Practical Cryptographic Systems

Asymmetric Cryptography Continued

Some Housekeeping

- Late submissions for Assignment 1 until Friday 9/24 at 11:59pm
- Weekly HW#1 due tonight at midnight
- Weekly HW#2 coming out shortly
- Start looking for a project group (proposal due in 2 weeks 10/6)!

News

Lattice problems

Applications: factoring rational polynomials (LLL'83), closest vector (Babai'86)

In last 20 years: lattice problems became the most important hardness assumptions in post-quantum cryptography.

Have worst-case to average-case reduction (Ajtai, Regev):

Cannot solve worst-case lattice problem implies cannot break crypto.

Flexible enough to build many primitives

New era: Fully homomorphic encryption (Gentry 2009,..., BGV 2011)

Obfuscation (BISW, 2017)

Lossy trapdoor functions (PW 2008, LSSS 2017)

Test of quantumness, certifiable randomness (BCMVV 2018, Mahadev 2018)

Lattices are unique in this flexibility and theoretical guarantee.

Vital to investigate whether or not an efficient classical/quantum algorithm exists to:

1) break the worst-case assumption -> cryptosystems can still be secure, or
2) break the cryptosystems

Today: efficient quantum algorithm for lattices with certain parameter ranges.

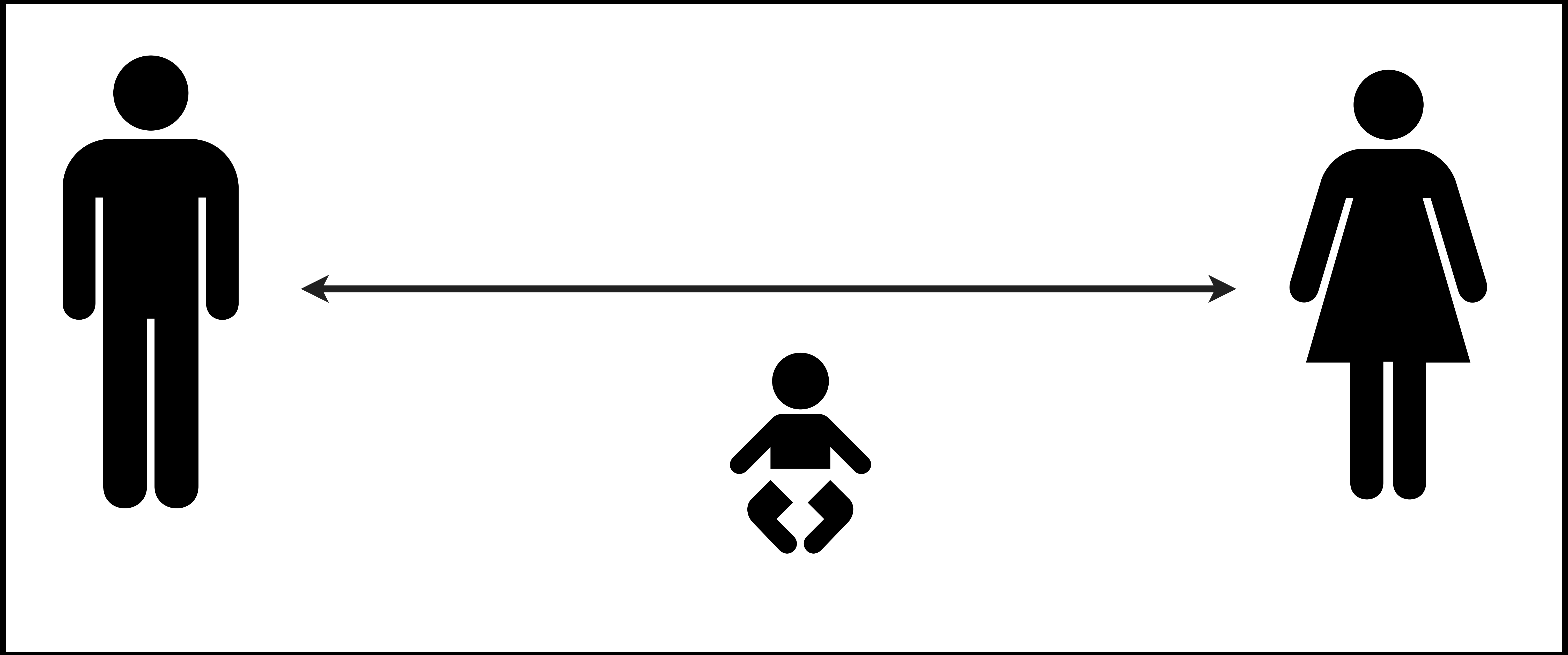
https://youtube.com/c/SimonsInstituteTOC/live

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

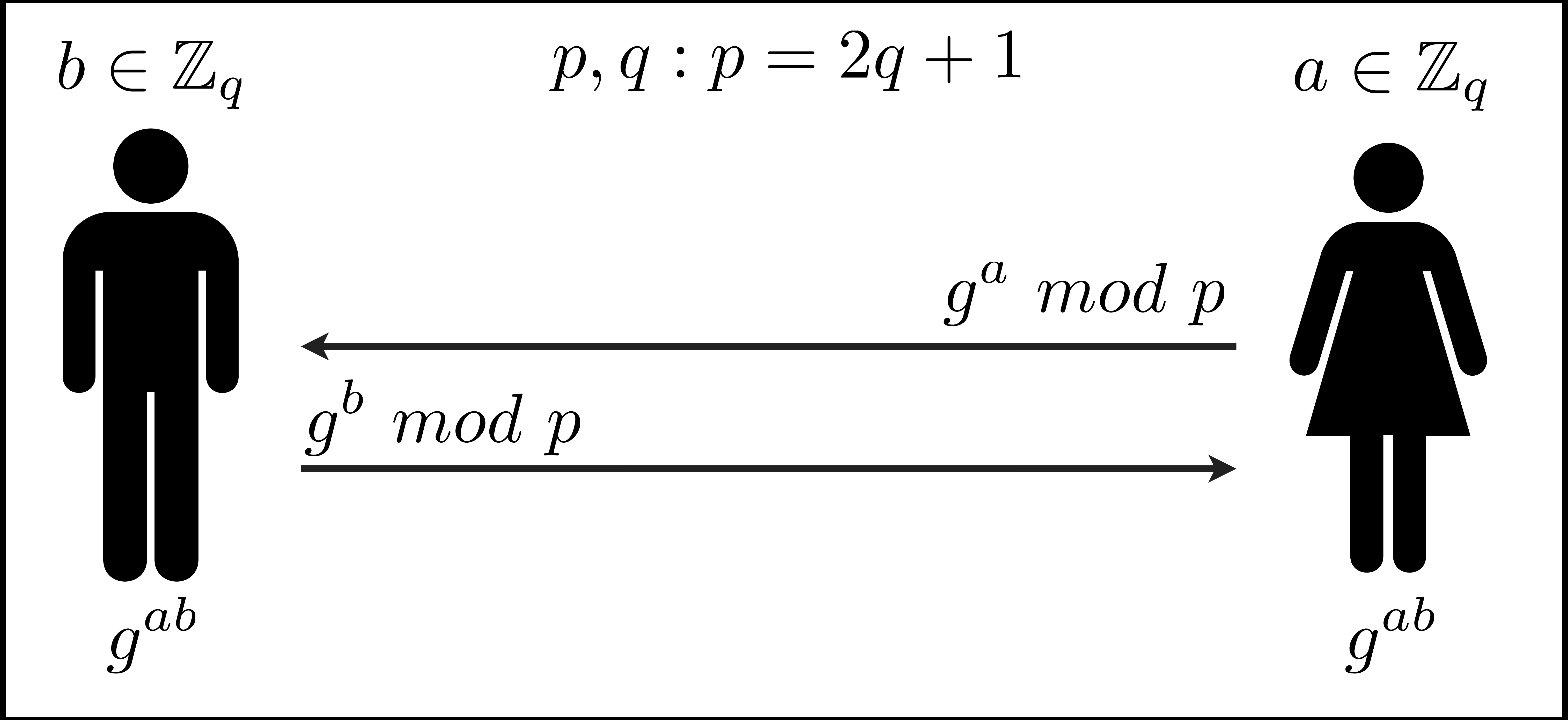
- Establish a shared key in the presence of a passive adversary



D-H Protocol

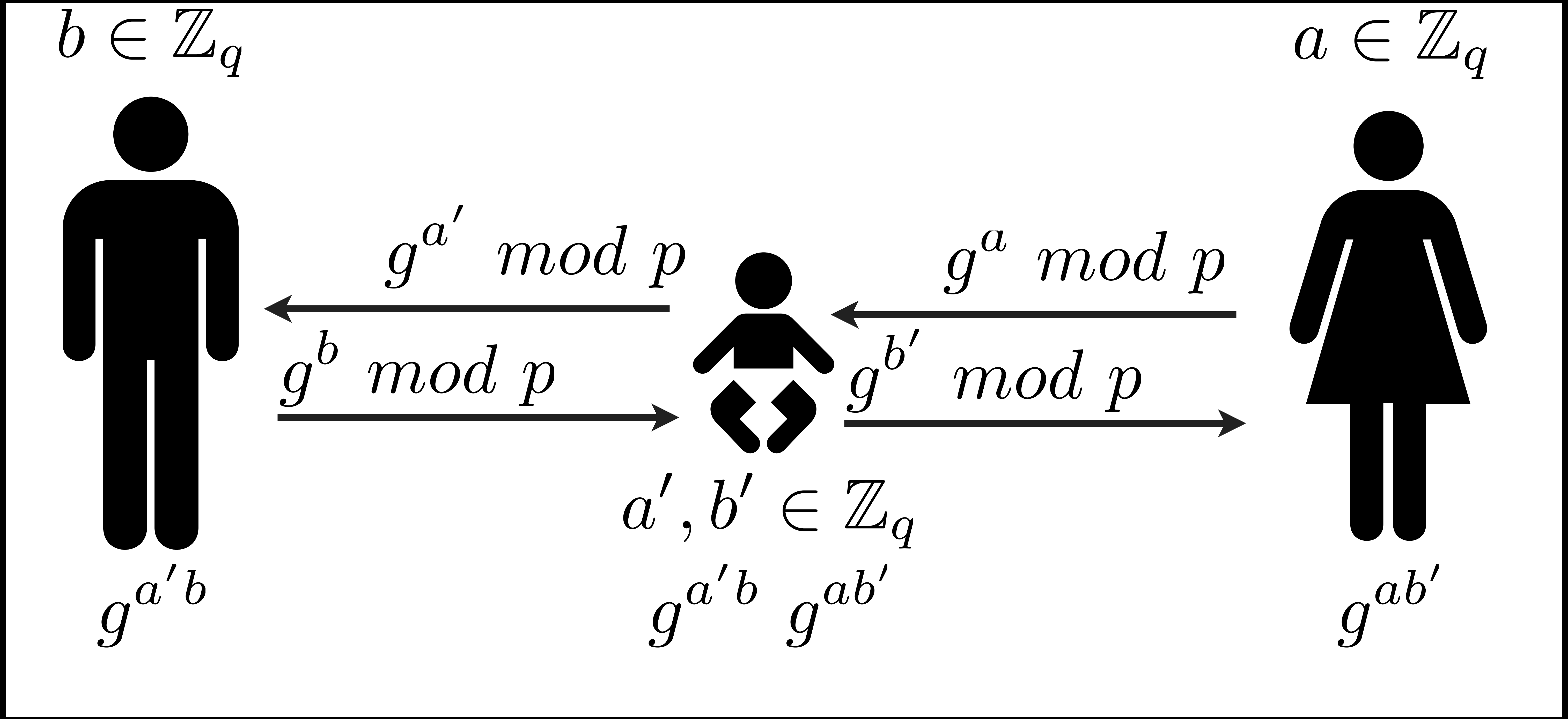
Malcolm Williamson in 72

Diffie-Hellman in 76



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

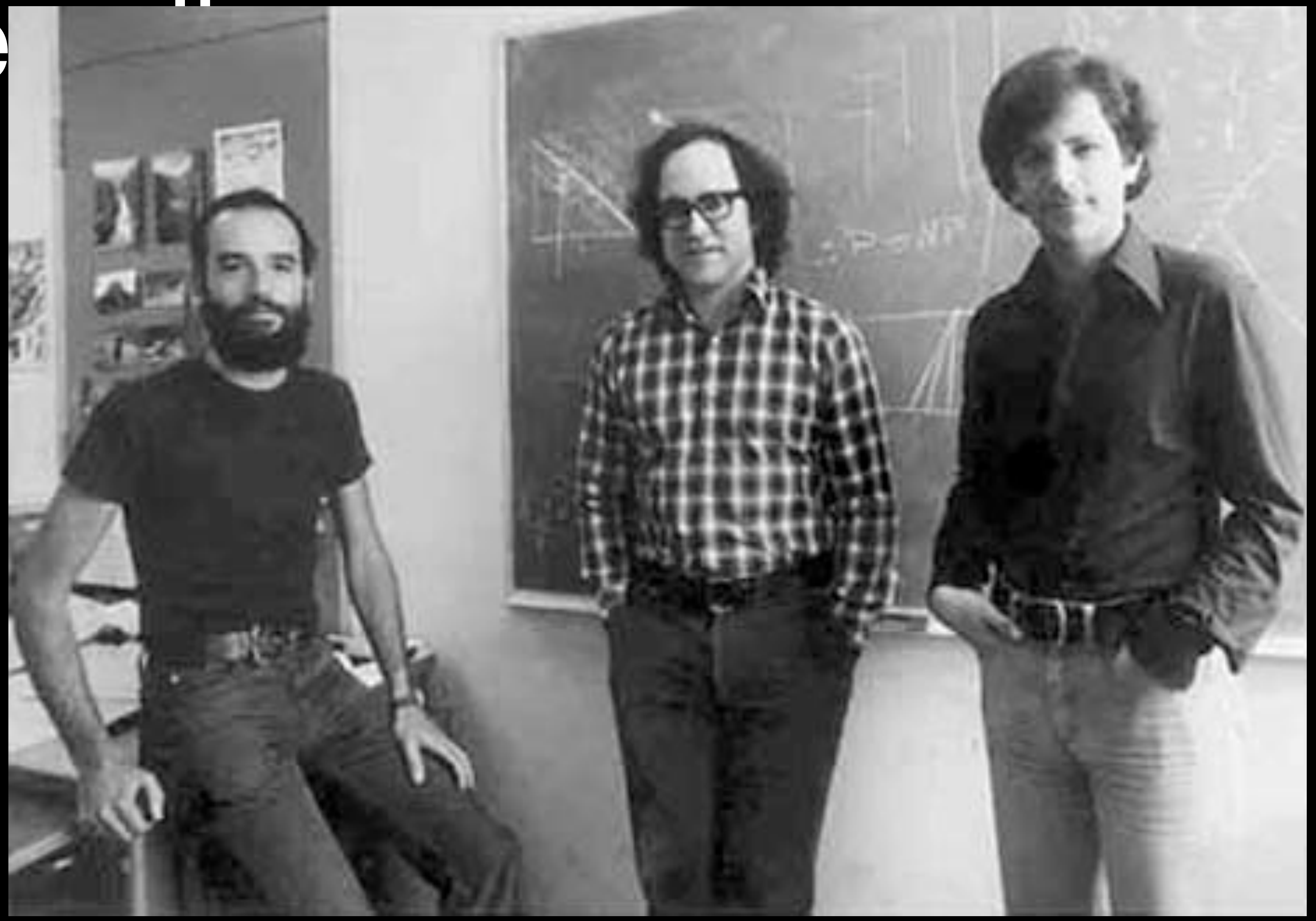
Preventing MITM

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

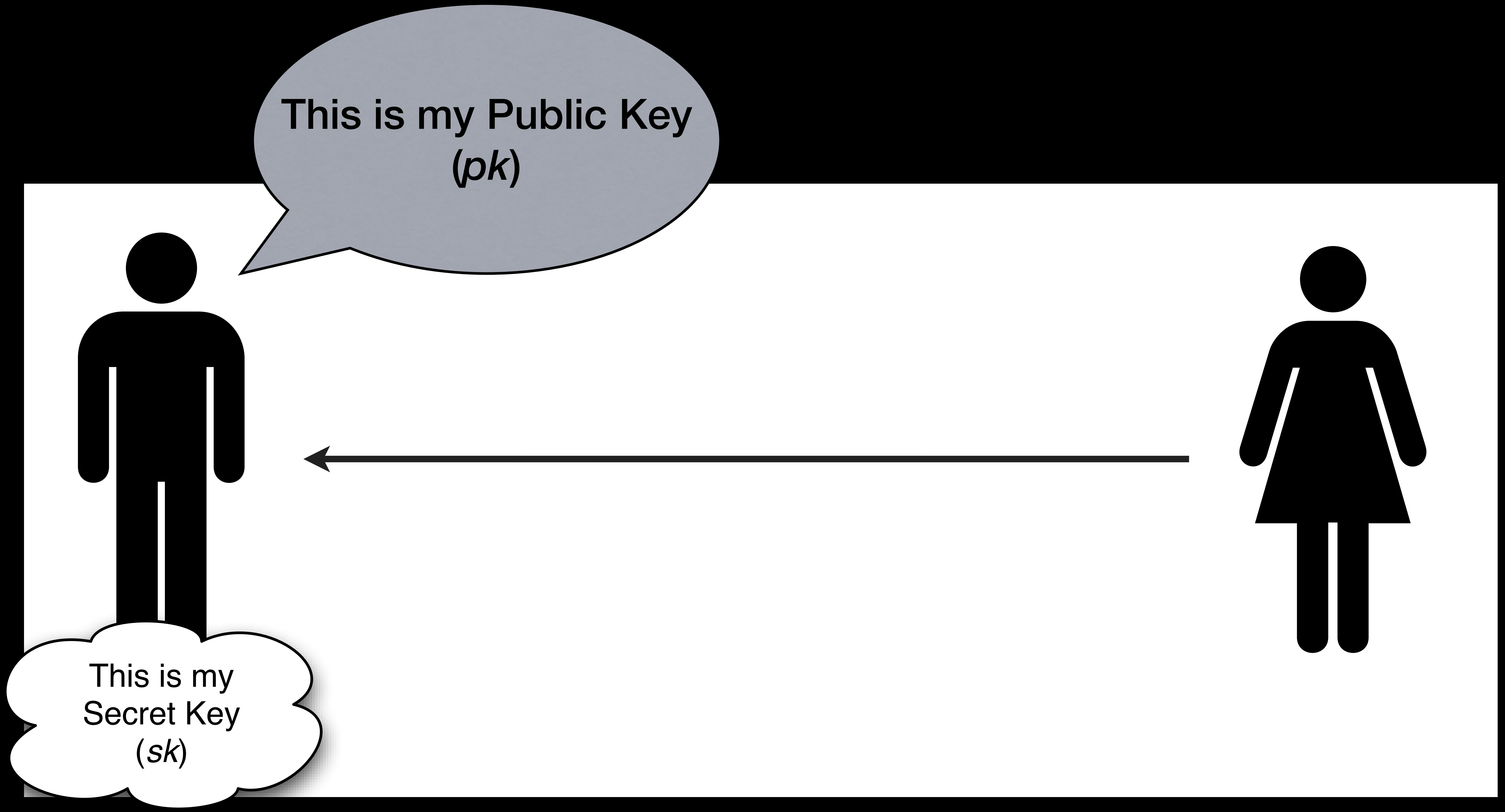
Public Key Encryption

Ellis in 72, Cocks a few months later

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



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

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

Factoring Assumption

- **Assumption:** Given $N = pq$, hard to compute p, q efficiently
- Assumed to be hard for properly chosen large factors p, q (>1024 bits)
- Best Algorithm: General Number Field Sieve or Quadratic Sieve Algorithm
- Quantum Setting?

Factoring Assumption

- **Assumption:** Given $N = pq$, hard to compute p, q efficiently
- Assumed to be hard for properly chosen large factors p, q (>1024 bits)

Not Exactly RSA

- Best Algorithm: General Number Field Sieve or Quadratic Sieve Algorithm
- Quantum Setting?

RSA Assumption

- **Assumption:** Given N, e, where N is a RSA modulus, $e > 2$ with $\gcd(e, \varphi(N)) = 1$ and a uniformly random $y \in Z_N^*$, it is hard to find $x \in Z_N^*$ such that $x^e \equiv y \ mod\ N$

RSA Assumption

- **Assumption:** Given N, e, where N is a RSA modulus, $e > 2$ with $\gcd(e, \varphi(N)) = 1$ and a uniformly random $y \in Z_N^*$, it is hard to find $x \in Z_N^*$ such that $x^e \equiv y \ mod\ N$
- Does not hold if factoring assumption does not hold. Why?

RSA Assumption

- **Assumption:** Given N, e, where N is a RSA modulus, $e > 2$ with $\gcd(e, \varphi(N)) = 1$ and a uniformly random $y \in Z_N^*$, it is hard to find $x \in Z_N^*$ such that $x^e \equiv y \ mod \ N$
- Does not hold if factoring assumption does not hold. Why?
- Does factoring assumption hold if RSA assumption does not hold?

“Textbook RSA”

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

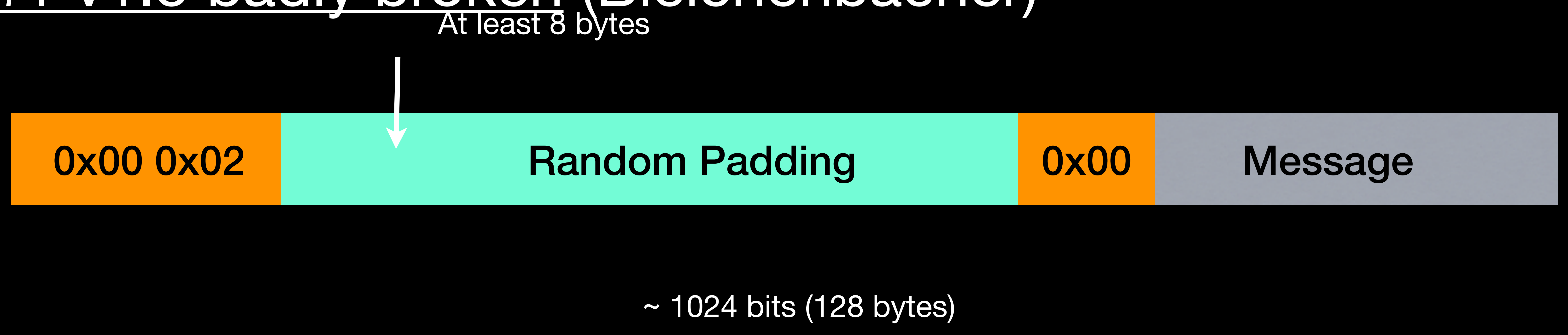
$$c' = c \cdot x^e \ mod \ N$$

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

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

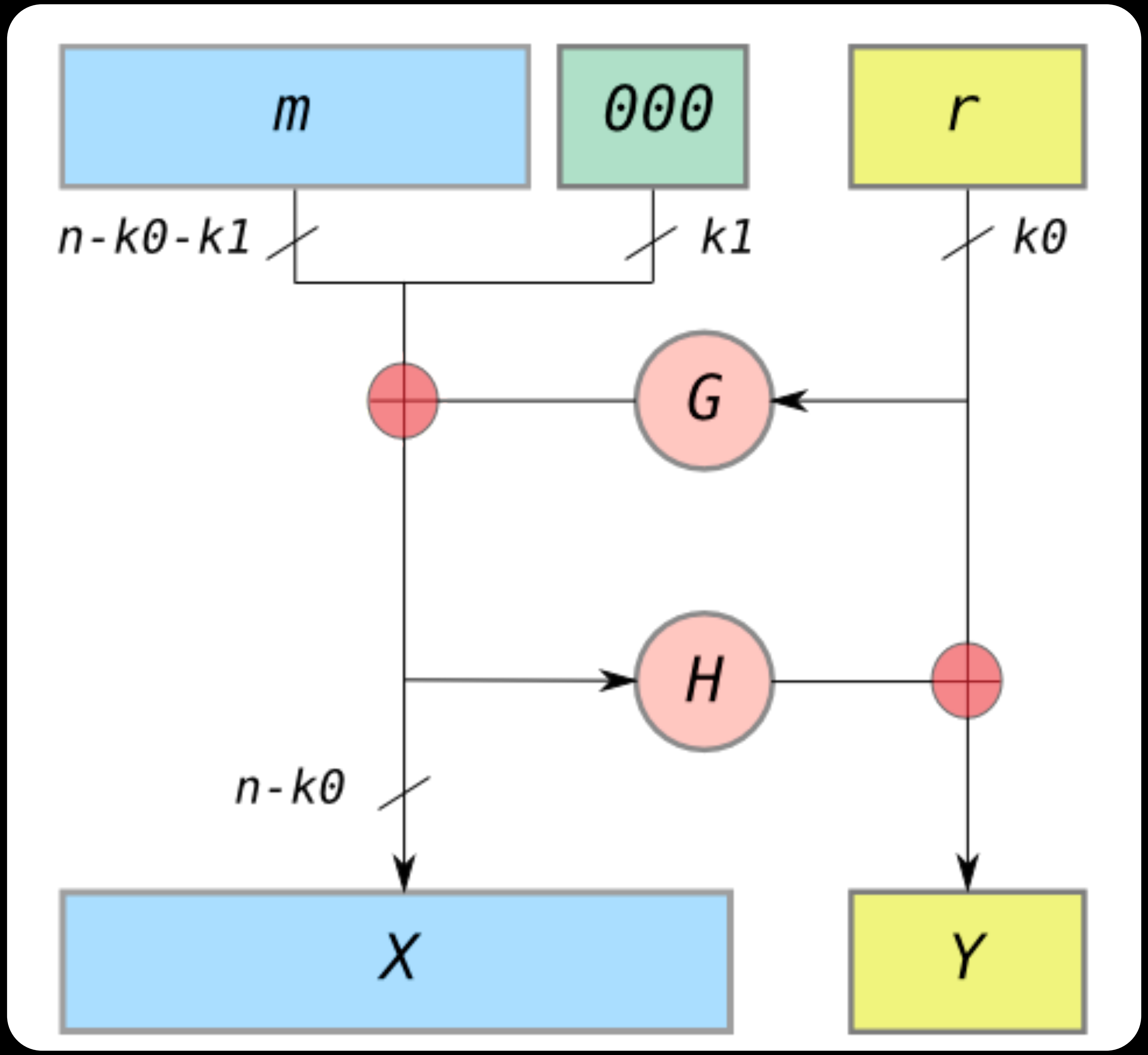
RSA Padding

- 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

		Cycles/Byte
	AES (128 bit key)	18
	DES (56 bit key)	51
$m^e \ mod \ N$ $e = 65,537$		1,016
$m^d \ mod \ N$		21,719

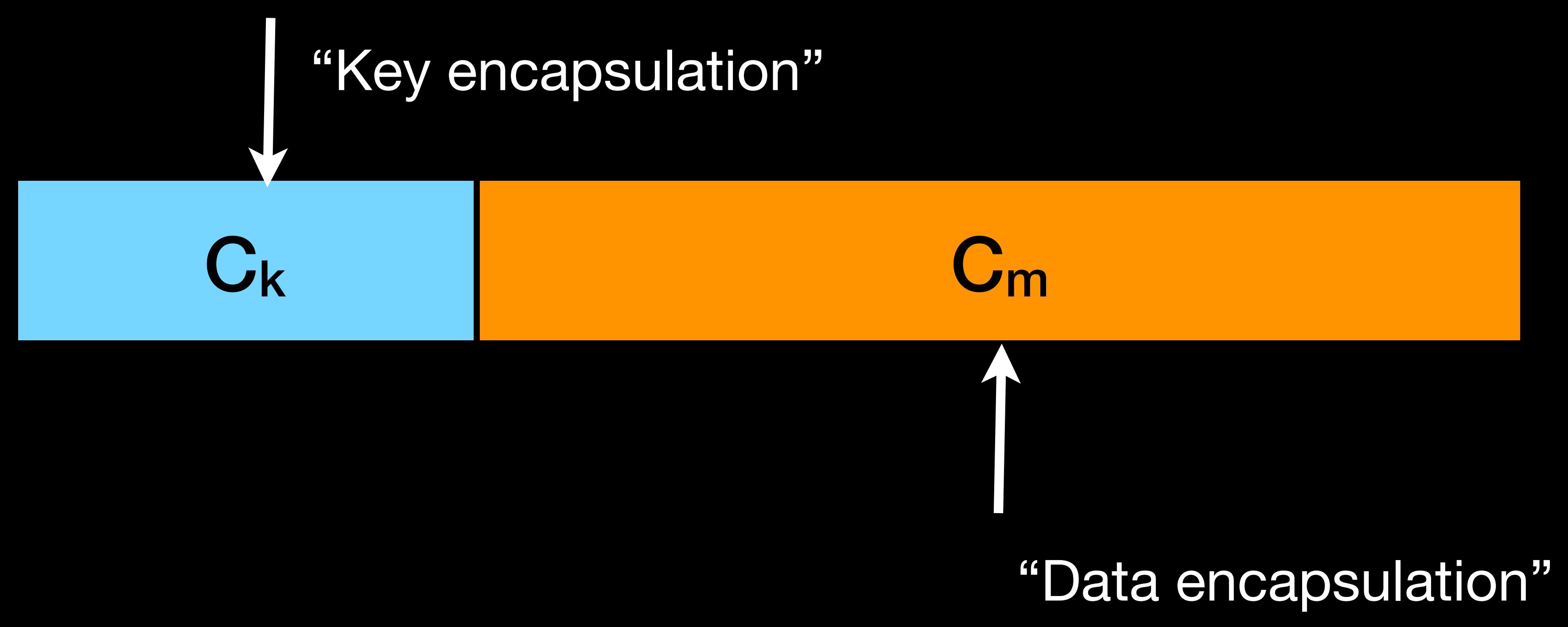
Benchmarks from: http://www.cryptopp.com/benchmarks.html
Microsoft Visual C++, Windows XP, Intel Core 2 1.83Mhz in 32-bit mode

Hybrid Encryption

- Mixed Approach
  - Use PK encryption to encrypt a symmetric key

$$k \xleftarrow{\$} \{0,1\}^k$$
$$C_k \leftarrow RSA.Encrypt_{pk}(k)$$
$$C_m \leftarrow AES.Encrypt_k(message)$$

  - Use (fast) symmetric encryption on data



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

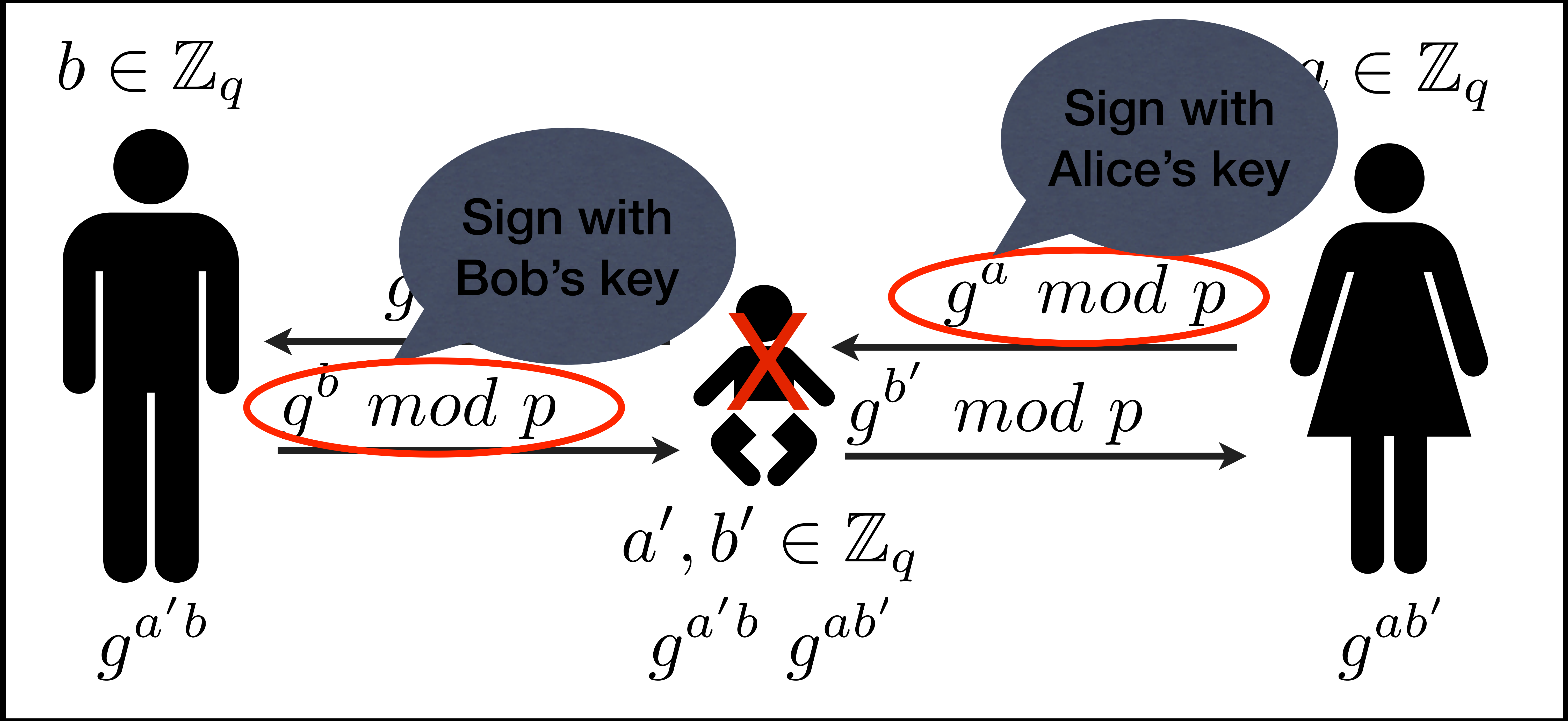
Source: www.keystrength.com (BlueKrypt). Based on ECrypt recommendations.

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

- Elliptic Curve Cryptography!