601.445/645 Practical Cryptographic Systems

Asymmetric Cryptography

Instructor: Matthew Green

Housekeeping

- Programming Assignment 2, out tomorrow or Friday (Piazza/BB)
- Weekly written assignment 2 out, 2pm
- Syllabus/Reading:
  - I've been updating this 1 week at a time
  - Reading:
    20 years of RSA
    Imperfect Diffie Hellman
    Mining your Ps and Qs

News?

News?

DAVID NIELD SECURITY 07.08.18 07:00 AM

ALL THE WAYS IOS 12 WILL MAKE YOUR IPHONE MORE SECURE

Encrypted Group Video Chat

Encryption is everywhere in iOS, from the text chats you send through iMessage to the location data logged by apps. Without the passcode or fingerprint or face you've assigned to your iPhone—which act as the decryption keys—the data can't be read.

Stronger Hacking Protections

When someone wants to hack into your iPhone without your permission—whether it's a criminal saboteur or a law enforcement agent—they often do so via some kind of brute-force approach, making multiple attempts at entry in quick succession.

With iOS 12, Apple is drastically narrowing the window of time in which that can be effective. If an iPhone isn't unlocked for an hour, it will switch the Lightning port to a charging only state, neutralizing attempts to pull data from it.

News?

iOS Security
iOS 12

September 2018

Data saved to the file system by the Secure Enclave is encrypted with a key entangled with the UID and an anti-replay counter. The anti-replay counter is stored in a dedicated nonvolatile memory **integrated circuit (IC)**.

On devices with A12 and S4 SoCs, the Secure Enclave is paired with a secure storage integrated circuit (IC) for anti-replay counter storage. The secure storage IC is designed with immutable ROM code, a hardware random number generator, cryptography engines, and physical tamper detection. To read and update counters, the Secure Enclave and storage IC employ a secure protocol that ensures exclusive access to the counters.

News?

iOS Security

iOS 12

September 2018

Anti-replay services on the Secure Enclave are used for revocation of data over events that mark anti-replay boundaries including, but not limited to, the following:

- Passcode change
- Touch ID or Face ID enable/disable
- Touch ID fingerprint add/delete
- Face ID reset
- Apple Pay card add/remove
- Erase All Content and Settings

The Secure Enclave is also responsible for processing fingerprint and face data from the Touch ID and Face ID sensors, determining if there's a match, and then enabling access or purchases on behalf of the user.

News?

iOS Security
iOS 12

September 2018

Pointer Authentication Codes

Pointer authentication codes (PACs) are used to protect against exploitation of memory corruption bugs. System software and built-in apps use PAC to prevent modification of function pointers and return addresses (code pointers). Doing so increases the difficulty of many attacks. For example, a Return Oriented Programming (ROP) attack attempts to trick the device into executing existing code maliciously by manipulating function return addresses stored on the stack.

PAC is supported on A12 and S4 SoCs.

Review: Generic AE

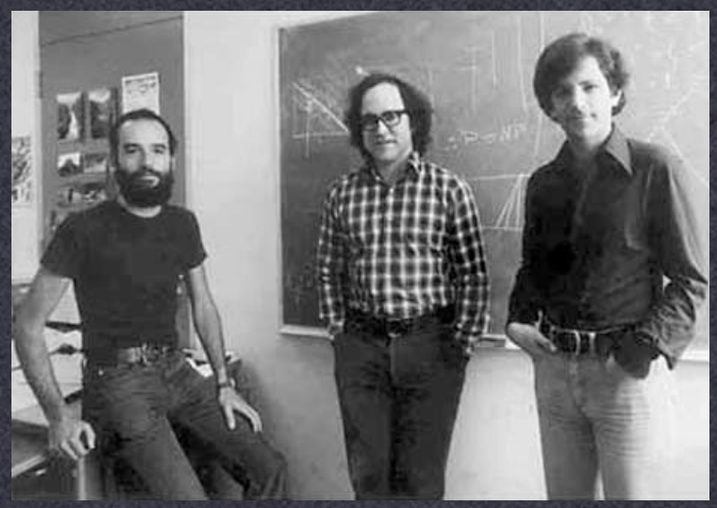
- Encrypt-and-MAC (?)
- Encrypt-then-MAC (?)
- Mac-then-Encrypt (?)

Review: Hash functions

- What are the properties of a hash function?

Asymmetric Crypto

- So far we've discussed <u>symmetric</u> 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

(Cyclic) groups

- An abelian group is:
  - A finite set of elements G and a (binary) group operation ∘

<u>Closure</u>: A∘B is an element of G if A,B are
<u>Commutative:</u> A∘B=B∘A for all A,B in G
<u>Identity</u>: There is an I in G s.t. for all A,

$$A \circ I = I \circ A = A$$

<u>Associativity</u>: (A∘B)∘C = A∘(B∘C)
<u>Inverses</u>: For each A in G, there exists A^{-1} s.t.:

$$A \circ A^{-1} = A^{-1} \circ A = I$$

(Cyclic) groups

- An cyclic group is an abelian group s.t.:
  - There exists a generator g in G such that G generates the group

    I.e., for integer n, every element can be represented as $g^n = (g \circ g \circ \ldots \circ g)$ n times

    Example: set of integers modulo p that are relatively prime with n (Z*p)

    p = 7: 1, 2, 3, 4, 5, 6
    Order of the group is phi(p) = p-1

D-H Protocol

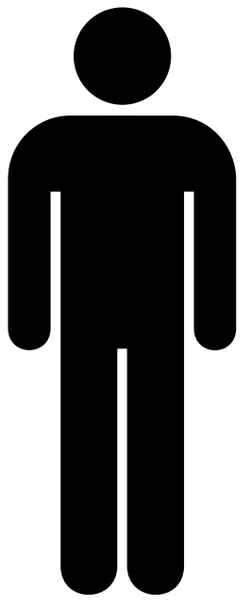
Malcolm Williamson in 72

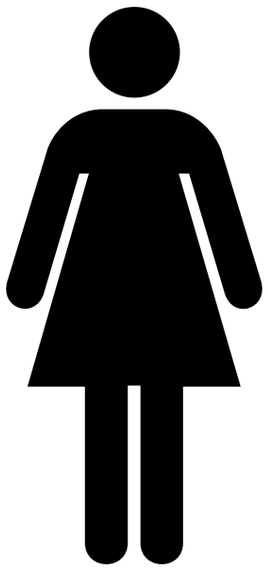
Diffie-Hellman in 76

$b \in \mathbb{Z}_q$

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

$a \in \mathbb{Z}_q$





$g^a \ mod \ p$

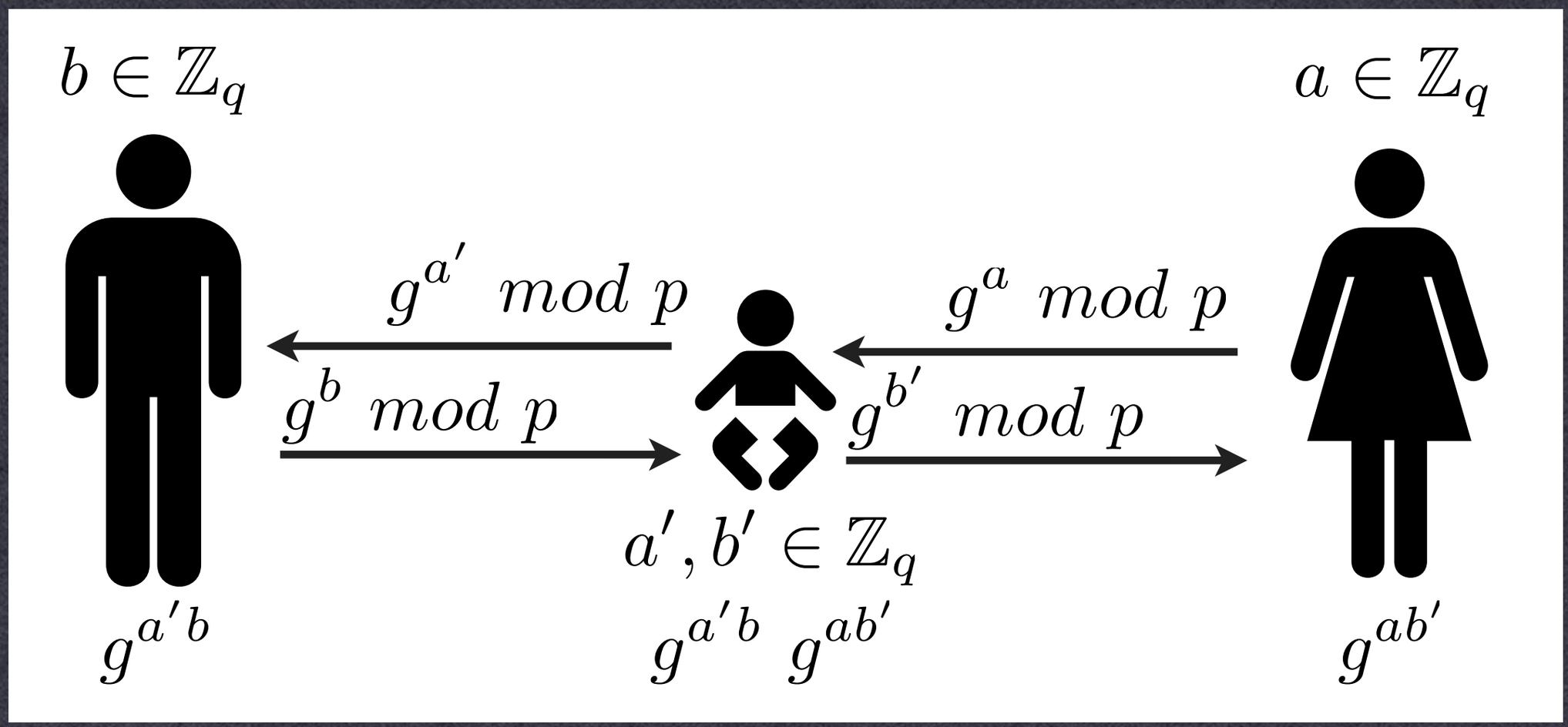
$g^b \ mod \ p$

g^{ab}

g^{ab}

Man in the Middle

- **Assume an <u>active</u> adversary:**



Man in the Middle

- Caused by lack of <u>authentication</u>
  - 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 <u>offline?</u>
  - Key agreement protocols are interac
  - 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:

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

$$d \; : \; ed \; mod \; \phi(N) = 1$$

Output:

$$pk = (e, N)$$

$$sk = d$$

Encryption

$$c = m^e \; mod \; N$$

Decryption

$$m = c^d \; mod \; N$$

“Textbook RSA”

- In practice, we don’t use Textbook RSA
  - Fully deterministic (not semantically secure)
  - <u>Malleable</u>

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

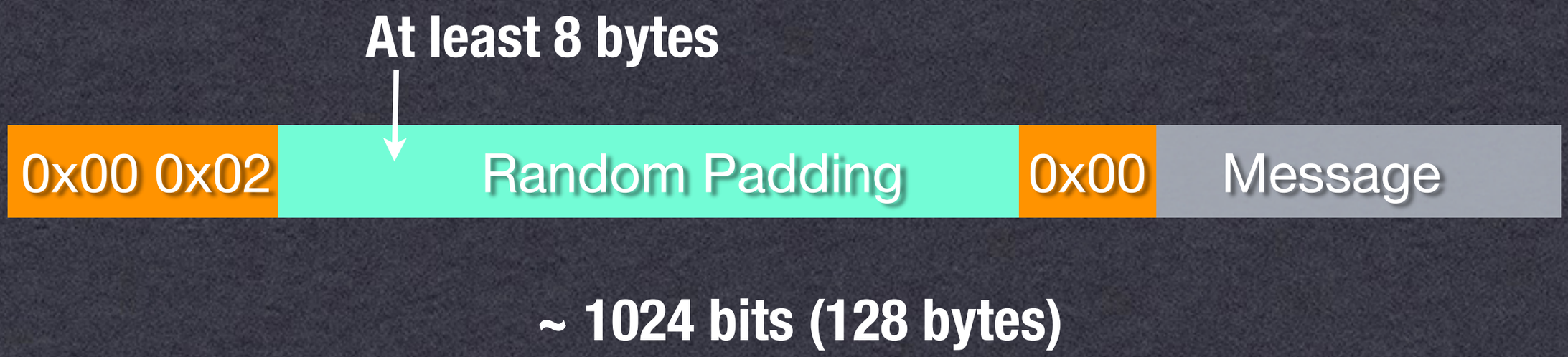
$$c'^d = (m^e \cdot x^e)^d = m \cdot x \ mod \ N$$

  - Might be <u>partially</u> invertible

-Coppersmith’s attack: recover part of plaintext (when m and e are small)

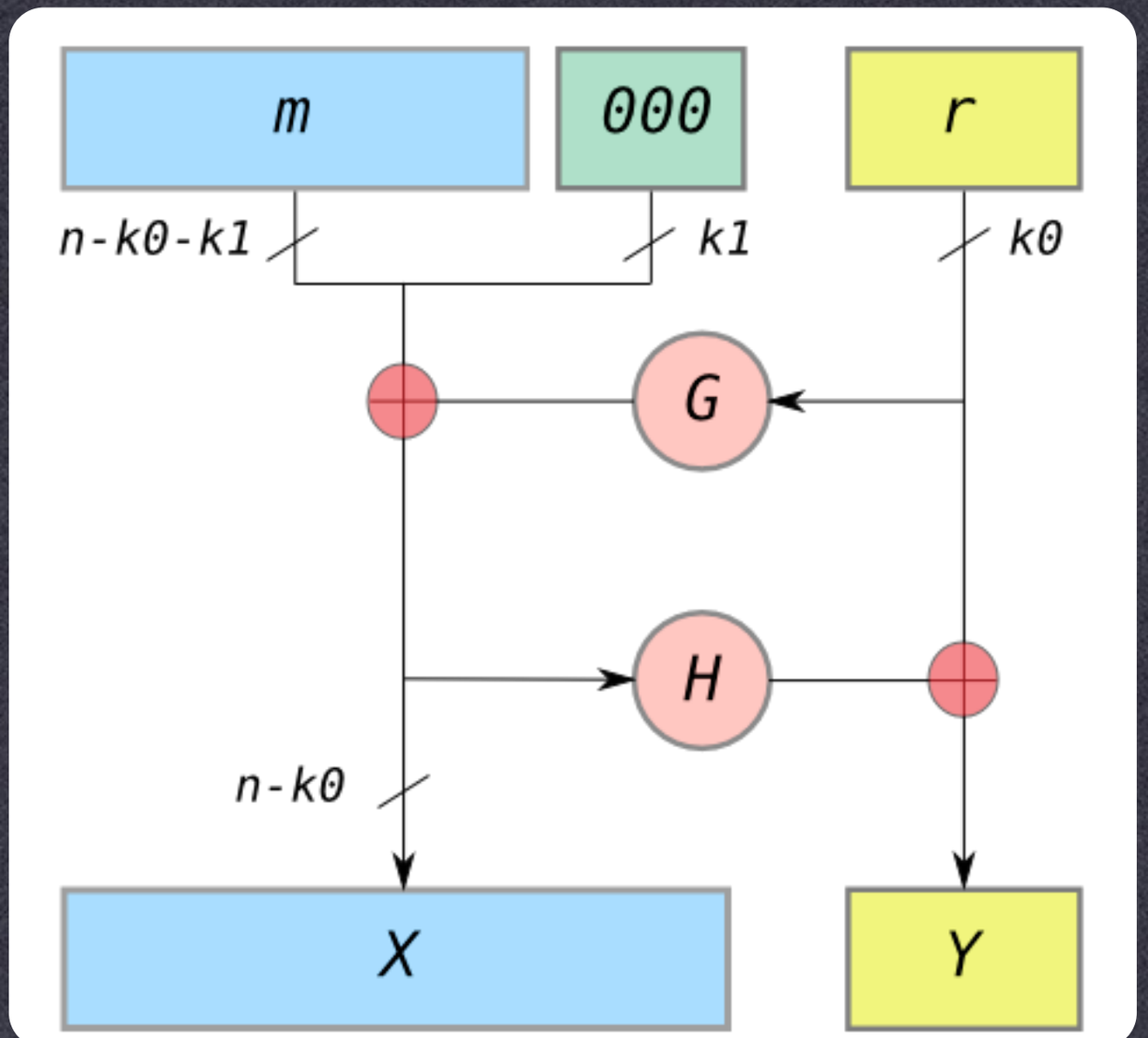
RSA Padding

- 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

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

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

$m^d \ mod \ N$

Benchmarks from: http://www.cryptopp.com/benchmarks.html
Microsoft Visual C++, Windows XP, Intel Core 2 1.83Mhz in 32-bit mode

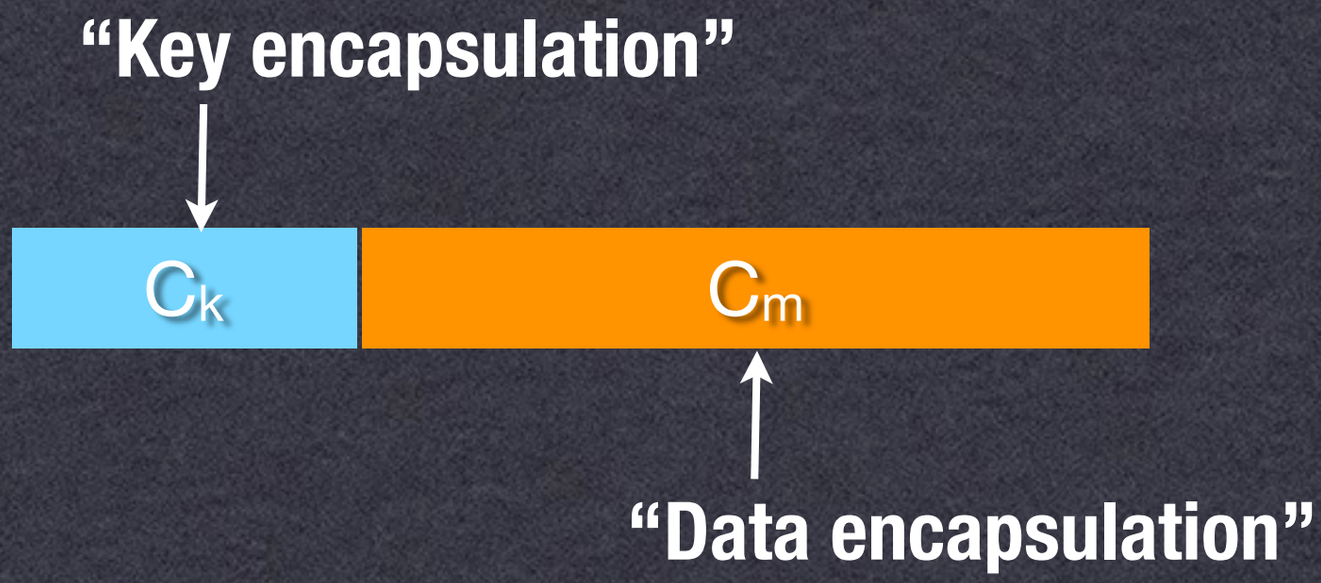
Hybrid Encryption

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

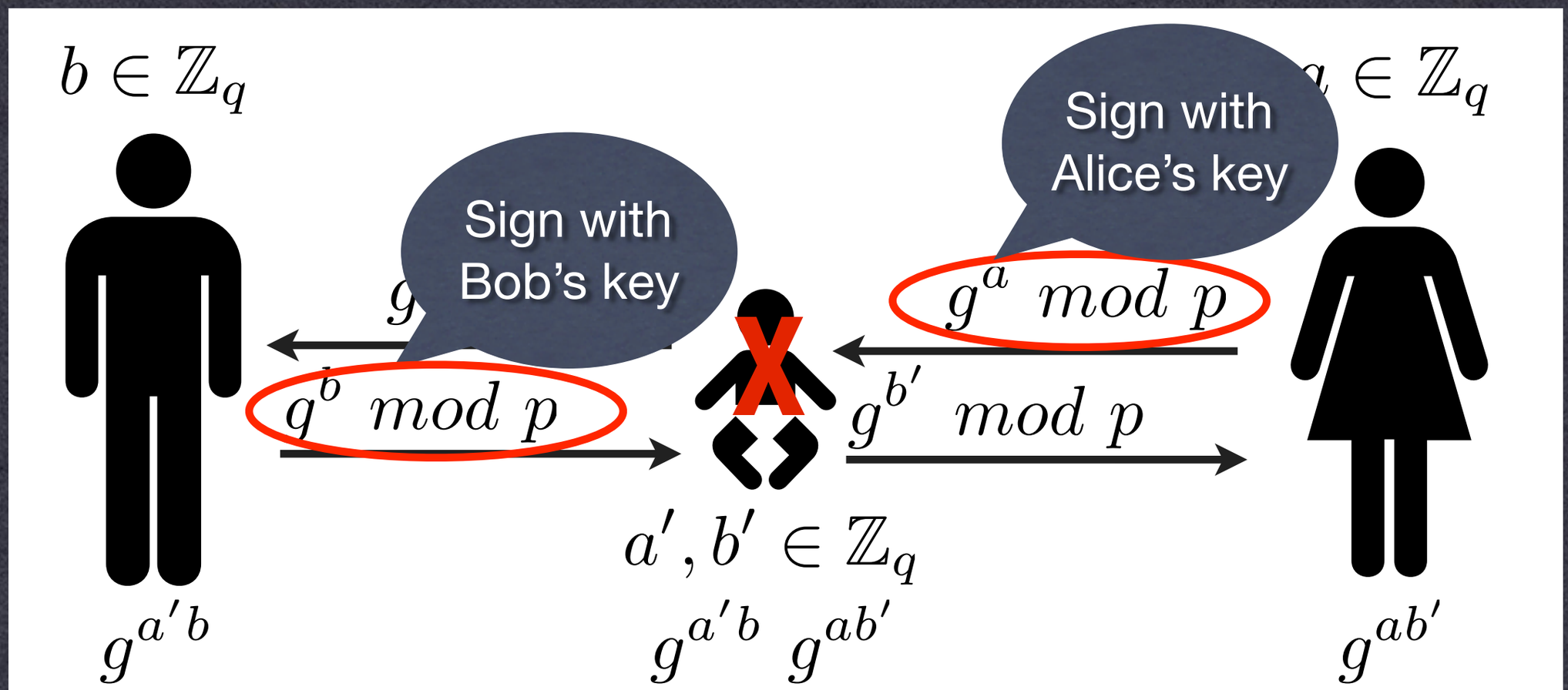
Source: www.keystrength.com (BlueKrypt). Based on ECrypt recommendations.

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 <u>active</u> 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