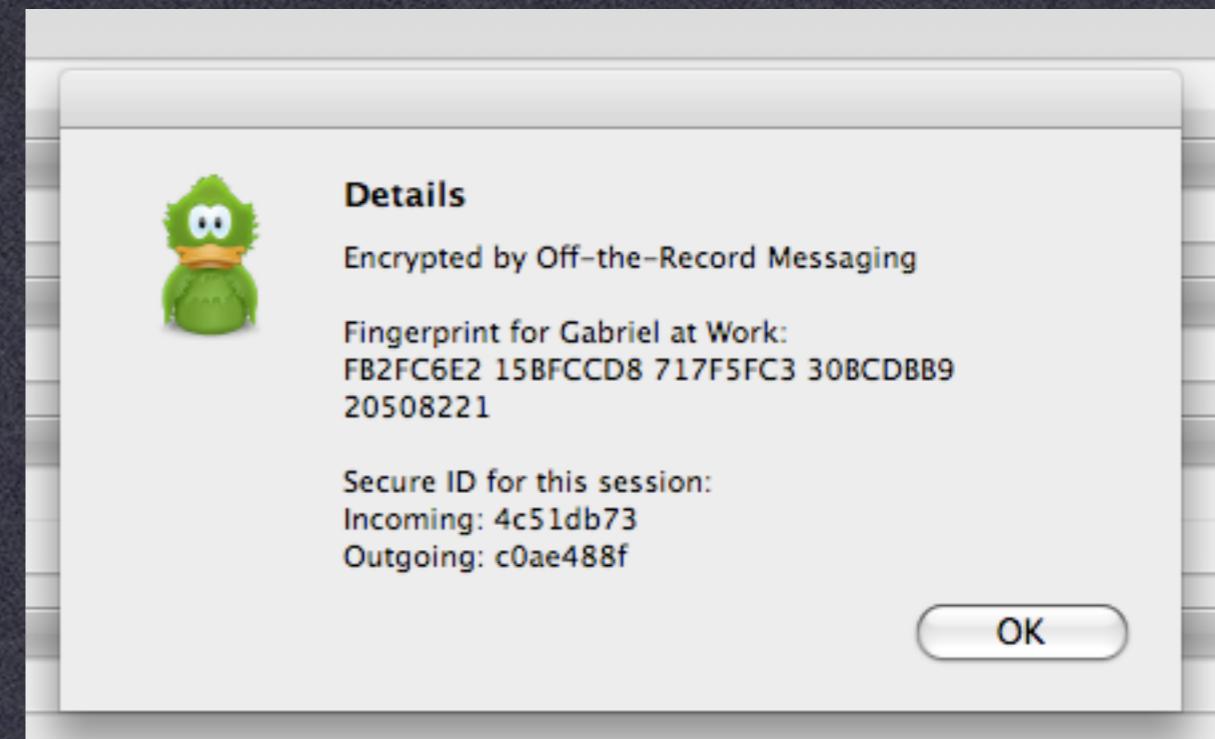


# Man in the Middle

- Caused by lack of authentication
  - D-H lets us establish a shared key with anyone...  
but that's the problem...
- Solution: Authenticate the remote party

# Preventing MITM

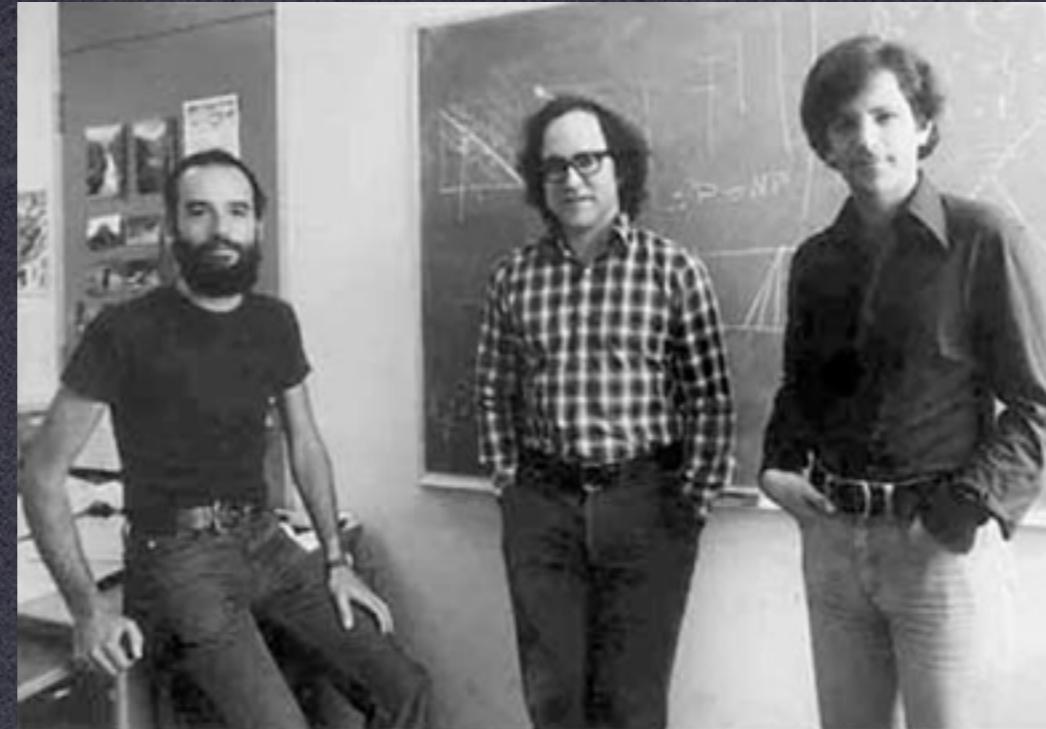
- Verify key via separate channel
- Password-based authentication
- Authentication via PKI



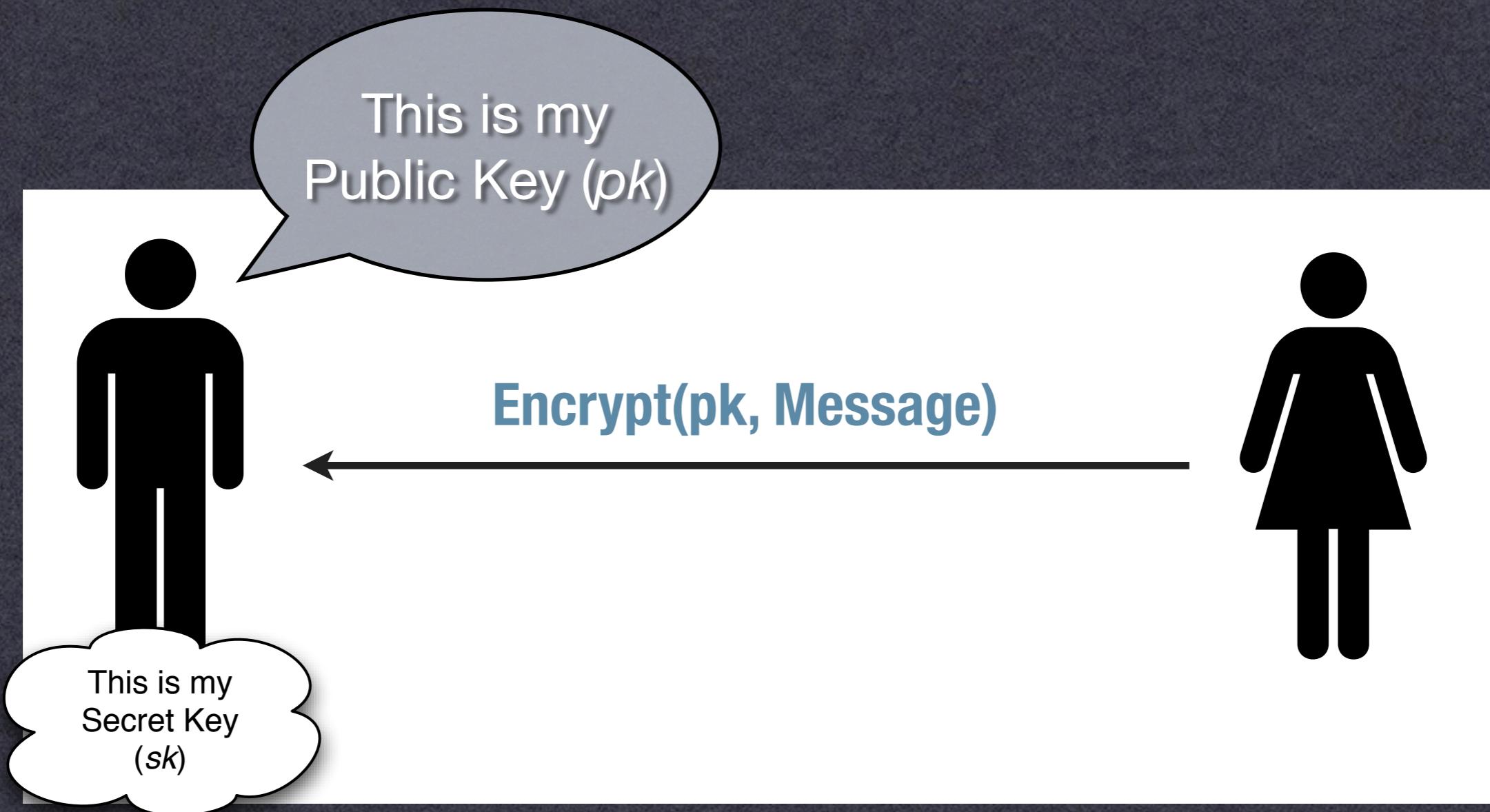
# Public Key Encryption

- What if our recipient is offline?
  - Key agreement protocols are interactive
  - e.g., want to send an email

Ellis in 72, Cocks a few months later



# Public Key Encryption



# RSA Cryptosystem

## Key Generation

**Choose large primes:**  $p, q$

$$N = p \cdot q$$

$$\phi(N) = (p - 1)(q - 1)$$

**Choose:**

$$e : \gcd(e, \phi(N)) = 1$$

$$d : ed \bmod \phi(N) = 1$$

**Output:**

$$pk = (e, N)$$

$$sk = d$$

## Encryption

$$c = m^e \bmod N$$

## Decryption

$$m = c^d \bmod N$$

# “Textbook RSA”

- In practice, we don't use Textbook RSA
  - Fully deterministic (not semantically secure)
  - Malleable

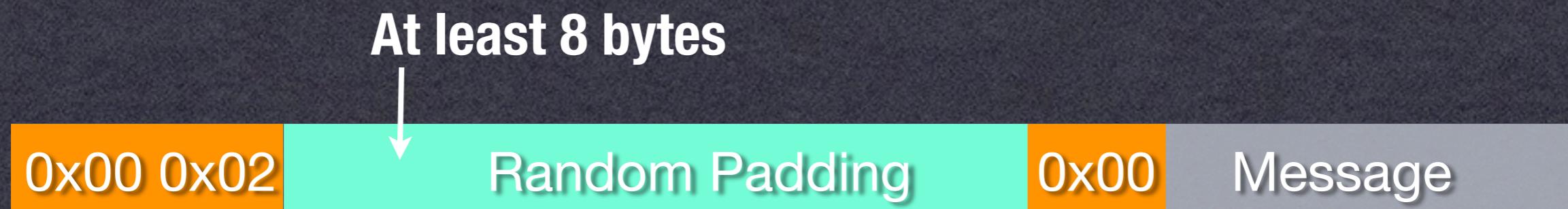
$$c' = c \cdot x^e \bmod N$$

$$c'^d = (m^e \cdot x^e)^d = m \cdot x \bmod N$$

- Might be partially invertible
- Coppersmith's attack: recover part of plaintext  
(when  $m$  and  $e$  are small)

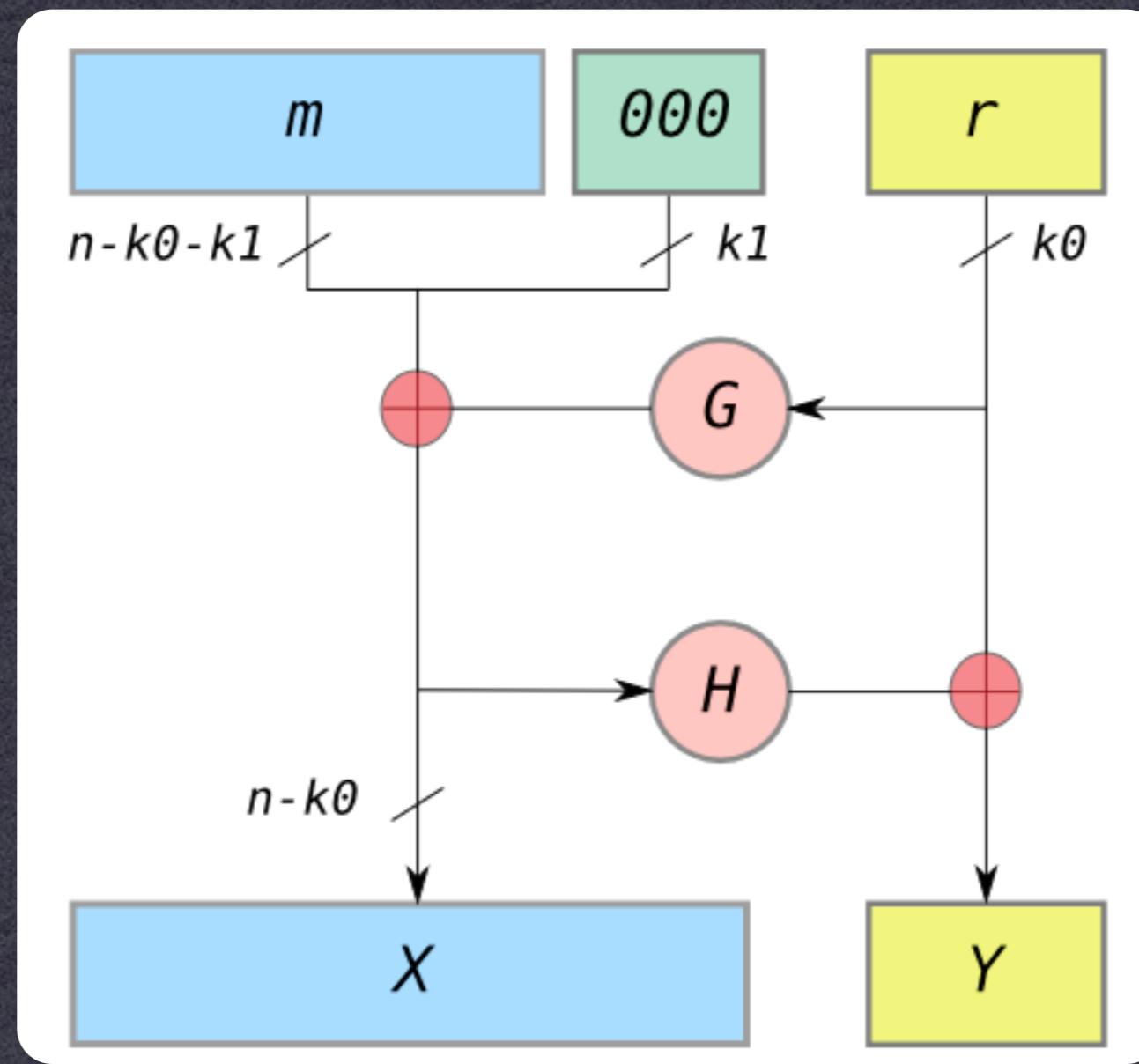
# RSA Padding

- Early solution (RSA PKCS #1 v1.5):
  - Add “padding” to the message before encryption
  - Includes randomness
  - Defined structure to mitigate malleability
  - PKCS #1 v1.5 badly broken (Bleichenbacher)



# RSA Padding

- Better solution (RSA-OAEP):
  - G and H are hash functions



# Efficiency

$m^e \bmod N$   
 $e = 65,537$

$m^d \bmod N$

	Cycles/Byte
AES (128 bit key)	18
DES (56 bit key)	51
RSA (1024 bit key) <u>Encryption</u>	1,016
RSA (1024 bit key) <u>Decryption</u>	21,719

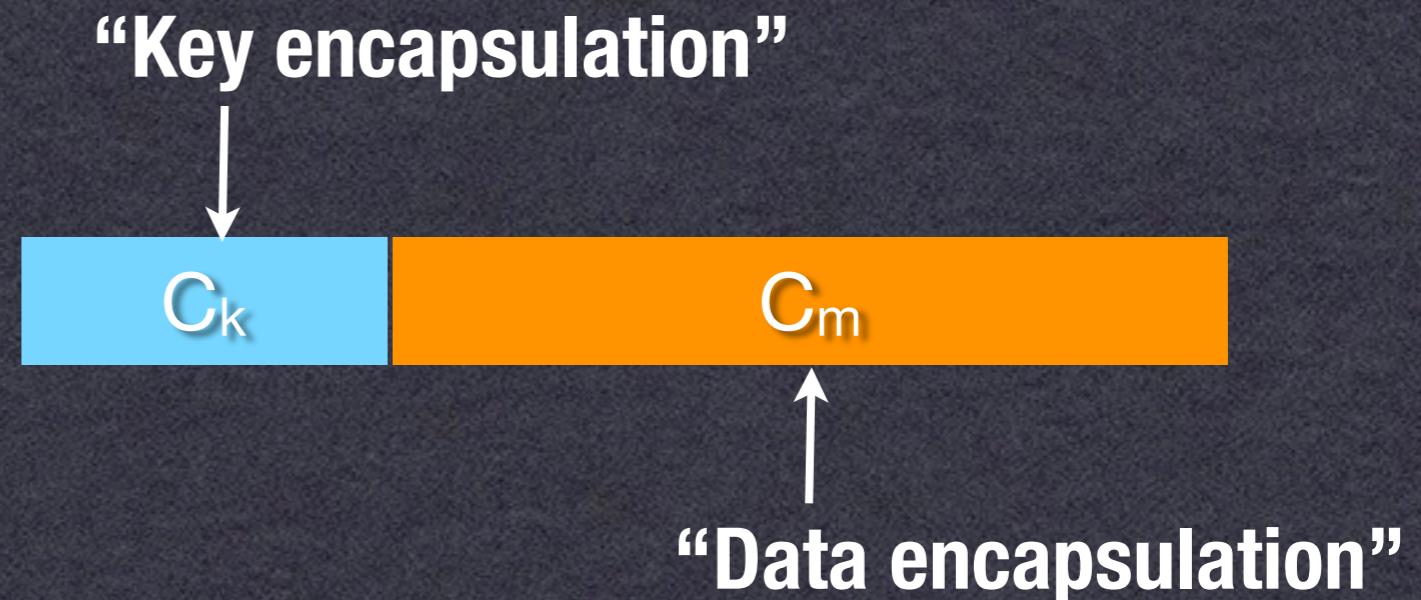
# Hybrid Encryption

- Mixed Approach
  - Use PK encryption to encrypt a symmetric key
  - Use (fast) symmetric encryption on data

$$k \xleftarrow{\$} \{0, 1\}^k$$

$$C_k \leftarrow RSA.Encrypt_{pk}(k)$$

$$C_m \leftarrow AES.Encrypt_k(message)$$



# Key Strength

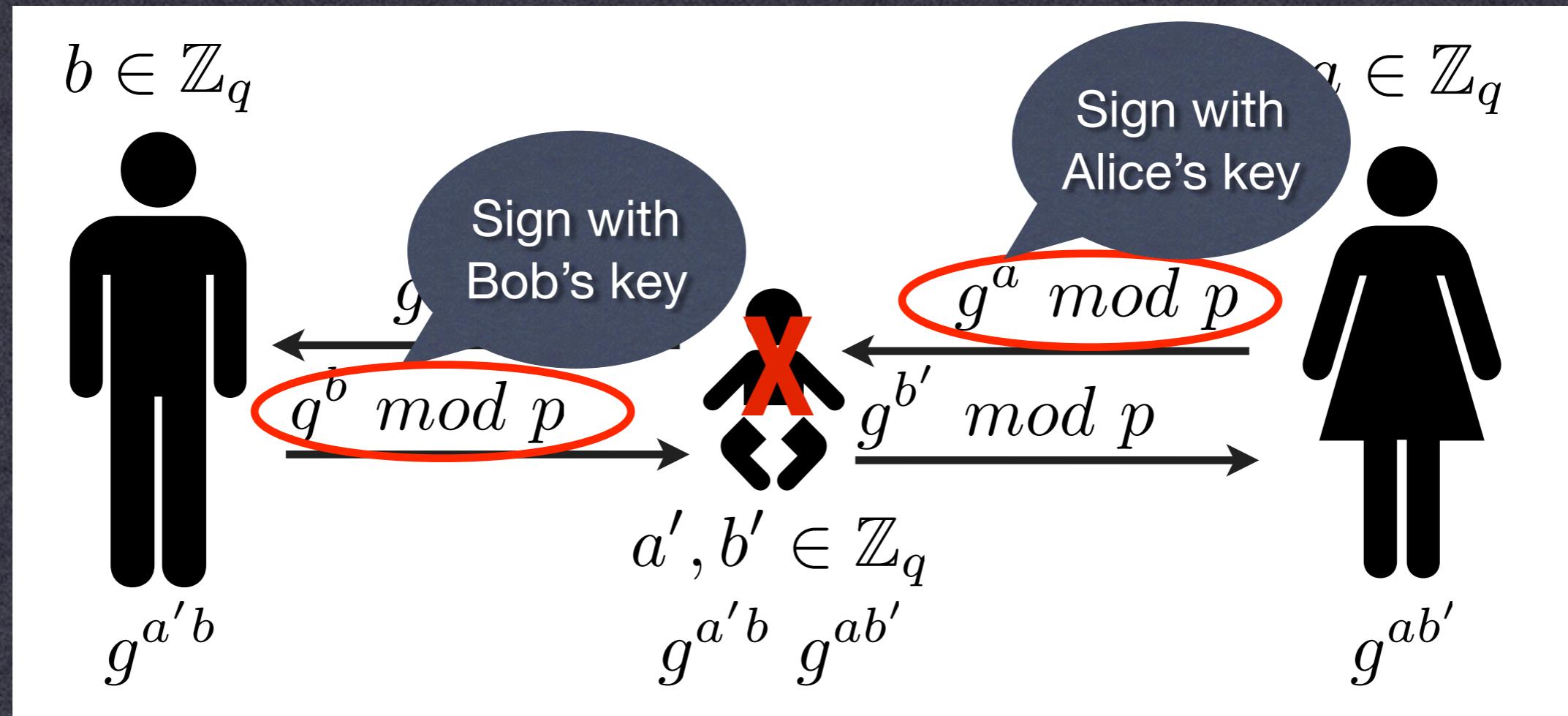
Level	Protection	Symmetric	Asymmetric	Discrete Logarithm Key Group	Elliptic Curve Hash
1	Attacks in "real-time" by individuals <i>Only acceptable for authentication tag size</i>	32	-	- - - -	- - - -
2	Very short-term protection against small organizations <i>Should not be used for confidentiality in new systems</i>	64	816	128 816 128 128	128
3	Short-term protection against medium organizations, medium-term protection against small organizations	72	1008	144 1008 144 144	144
	Very short-term protection against agencies, long-term protection against small organizations				
4	<i>Smallest general-purpose level, Use of 2-key 3DES restricted to <math>2^{40}</math> plaintext/ciphertexts, protection from 2009 to 2011</i>	80	1248	160 1248 160 160	160
	Legacy standard level				
5	<i>Use of 2-key 3DES restricted to <math>10^6</math> plaintext/ciphertexts, protection from 2009 to 2018</i>	96	1776	192 1776 192 192	192
	Medium-term protection <i>Use of 3-key 3DES, protection from 2009 to 2028</i>				
6		112	2432	224 2432 224 224	224
	Long-term protection				
7	<i>Generic application-independent recommendation, protection from 2009 to 2038</i>	128	3248	256 3248 256 256	256
	"Foreseeable future" <i>Good protection against quantum computers</i>	256	15424	512 15424 512 512	512

# Digital Signatures

- Similar to MACs, with public keys
  - Secret key used to sign data
  - Public key can verify signature
  - Advantages over MACs?

# Preventing MitM

- Assume an active adversary:



# PKI & Certificates

- How do I know to trust your public key?
  - Put it into a file with some other info, and get someone else to sign it!



# Next Time

- Protocols & Implementation
- Reading!
- A2 coming up this week