Practical Cryptographic Systems

Asymmetric Cryptography II

Instructor: Matthew Green

Housekeeping

- A1 due this week
  - **Gradescope**
  - **Note: there are separate Gradescope sections for 4xx, 6xx!**
- Written assignment
  - Gradescope

News?

Review

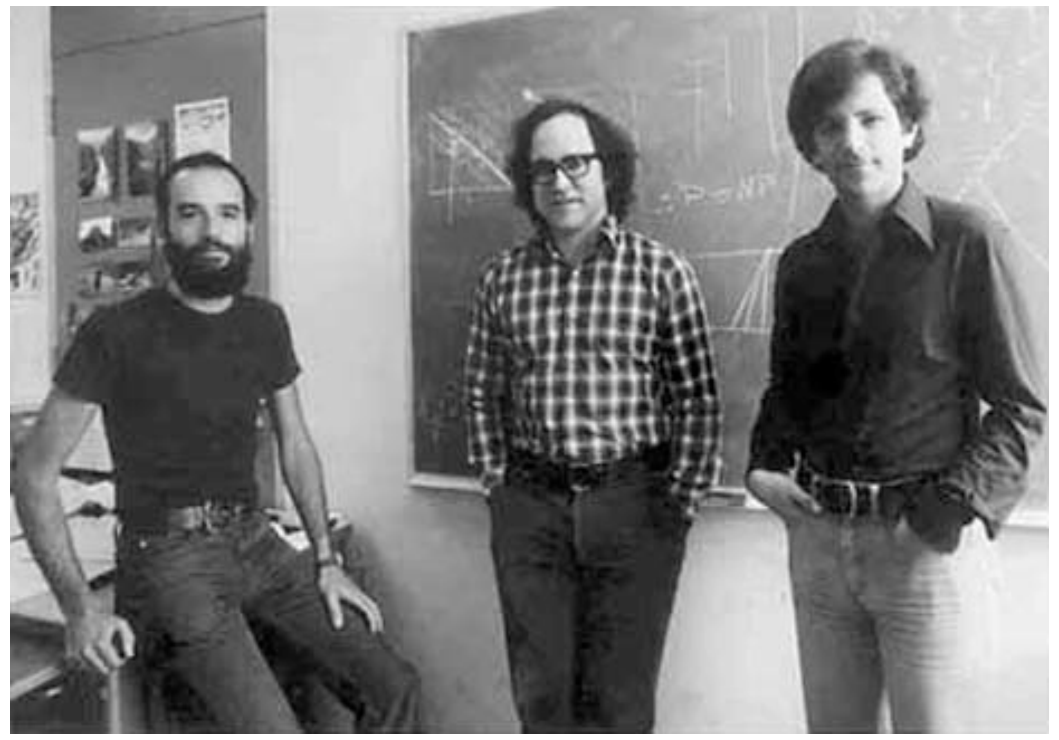
- Key distribution
- How do we do it with symmetric crypto?

Review

- Key distribution
- How do we do it with <u>only</u> symmetric crypto?
  - One answer: Kerberos
  - Everyone has shared keys with one trusted party (“introduction point”)
  - Trusted party creates new “session keys” between parties it knows
  - This is still pretty inconvenient

Asymmetric Crypto

- Also known as “public key” crypto
  - Gives us a way to encrypt material without *pre-existing* shared secrets

Two slides of number theory

- Arithmetic modulo primes (Zp)

  1. What is Zp?
  2. Addition (+) and multiplication (*) in Zp
  3. Multiplicative inverses
  4. Z*p is the set of invertible elements (excludes 0)

See Boneh-Shoup, Appendix A

What is a cyclic group?

- Definition:

  1. A finite set of elements
  2. An associative “group operation” (a * b = c)
  3. All elements have <u>inverses</u> (w.r.t. the group operation)
  4. There exists at least one <u>generator</u> (g) s.t.:

     (g^1, g^2, g^3, …) produces every element of the group

See Boneh-Shoup, Appendix A

Constructing a cyclic group

- Let p be a prime. Is Z^*p a cyclic group?

See Boneh-Shoup, Appendix A

Constructing a cyclic group

- Let p be a prime. Is Z*p a cyclic group?

✅ 1. A finite set of elements
✅ 2. An associative “group operation” (a * b = c)
✅ 3. All elements have <u>inverses</u> (w.r.t. the group operation)
✅ 4. There exists at least one <u>generator</u> (g) s.t.:

$(g^0, g^1, g^2, g^3, \ldots)$ produces every element of the group.

See Boneh-Shoup, Appendix A

Other notes on Z*p

- How many elements are in Z*p?
  - Note: every element a in *Zp* s.t. gcd(*a, p*) = 1 has an inverse, is in Z*p
  - This is also denoted by Euler's totient function, $\phi(\cdot)$
  - For all primes p: $\phi(p) = p - 1$
  - We also refer to this as the order of the group
    (sometimes we also refer to the order of the generator *g* as *order(g).)*
- Not every element of the group is a generator

See Boneh-Shoup, Appendix A

Other notes on Z*p (cont'd)

- Cyclic groups can have "subgroups"
  - These are subsets of the main group that are also groups
  - I.e., each subgroup has a generator that generates the subgroup
  - Useful fact:

    For every prime divisor of *p-1*, there exists one subgroup of that size.

    E.g., consider p=11. Here p-1 = 10. Divisors are (1, 2, 5, 10).

See Boneh-Shoup, Appendix A

Some convenient mathematical properties

$$a, b \in \{0, 1, \ldots, p-1\} \qquad \langle g \rangle = \mathbb{Z}_p^*$$

$$g^{order(g)} = 1$$

$$g^a \cdot g^b = g^{a+b \ mod \ order(g)}$$

$$(g^a)^b = (g^b)^a = g^{a*b \ mod \ order(g)}$$

See Boneh-Shoup, Appendix A

Discrete Logarithm problem

- **<u>Discrete logarithm problem</u>**

Given: $x \in_R 0, \ldots, p-2$

$$\langle g \rangle = \mathbb{G} \qquad order(g) = p-1$$

$$h = g^x$$

Find: x

This problem is <u>hard</u> if for all p.p.t. adversaries, all attackers find x with “small” probability

See Boneh-Shoup, Appendix A

Discrete Logarithm problem

This means that “reversing” exponentiation is assumed to have super-polynomial running time.

How about the exponentiation itself?

- **<u>Discrete logarithm problem</u>**

Given: $x \in_R 0, \ldots, p-2$

$$\langle g \rangle = \mathbb{G} \qquad order(g) = p - 1$$

$$h = g^x$$

Find: x

This problem is hard if for all p.p.t. adversaries, all attackers find x with “small” probability

See Boneh-Shoup, Appendix A

Discrete Logarithm problem

This means that "reversing" exponentiation is assumed to have super-polynomial running time.

How about the exponentiation itself?

- **Discrete logarithm problem**

Given: $x \in_R 0, \ldots, p-2$

$$\langle g \rangle = \mathbb{G} \qquad order(g) = p$$

$$h = g^x$$

Note that for this to hold, the size of *p* must be pretty large!

In practice, we typically assume *p* is at least 1024 bits. And 3072 bits is the minimum in modern protocols!

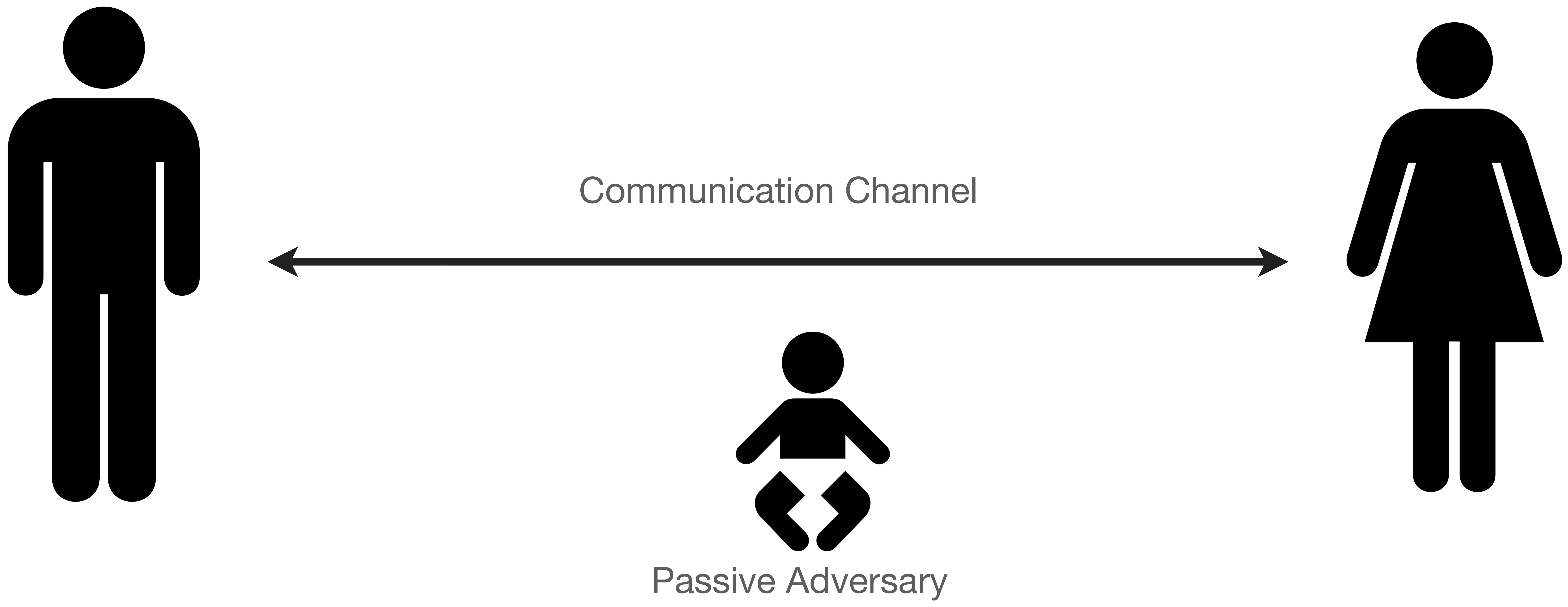
Find: x

This problem is hard if for all p.p.t. adversaries, all attackers find x with "small" probability

See Boneh-Shoup, Appendix A

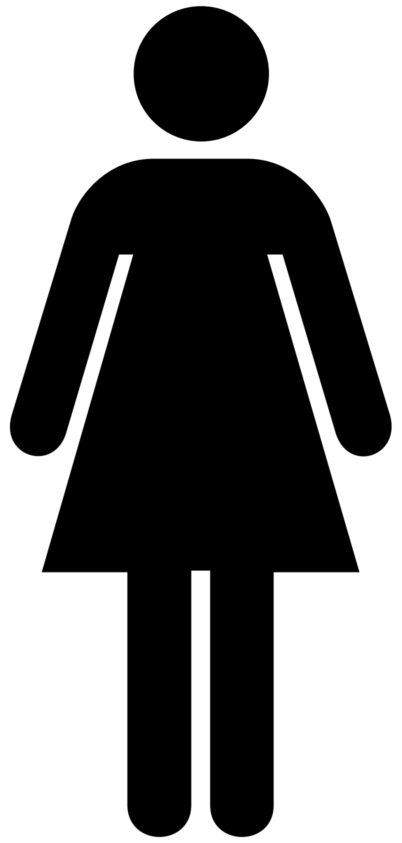
Key Agreement

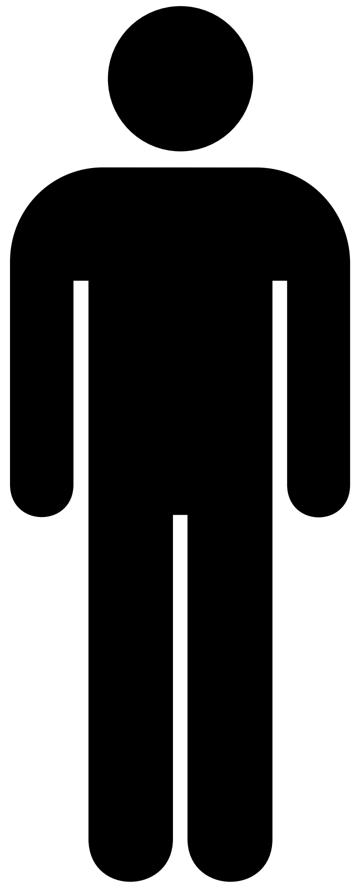
- Establish a shared key in the presence of a passive adversary



D-H Protocol

$$p, \langle g \rangle = \mathbb{Z}_p^*$$
$$a \in \mathbb{Z}_{\phi(p)}$$

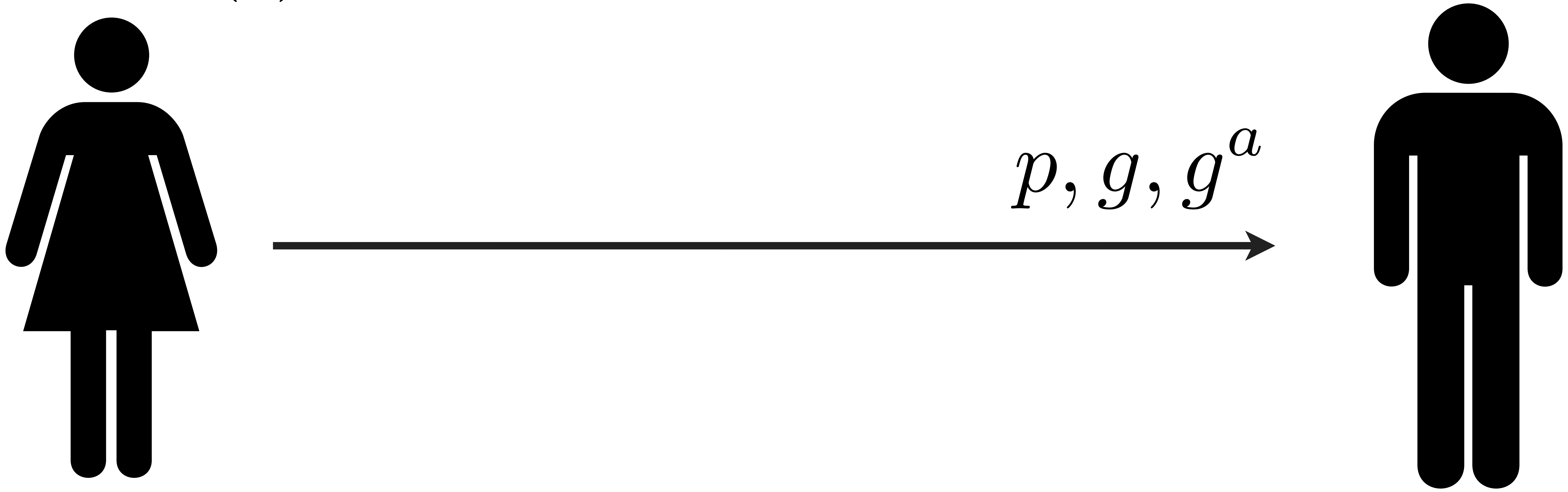




D-H Protocol

$$p, \langle g \rangle = \mathbb{Z}_p^*$$

$$a \in \mathbb{Z}_{\phi(p)}$$

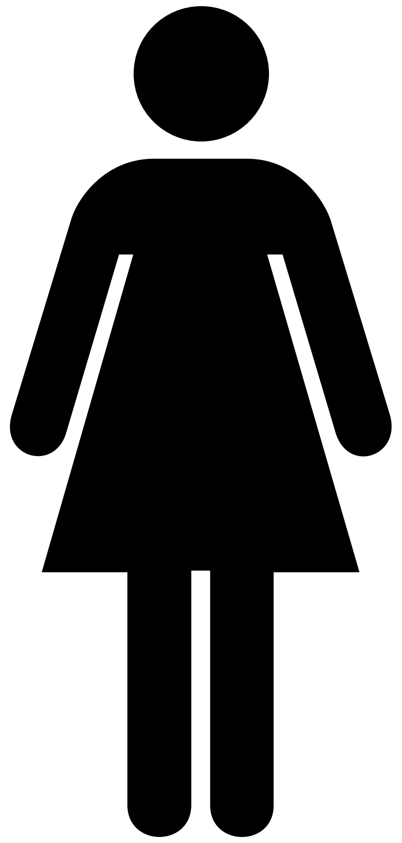


$$p, g, g^a$$

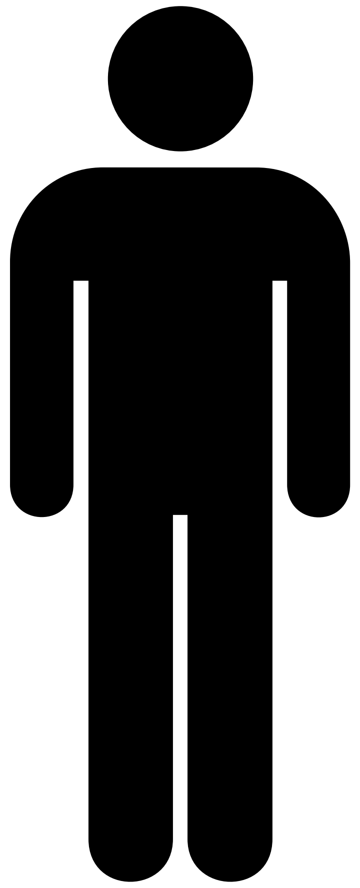
D-H Protocol

$p, \langle g \rangle = \mathbb{Z}_p^*$

$a \in \mathbb{Z}_{\phi(p)}$



$b \in \mathbb{Z}_{\phi(p)}$



$$\xrightarrow{p, g, g^a}$$

$$\xleftarrow{g^b}$$

D-H Protocol

$p, \langle g \rangle = \mathbb{Z}_p^*$

$a \in \mathbb{Z}_{\phi(p)}$

$b \in \mathbb{Z}_{\phi(p)}$

p, g, g^a

g^b

g^{ba}

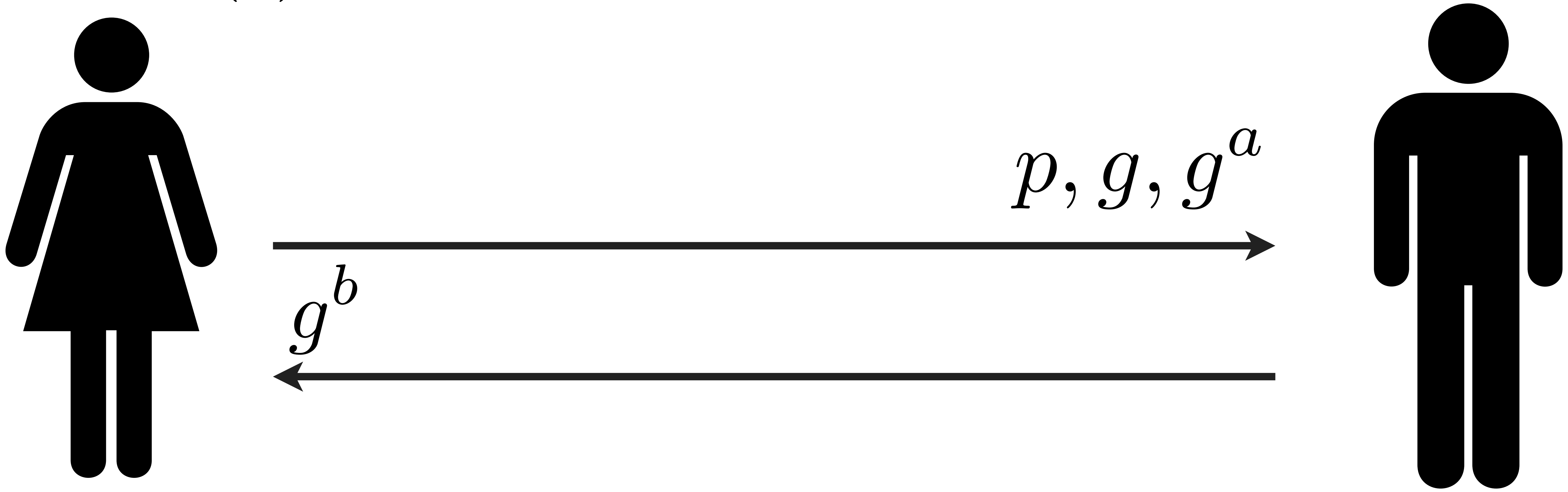
g^{ab}

D-H Protocol

$$p, \langle g \rangle = \mathbb{Z}_p^*$$

$$a \in \mathbb{Z}_{\phi(p)}$$

$$b \in \mathbb{Z}_{\phi(p)}$$



$$p, g, g^a$$

$$g^b$$

$$g^{ba}$$

$$g^{ab}$$

Usually we “hash” the shared secret value to make a secret encryption key, and then encrypt using a fast symmetric encryption scheme!

Hard problems (2)

- **Diffie-Hellman problem**

Given: $a, b \in_R 0, \ldots, p-2$

$$\langle g \rangle = \mathbb{G} \qquad order(g) = p-1$$

$$(g, g^a, g^b)$$

Find: g^{ab}

This problem is hard if for all p.p.t. adversaries, all attackers output a solution with “small” probability

See Boneh-Shoup, Appendix A

Hard problems (2)

- **Diffie-Hellman problem**

Notice this is just the Diffie-Hellman scheme re-written as a mathematical assumption!

Given: $a, b \in_R 0, \ldots, p-2$

$$\langle g \rangle = \mathbb{G} \qquad order(g) = p-1$$

$$(g, g^a, g^b)$$

Find: g^{ab}

This problem is hard if for all p.p.t. adversaries, all attackers output a solution with “small” probability.

See Boneh-Shoup, Appendix A

Hard problems (2)

- <u>**Diffie-Hellman problem**</u>

Given: $a, b \in_R 0, \ldots, p-2$

$$\langle g \rangle = \mathbb{G} \qquad order(g) = p - 1$$

$$(g, g^a, g^b)$$

Find: g^{ab}

Notice this is just the Diffie-Hellman scheme re-written as a mathematical assumption!

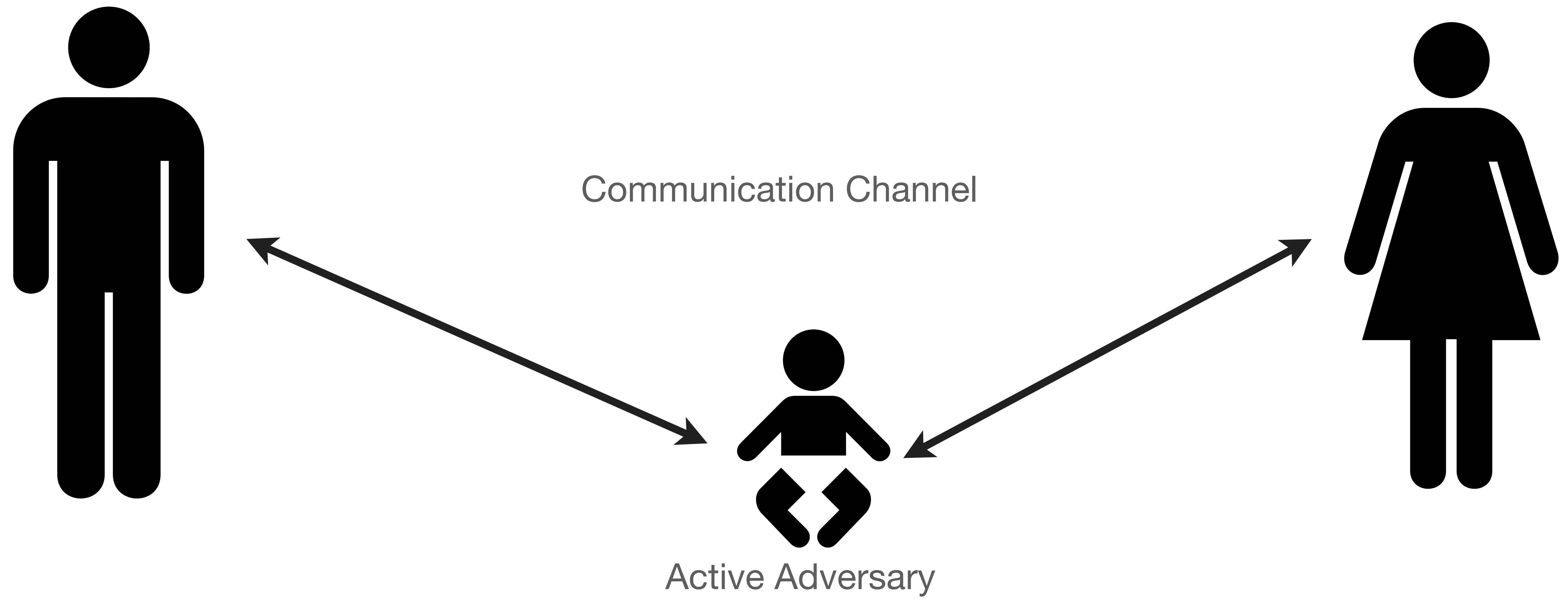
Note that for this to hold, the size of *p* must be pretty large!

In practice, we typically assume *p* is at least 1024 bits. And 3072 bits is the minimum in modern protocols!

This problem is <u>hard</u> if for all p.p.t. adversaries, all attackers output a solution with “small” probability.

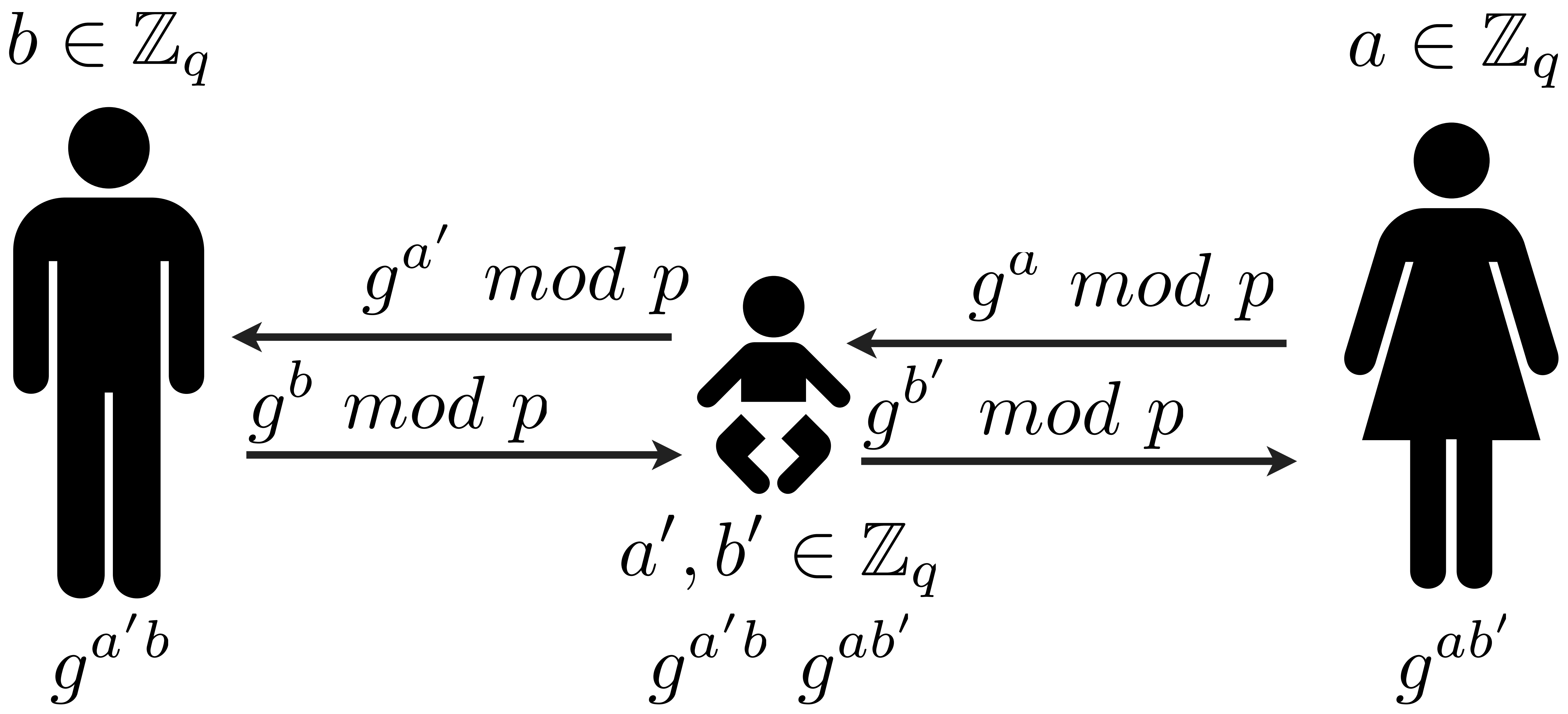
See Boneh-Shoup, Appendix A

What if we have an active adversary?



Man in the Middle

- Assume an active adversary:



Man in the Middle

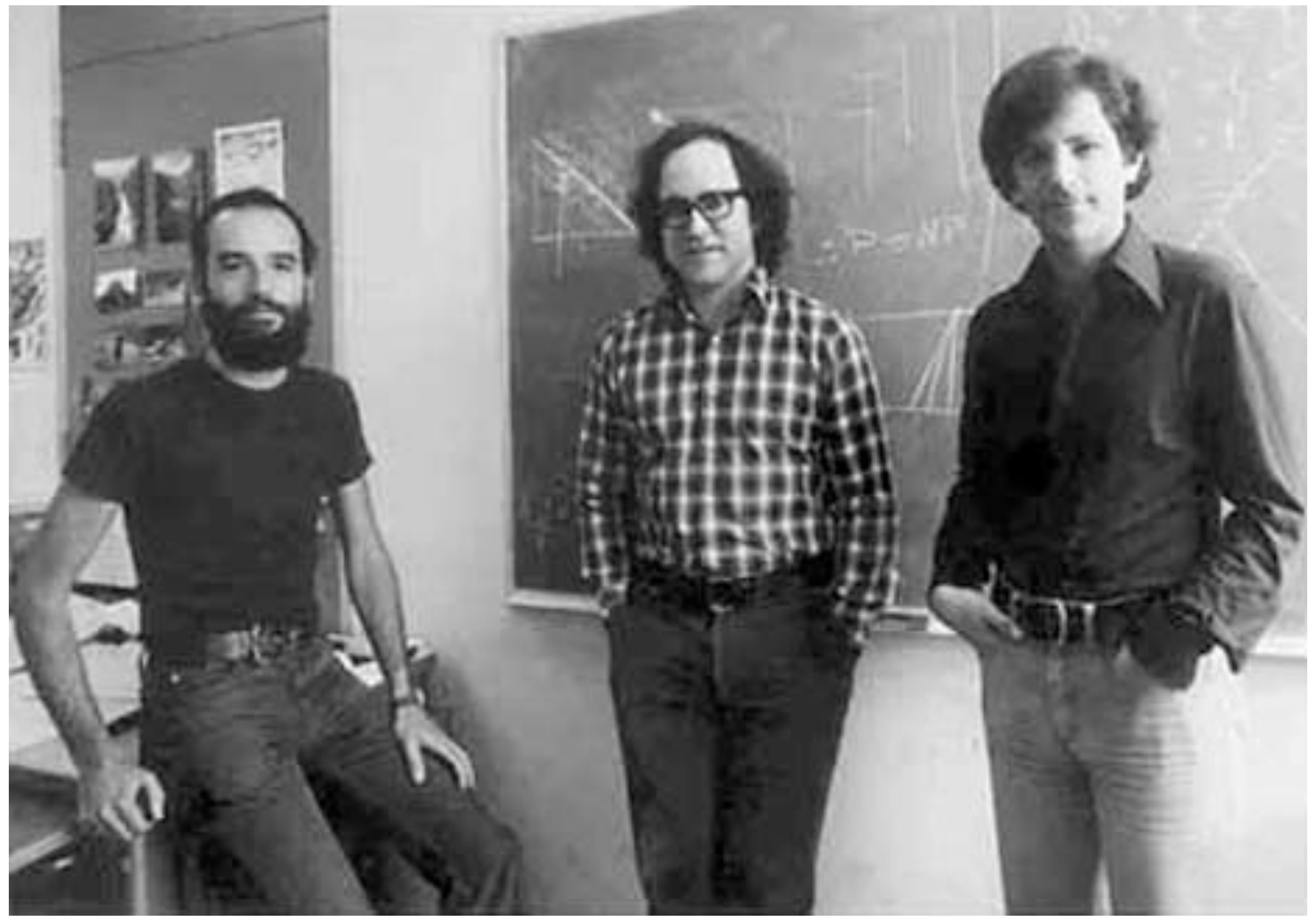
- Caused by lack of <u>authentication</u>
  - D-H lets us establish a shared key with anyone...
    but that's the problem…
  - We don't know if the person we're talking to is the right person
- Solution?

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 interactive
  - e.g., want to send an email



Public Key Encryption