Leakage-Resilient Key-Dependent Message Secure Encryption Schemes

Mahesh Sreekumar Rajasree
Post-Doctoral Researcher
CISPA Helmholtz

Joint work with Dhairya Gupta (IITD) and Harihar Swaminathan (IITD)

Contents

Contents

- Introduction

Contents

- Introduction
- Standard Security

Contents

- Introduction
- Standard Security
- Leakage-Resilience Security

Contents

- Introduction
- Standard Security
- Leakage-Resilience Security
- Key-Dependent Message Security

Contents

- Introduction
- Standard Security
- Leakage-Resilience Security
- Key-Dependent Message Security
- LR-KDM Security

Contents

- Introduction
- Standard Security
- Leakage-Resilience Security
- Key-Dependent Message Security
- LR-KDM Security
- Separation, Constructions and Amplifications

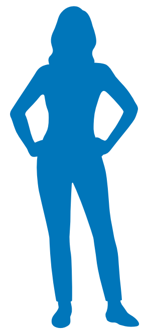
Contents

- Introduction
- Standard Security
- Leakage-Resilience Security
- Key-Dependent Message Security
- LR-KDM Security
- Separation, Constructions and Amplifications
- Conclusion

Introduction

Encryption Scheme

Encryption Scheme



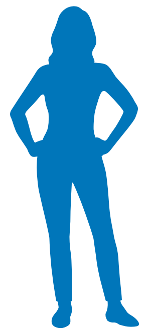
ALICE

Encryption Scheme

ALICE

BOB

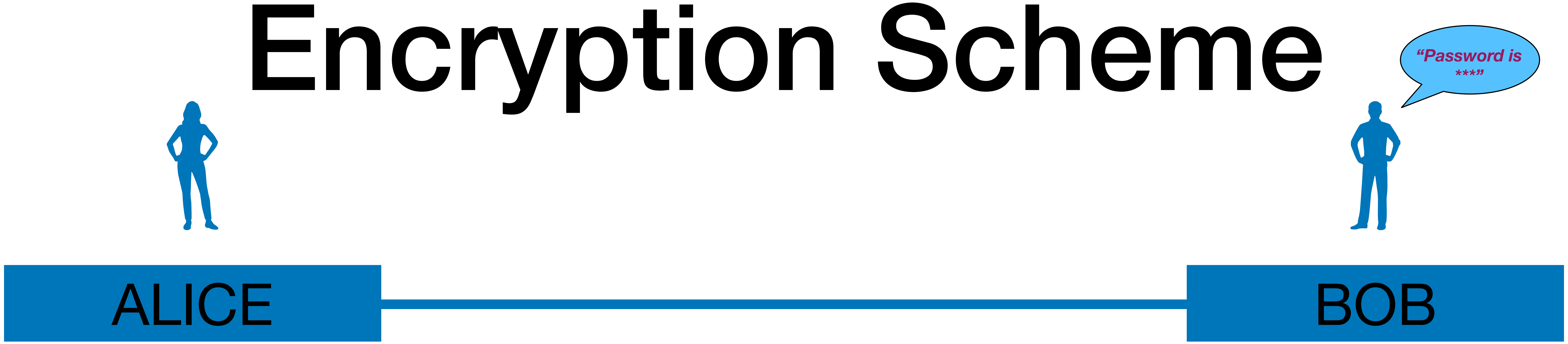
Encryption Scheme



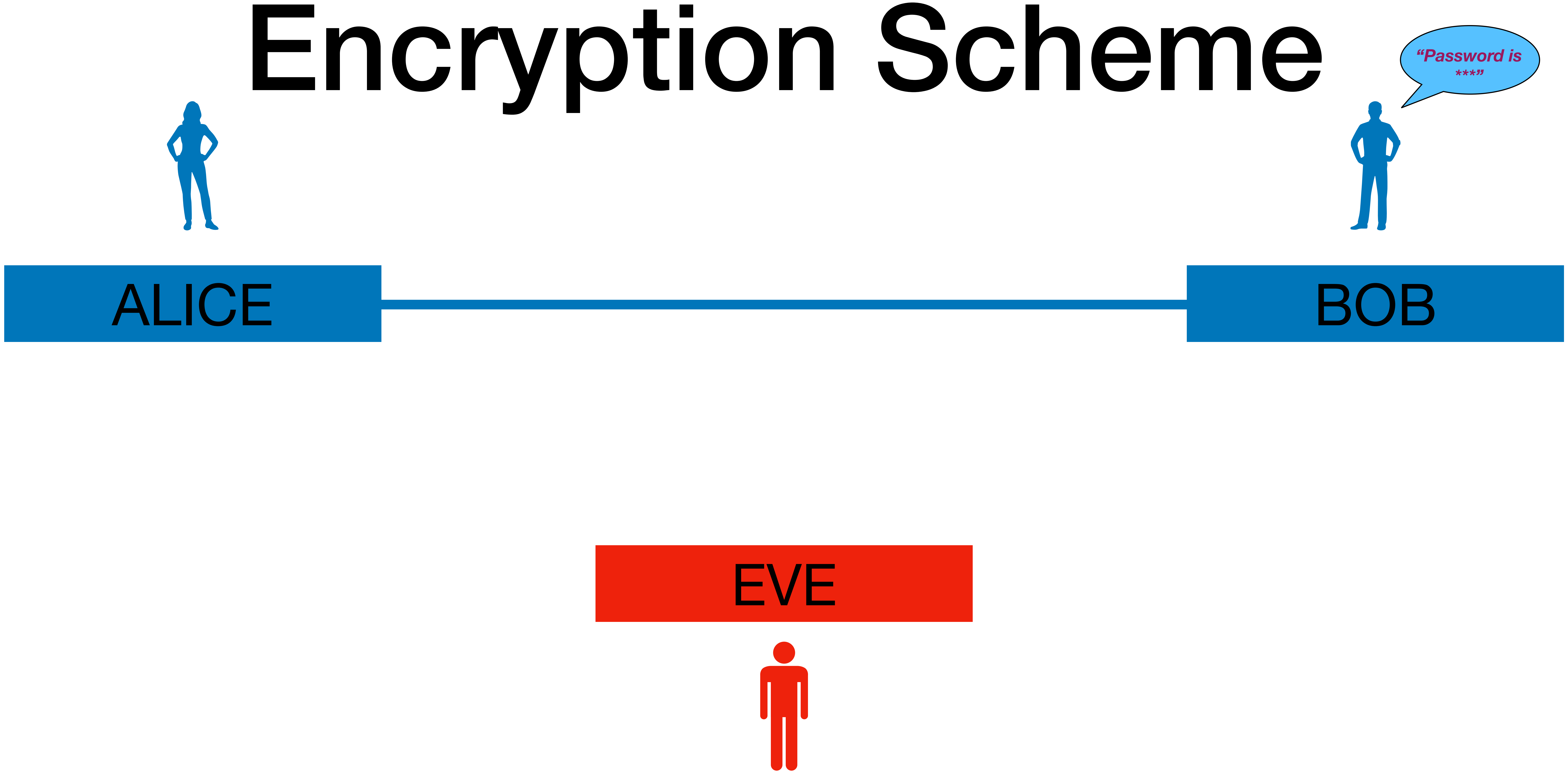
ALICE

BOB

Encryption Scheme



Encryption Scheme

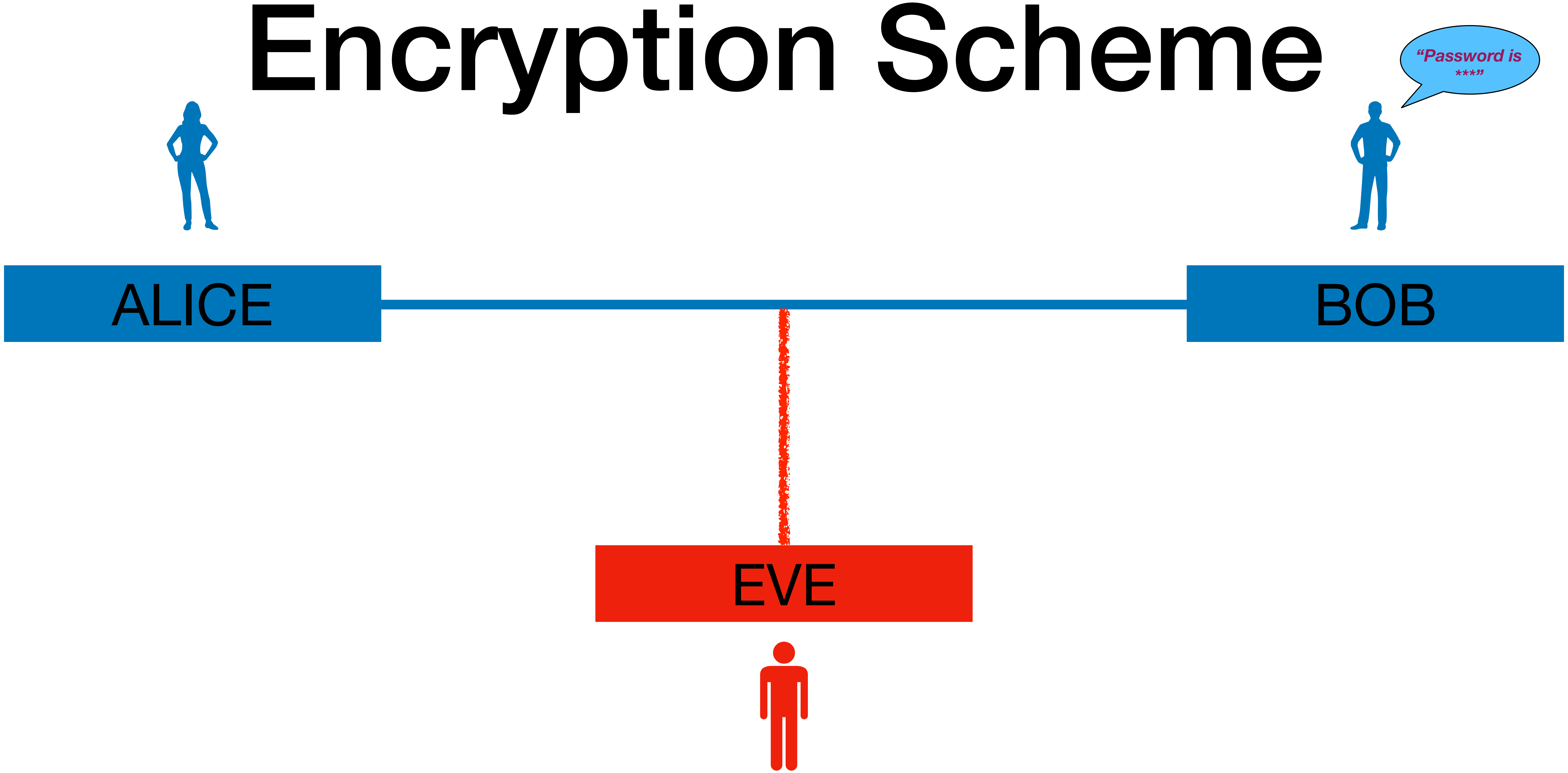


ALICE

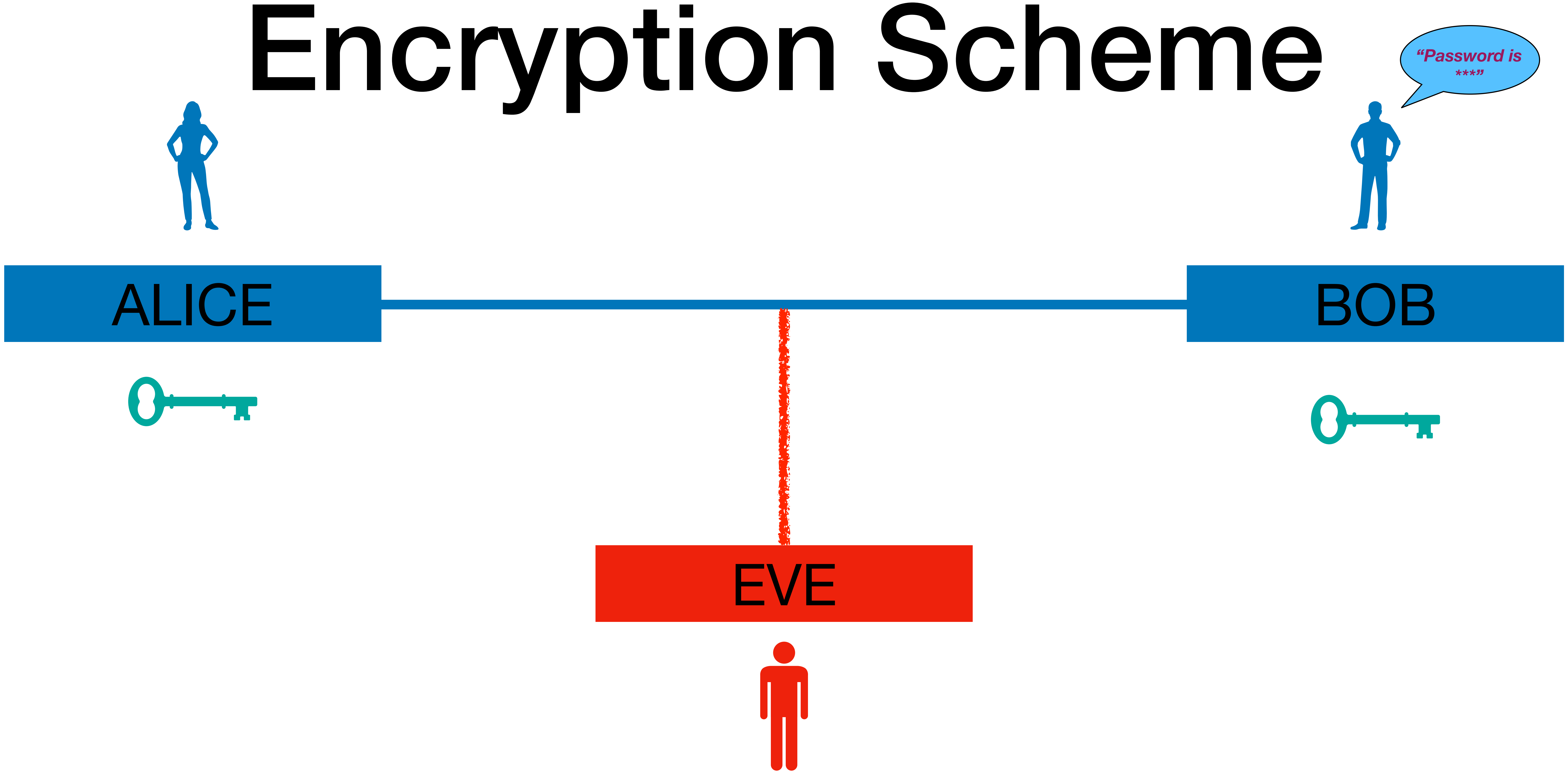
BOB

EVE

Encryption Scheme



Encryption Scheme

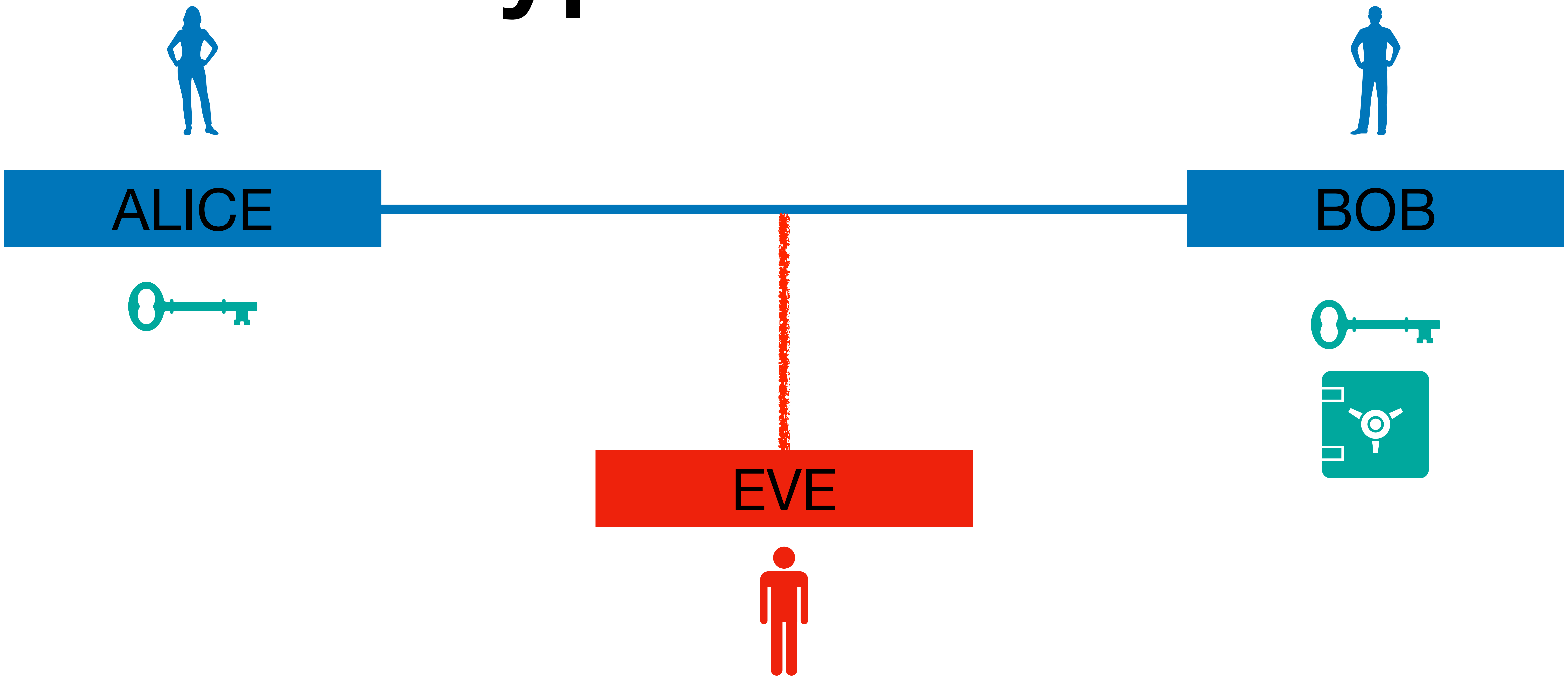


ALICE

BOB

EVE

Encryption Scheme

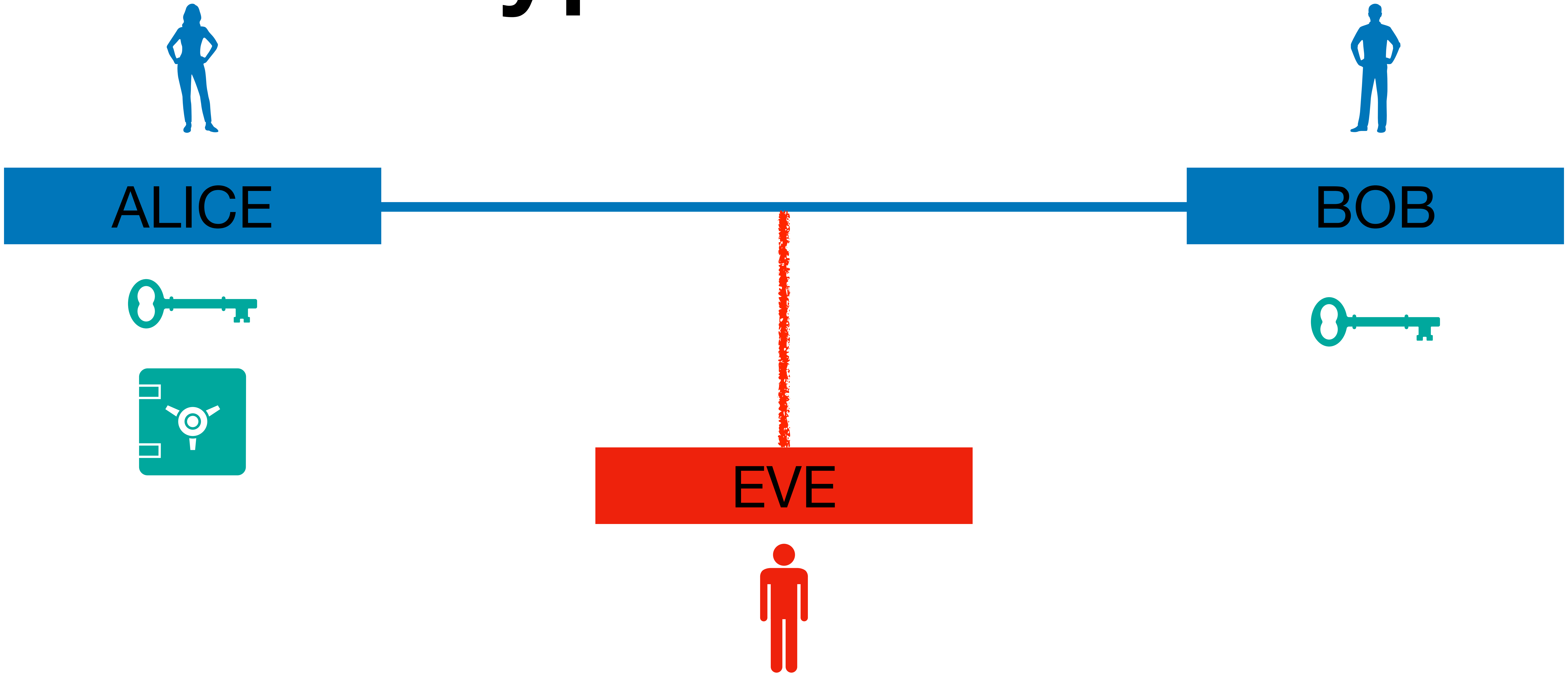


ALICE

BOB

EVE

Encryption Scheme

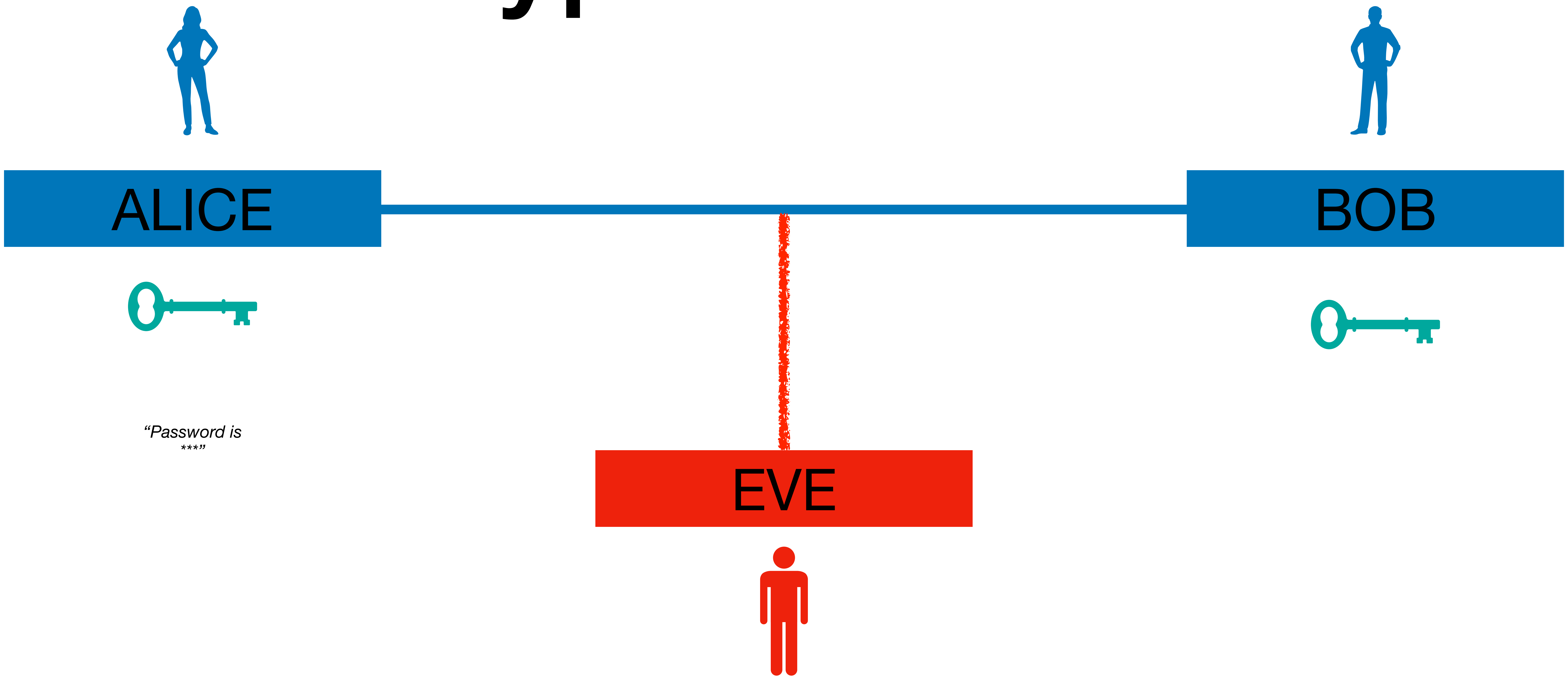


ALICE

BOB

EVE

Encryption Scheme



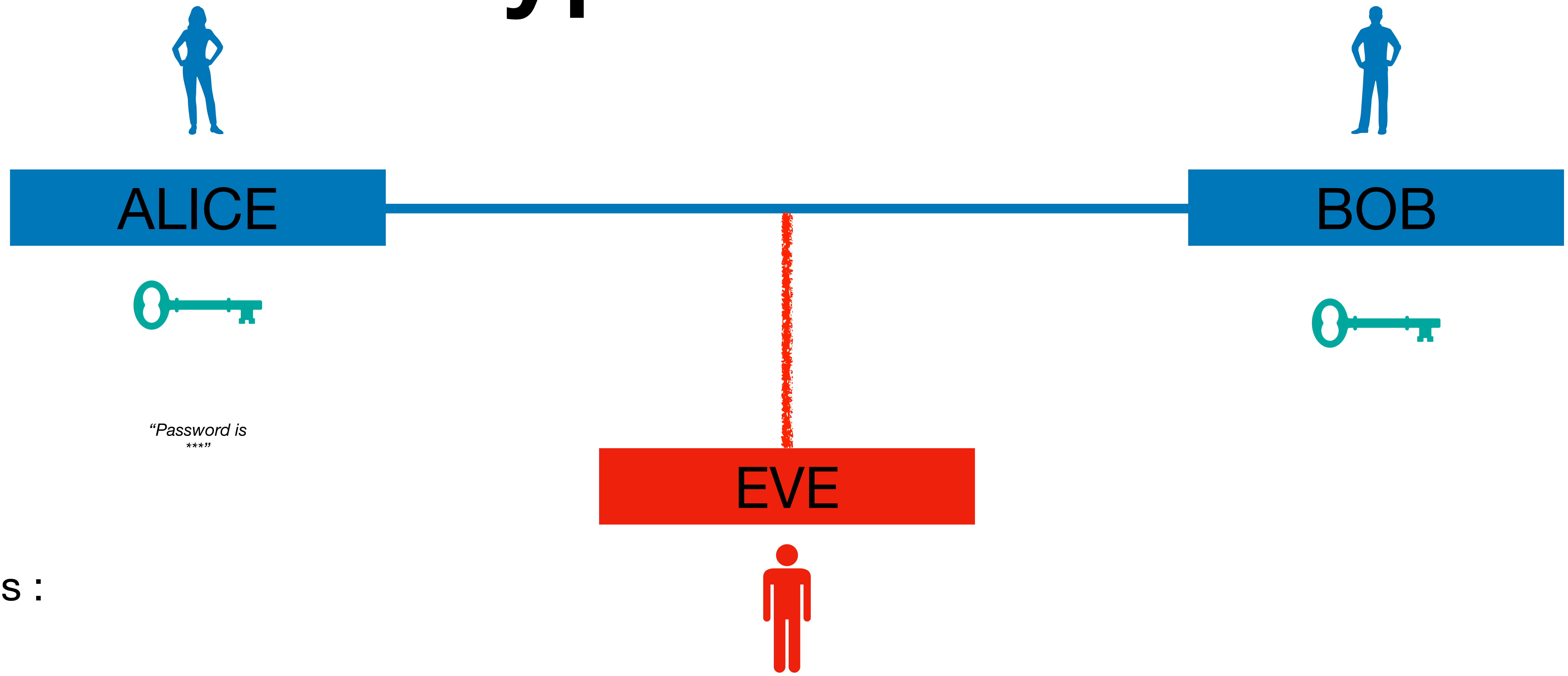
ALICE

BOB

“Password is ***”

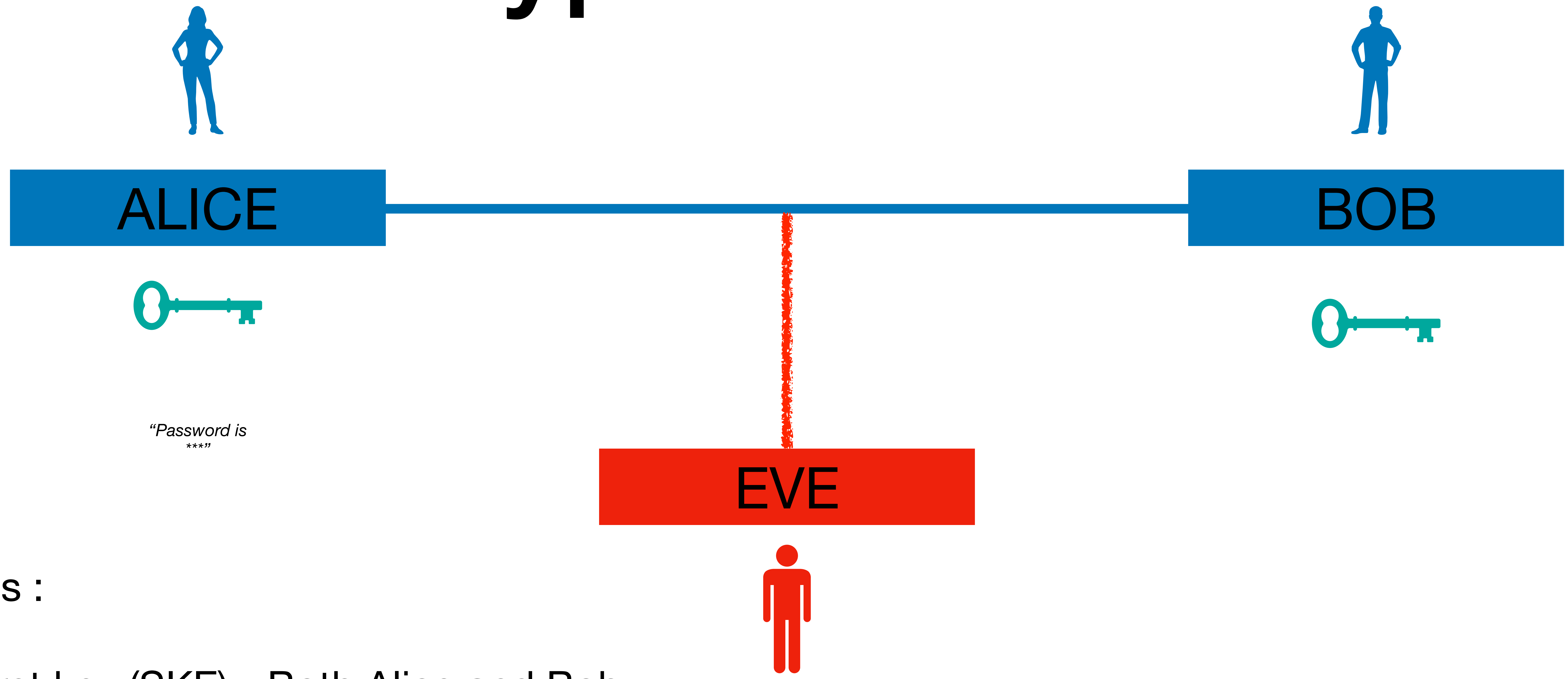
EVE

Encryption Scheme



2 types :

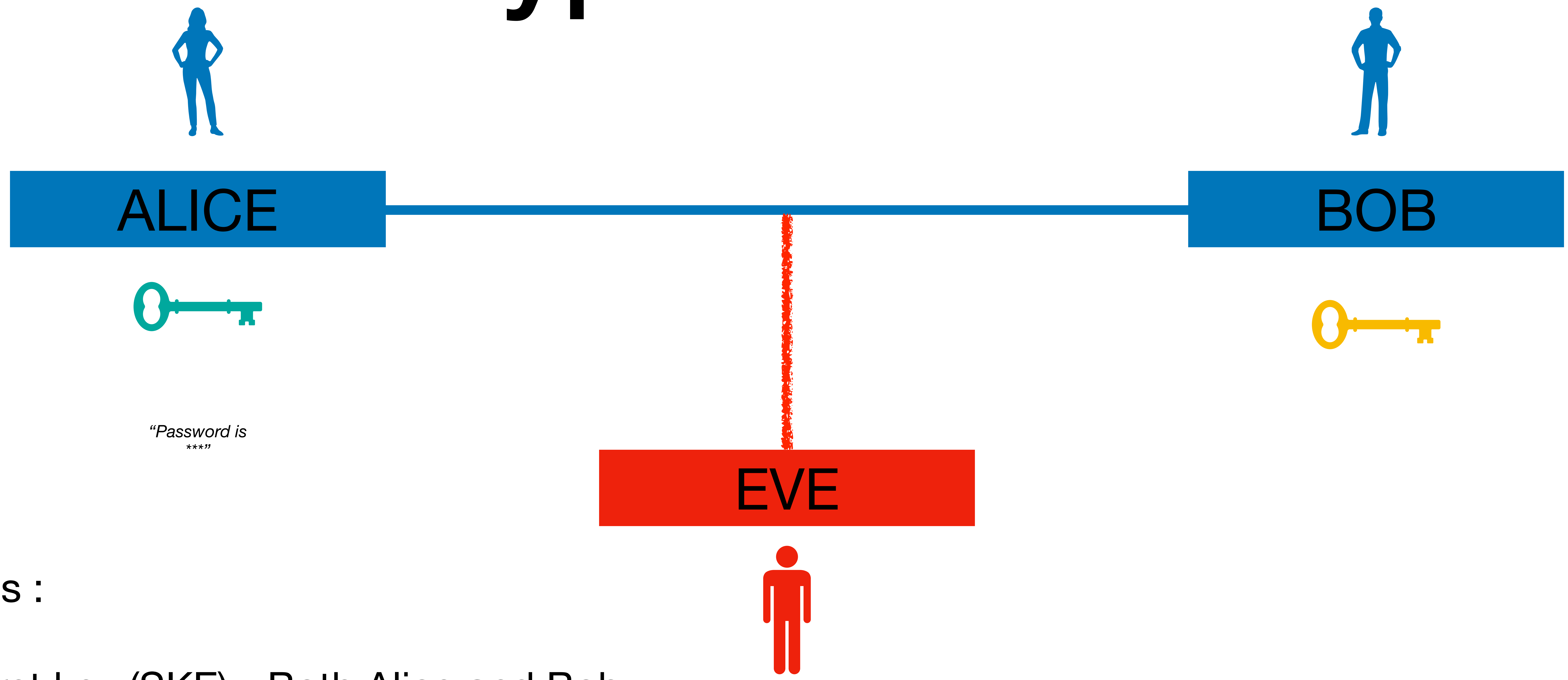
Encryption Scheme



2 types :

- secret key (SKE) - Both Alice and Bob have the same key.

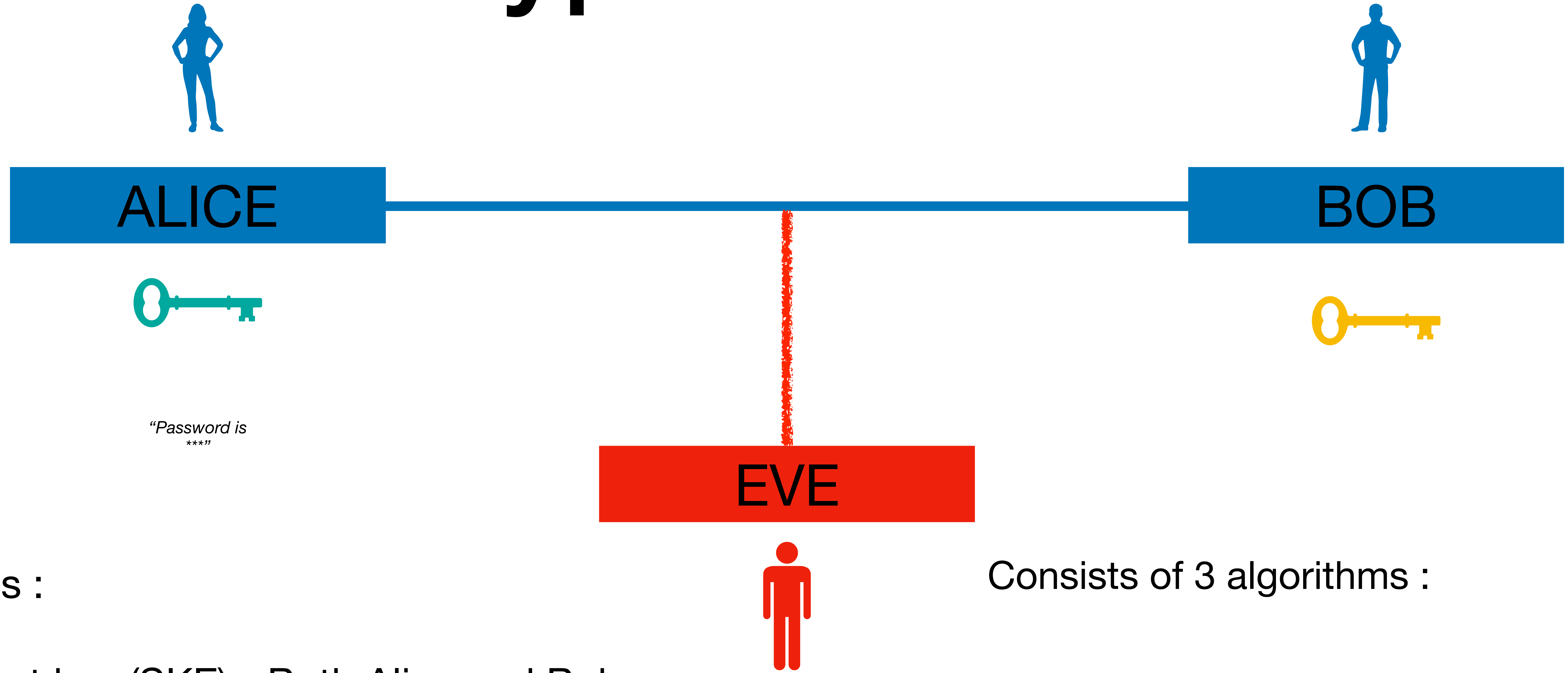
Encryption Scheme



2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Encryption Scheme

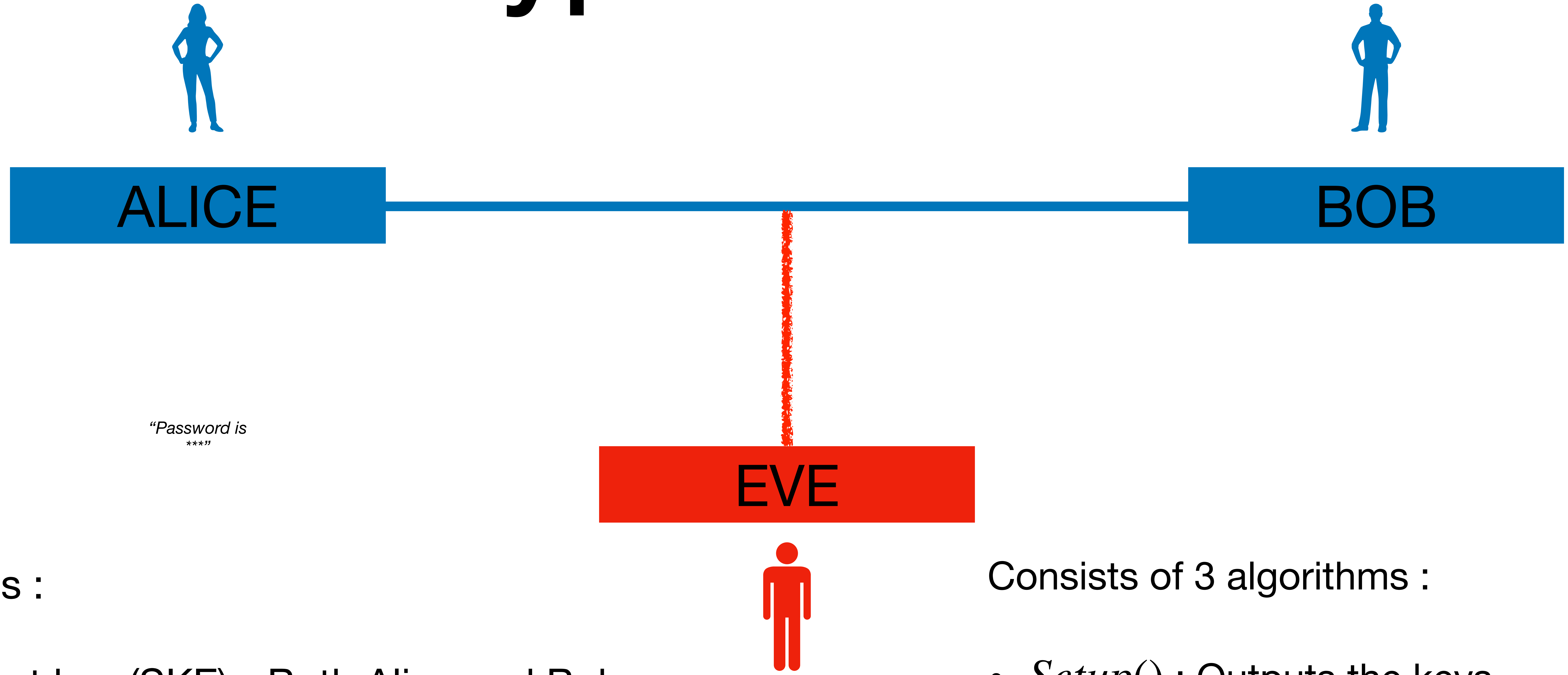


2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Consists of 3 algorithms :

Encryption Scheme


ALICE

BOB

EVE

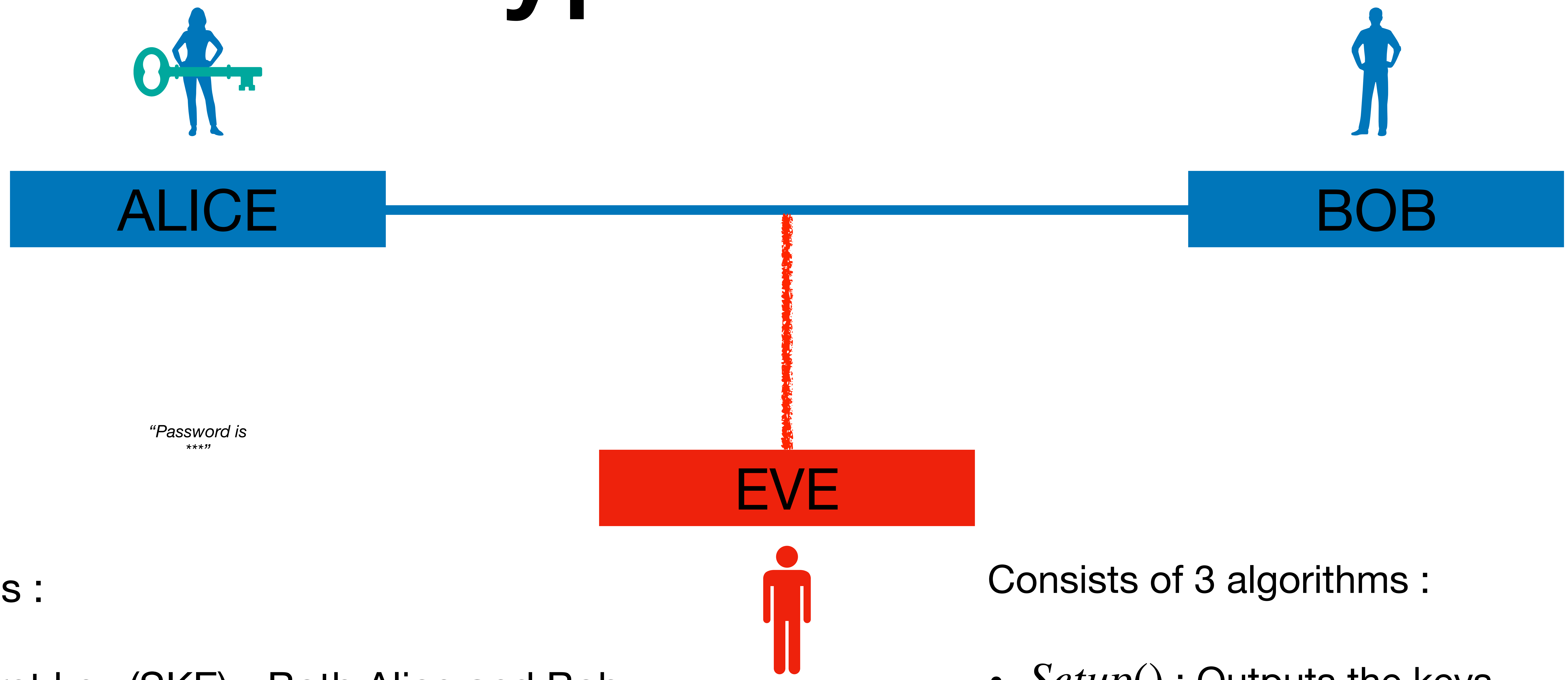
2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Consists of 3 algorithms :

- $Setup()$: Outputs the keys

Encryption Scheme



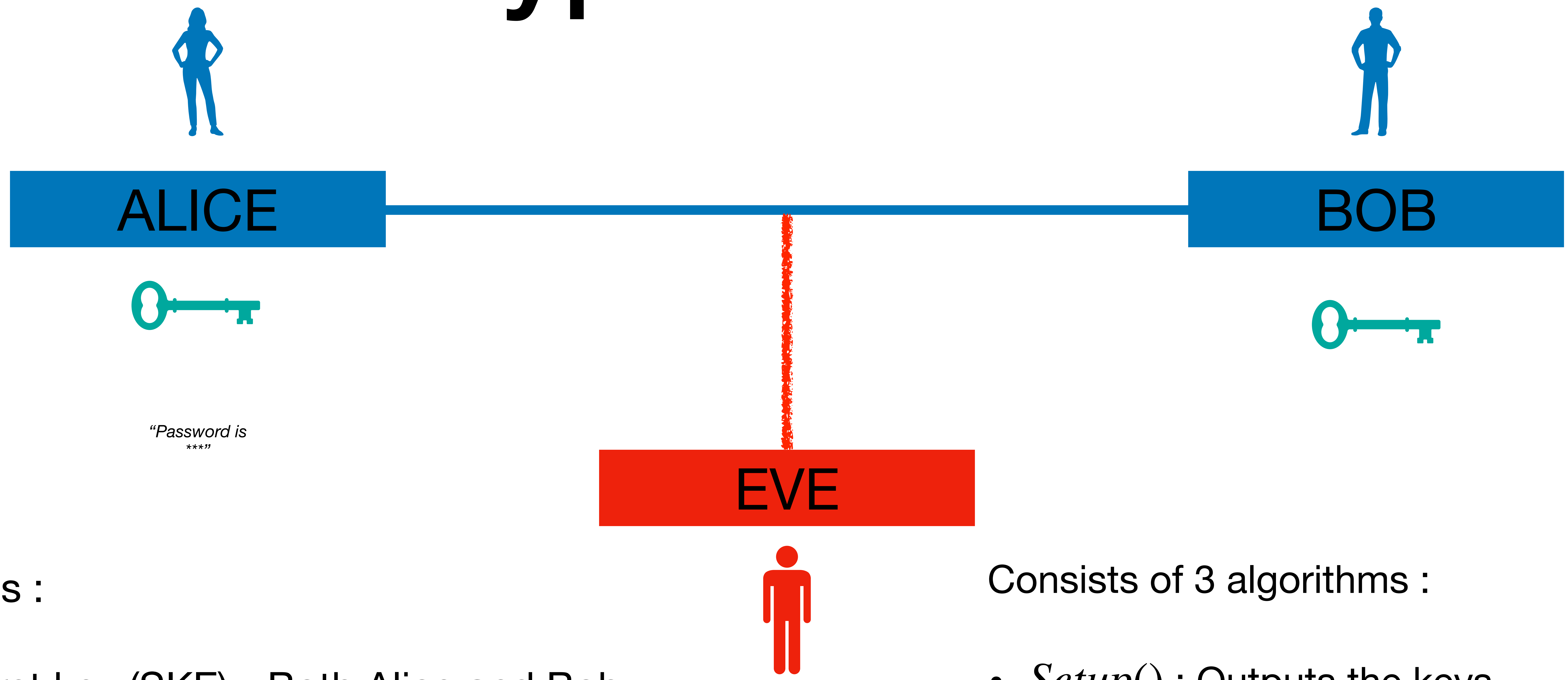
2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Consists of 3 algorithms :

- $Setup()$: Outputs the keys

Encryption Scheme



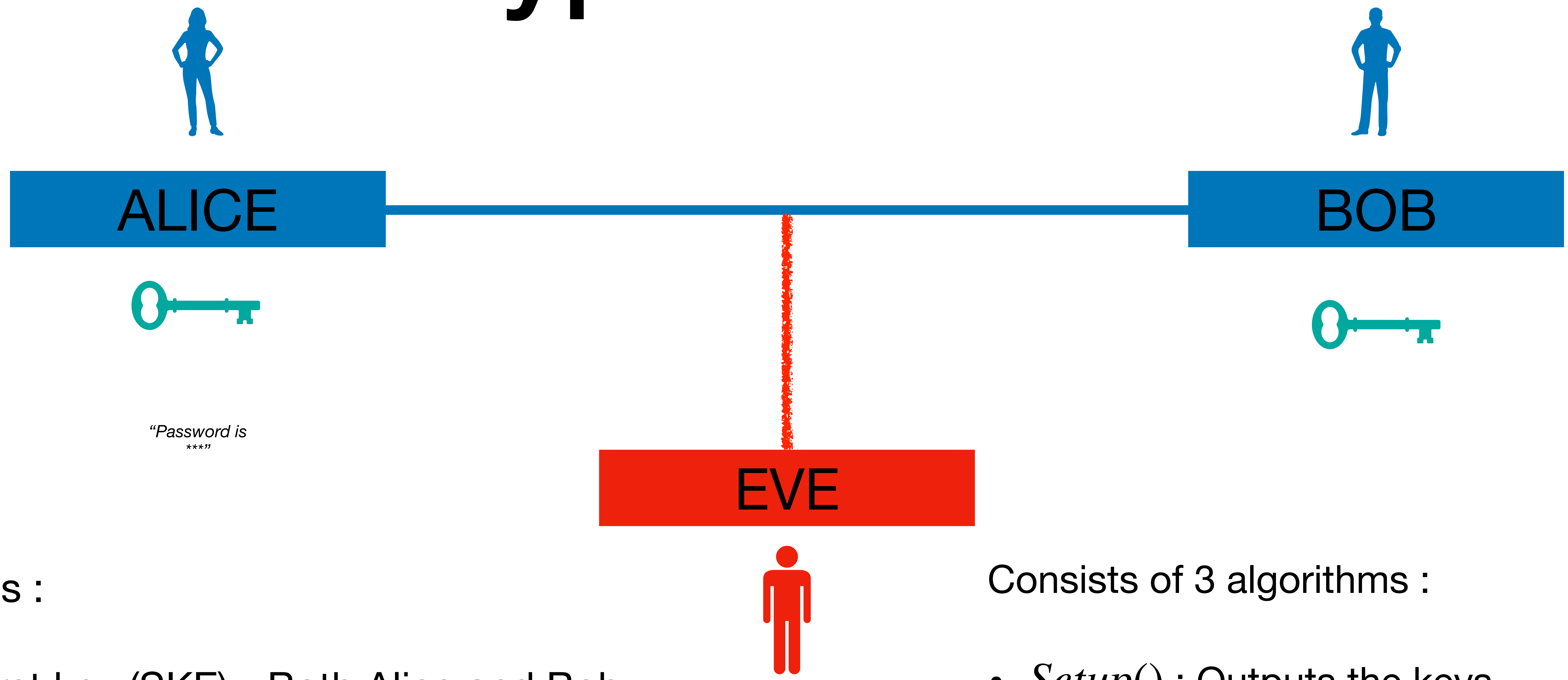
2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Consists of 3 algorithms :

- $Setup()$: Outputs the keys

Encryption Scheme



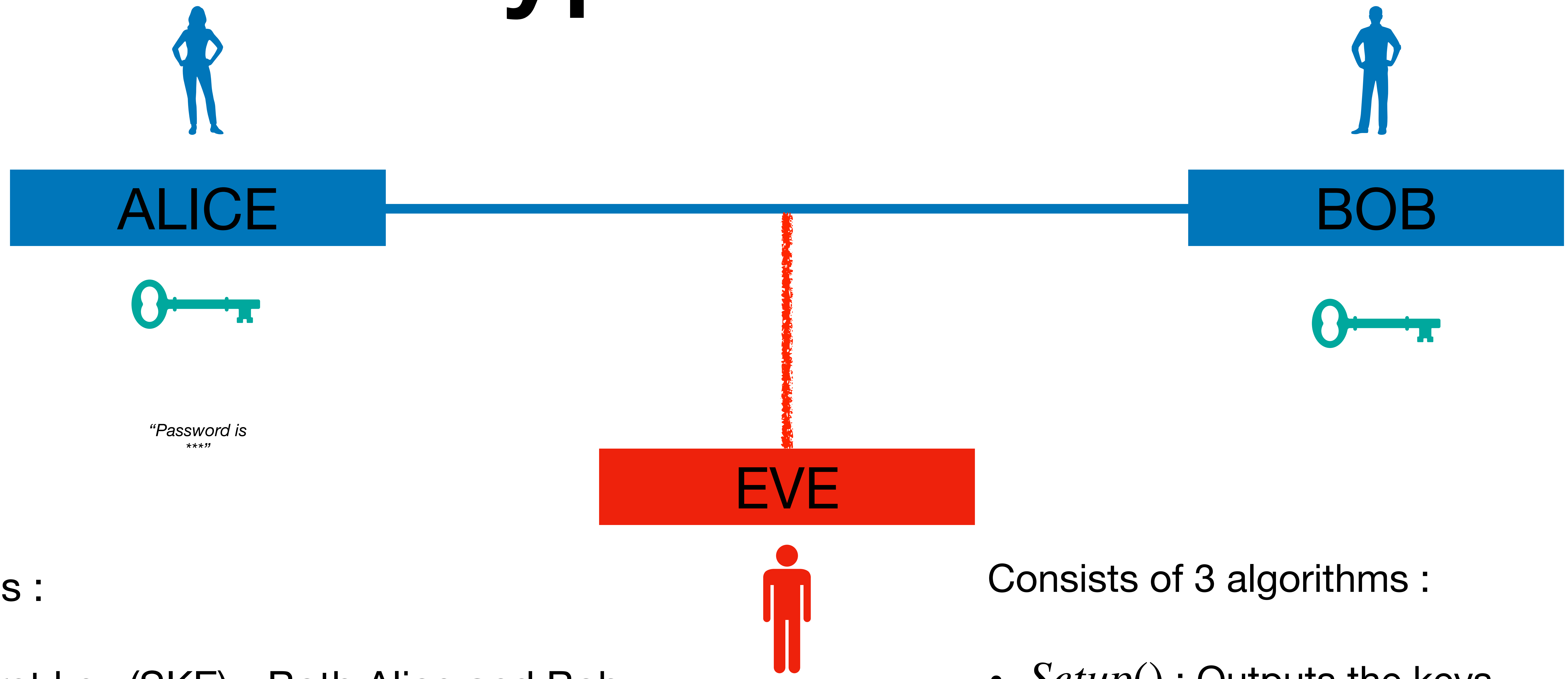
2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Consists of 3 algorithms :

- $Setup()$: Outputs the keys
- $Enc(pk/sk, m)$: Outputs ciphertext

Encryption Scheme



2 types :

- secret key (SKE) - Both Alice and Bob have the same key.
- public key (PKE) - Encryptor has public key and decryption has secret key.

Consists of 3 algorithms :

- $Setup()$: Outputs the keys
- $Enc(pk/sk, m)$: Outputs ciphertext
- $Dec(sk, c)$: Outputs message or error

Public-Key Encryption

Public-Key Encryption

- Diffie,Hellman-76 presented the first key exchanged photocol.

Public-Key Encryption

- Diffie,Hellman-76 presented the first key exchanged photocol.
- RSA cryptosystem was introduced in 1977.

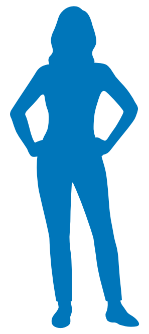
Public-Key Encryption

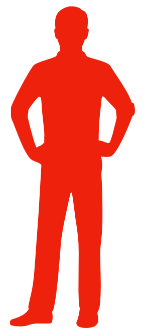
- Diffie,Hellman-76 presented the first key exchanged photocol.
- RSA cryptosystem was introduced in 1977.
- Goldwaser,Micali-84 proposed semantic security.

Security Definitions

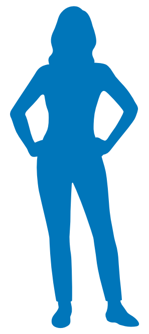
Standard Security [Goldwaser,Micali-84]

Standard Security [Goldwaser,Micali-84]

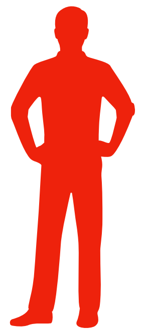




Standard Security [Goldwaser,Micali-84]



Challenger



Adversary

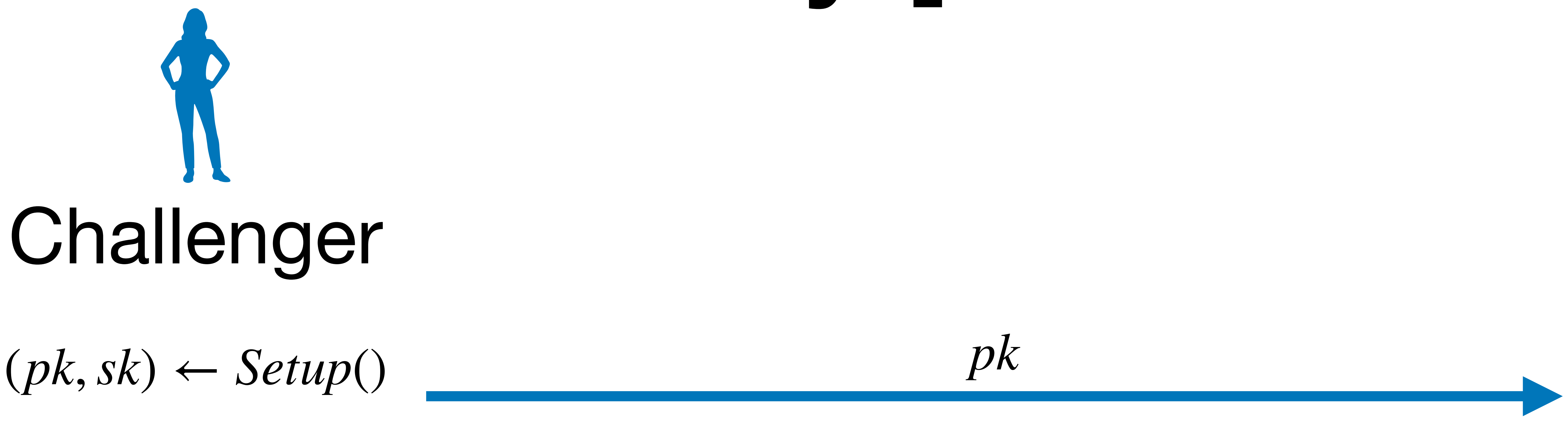
Standard Security [Goldwaser,Micali-84]

Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

Standard Security [Goldwaser,Micali-84]



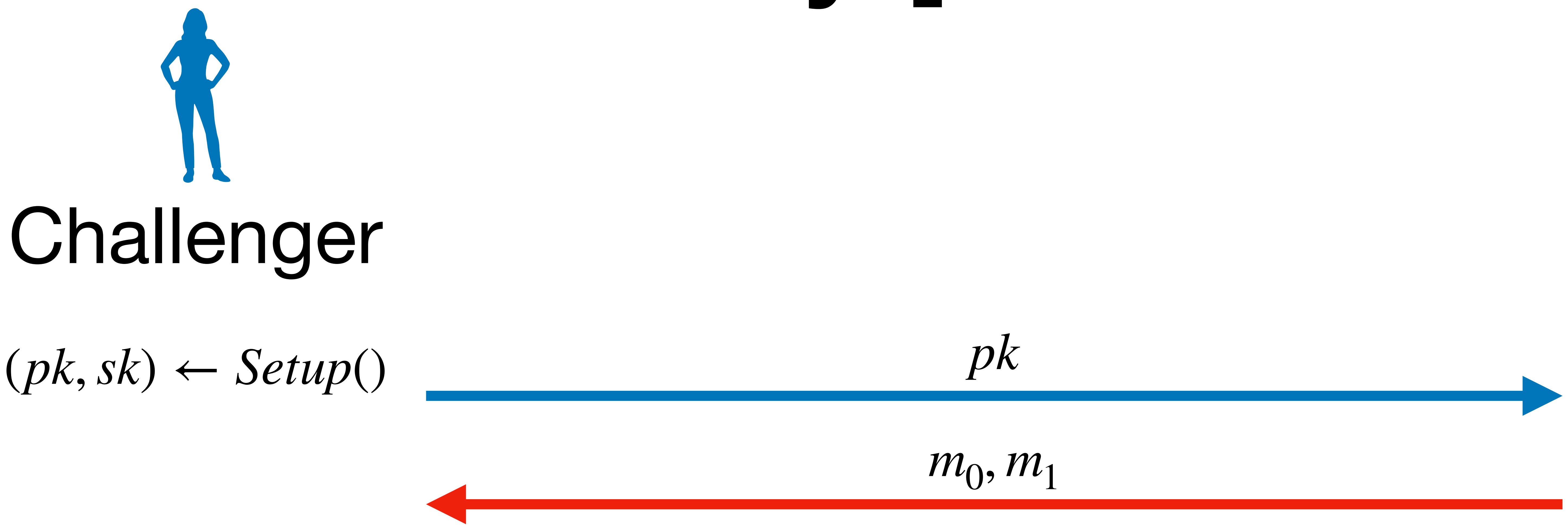
Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

$\xrightarrow{pk}$

Standard Security [Goldwaser,Micali-84]



Challenger

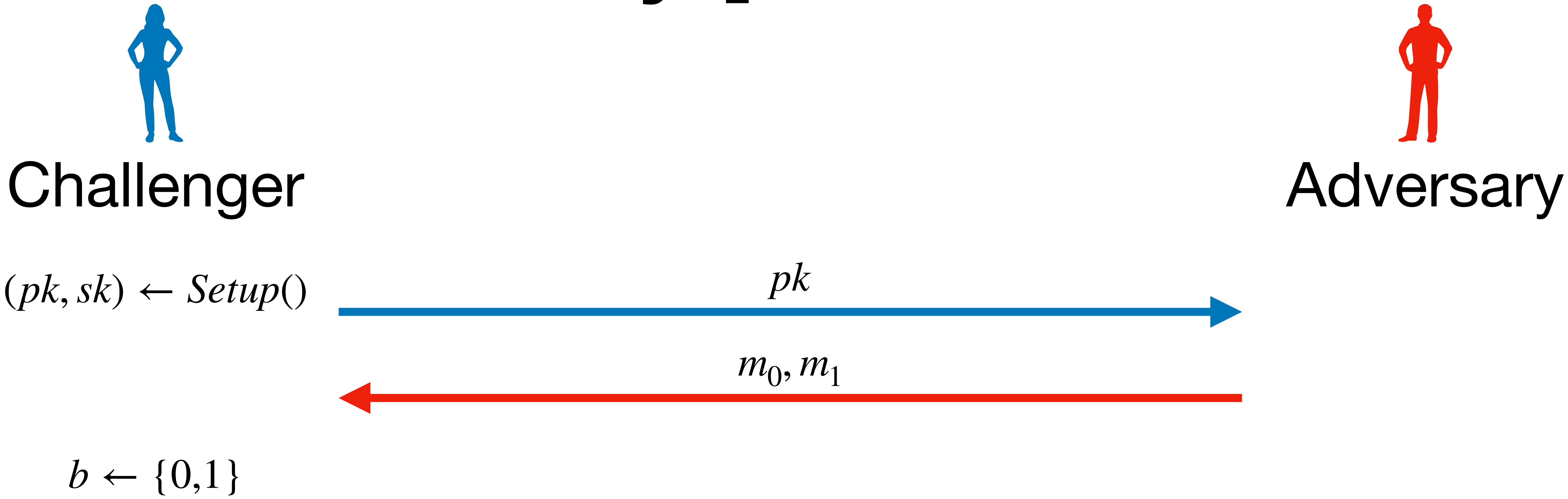
Adversary

$(pk, sk) \leftarrow Setup()$

pk

m_0, m_1

Standard Security [Goldwaser,Micali-84]



Challenger

Adversary

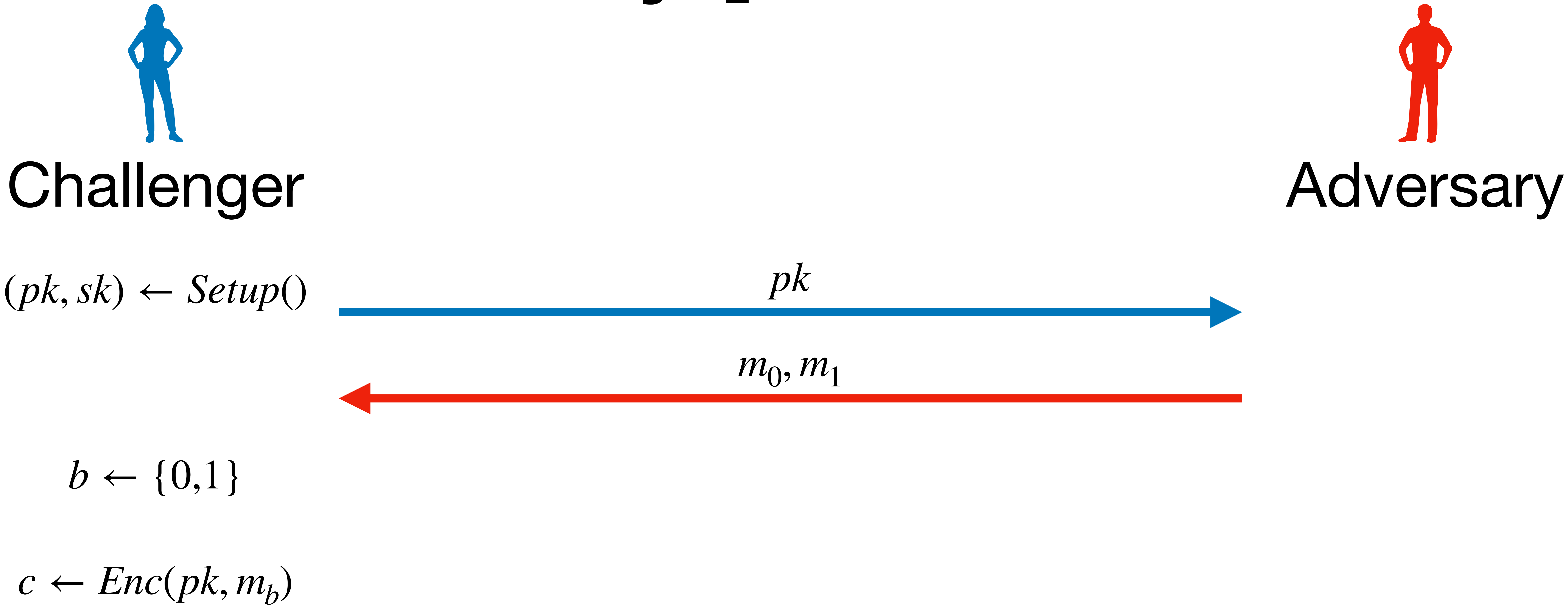
$(pk, sk) \leftarrow Setup()$

pk

m_0, m_1

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

Standard Security [Goldwaser,Micali-84]



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

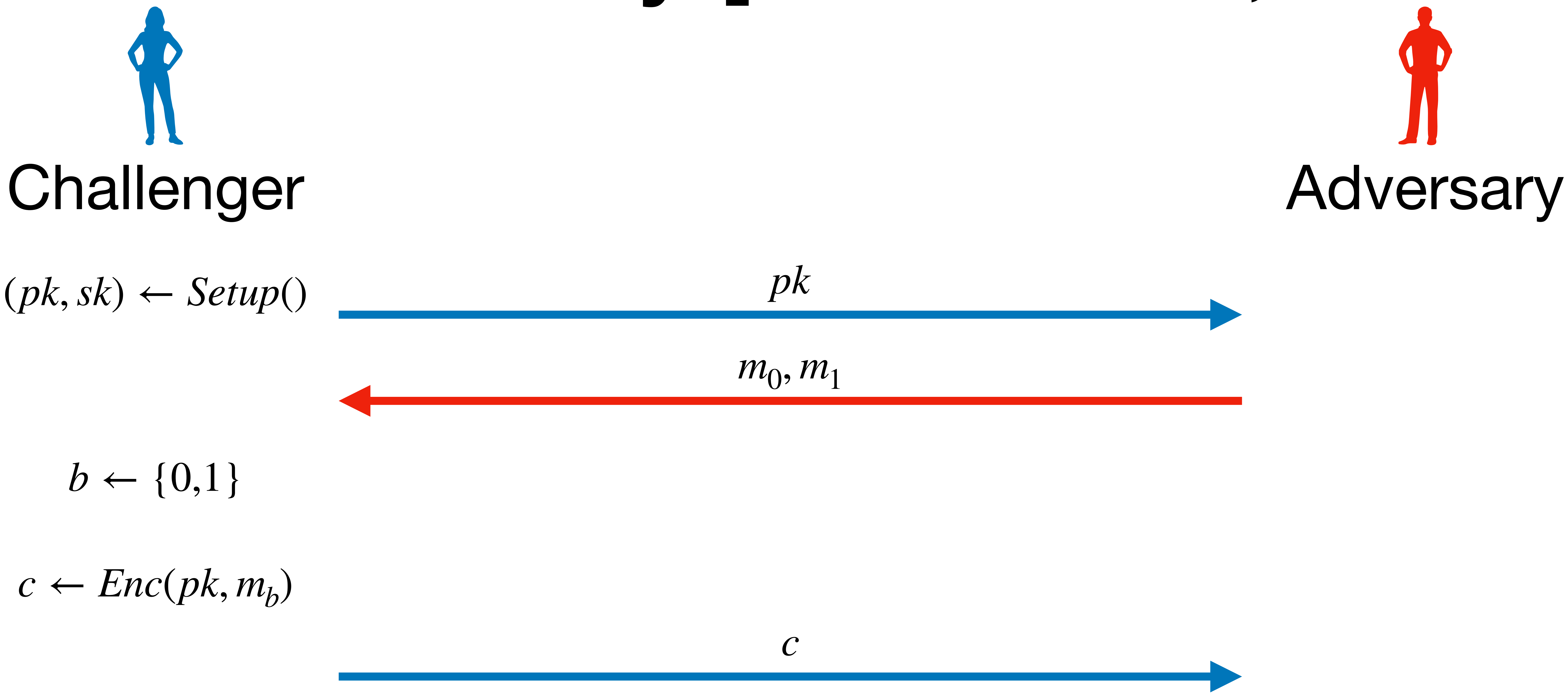
pk

m_0, m_1

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

$c \leftarrow Enc(pk, m_b)$

Standard Security [Goldwaser,Micali-84]



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

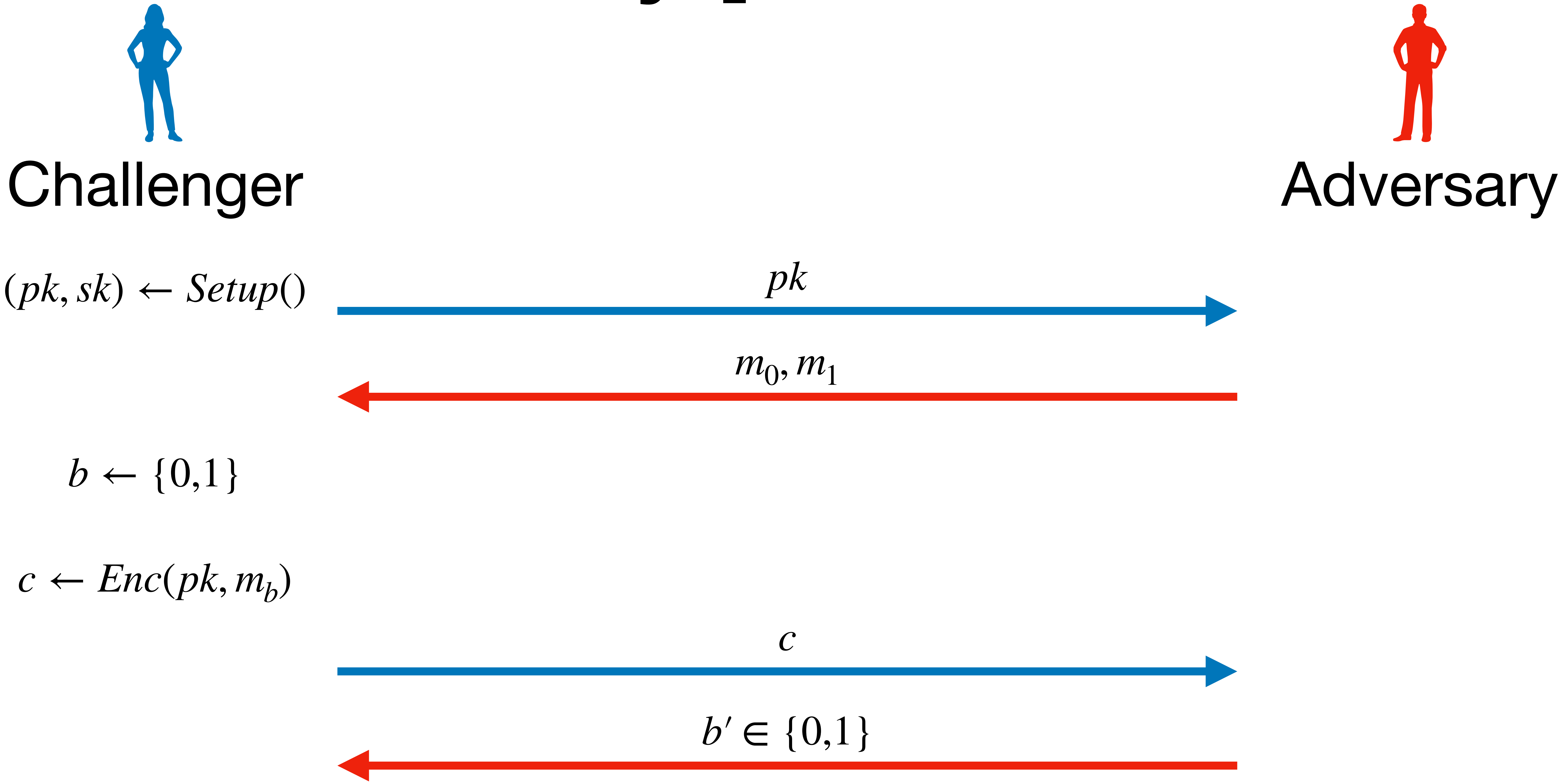
m_0, m_1

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

$c \leftarrow Enc(pk, m_b)$

c

Standard Security [Goldwaser,Micali-84]



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

m_0, m_1

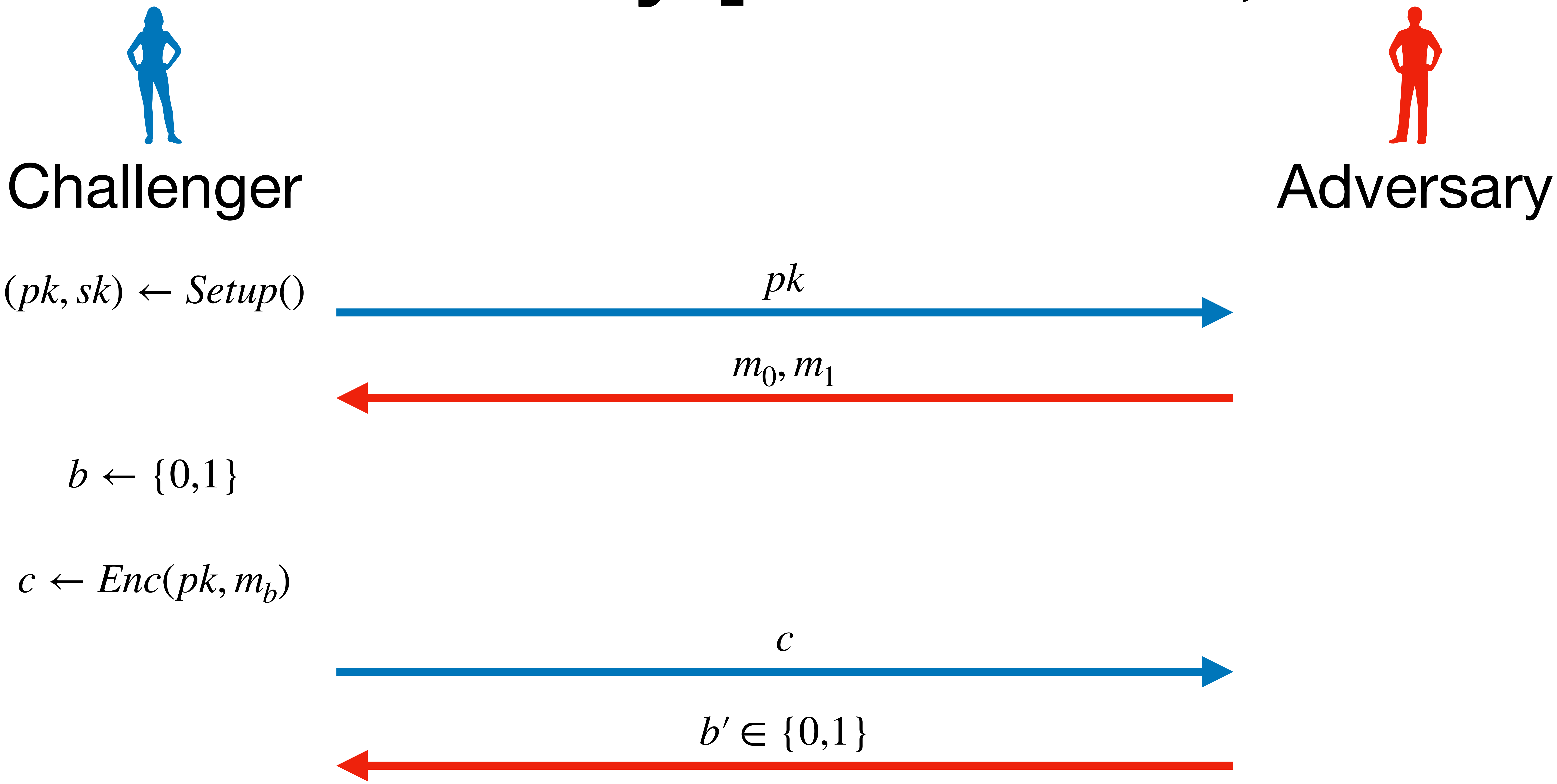
$b \leftarrow \{0,1\}$

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

Standard Security [Goldwaser,Micali-84]



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$ $\xrightarrow{pk}$

$\xleftarrow{m_0, m_1}$

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

$c \leftarrow Enc(pk, m_b)$

$\xrightarrow{c}$

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

Adversary wins if $b = b'$

More Security Notions

More Security Notions

- Chosen-Ciphertext Attacks

More Security Notions

- Chosen-Ciphertext Attacks
- Non-malleable

More Security Notions

- Chosen-Ciphertext Attacks
- Non-malleable
- Leakage-Resilient

More Security Notions

- Chosen-Ciphertext Attacks
- Non-malleable
- Leakage-Resilient
- Key-Dependent Message

More Security Notions

- Chosen-Ciphertext Attacks
- Non-malleable
- Leakage-Resilient
- Key-Dependent Message
- Selective Opening

More Security Notions

- Chosen-Ciphertext Attacks
- Non-malleable
- Leakage-Resilient
- Key-Dependent Message
- Selective Opening
- Incompressible

Can Secret Key be leaked?

Can Secret Key be leaked?

- Standard security says that adversary cannot distinguish between encryptions of two different message provided **no** information of secret key is leaked.

Can Secret Key be leaked?

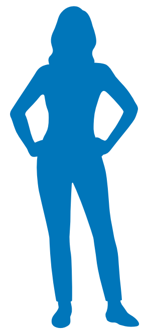
- Standard security says that adversary cannot distinguish between encryptions of two different message provided **no** information of secret key is leaked.

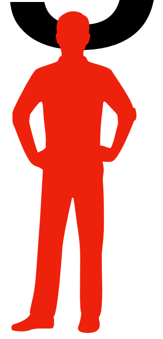
- In practice, secret key can be leaked using side-channel attacks.

Leakage-Resilience

Security against Leakage

Security against Leakage





Security against Leakage

Challenger

Adversary

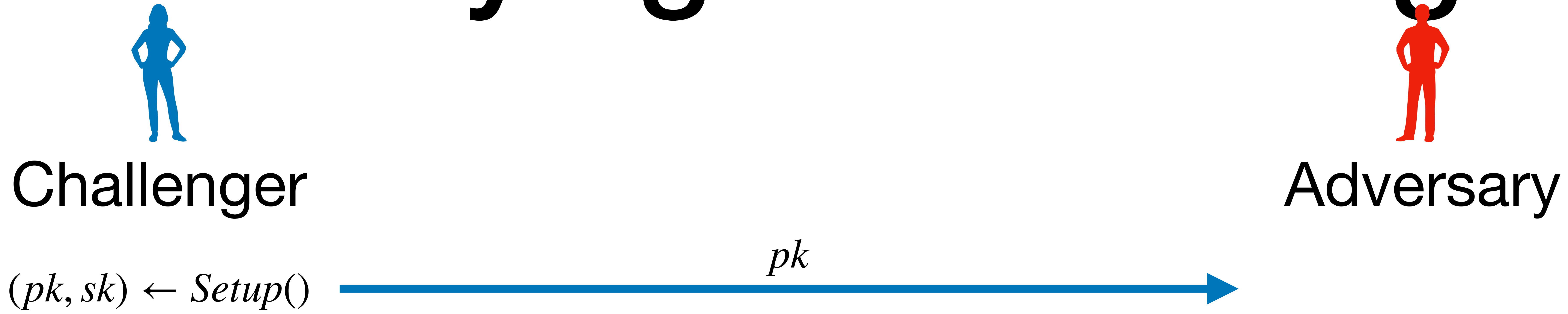
Security against Leakage

Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

Security against Leakage



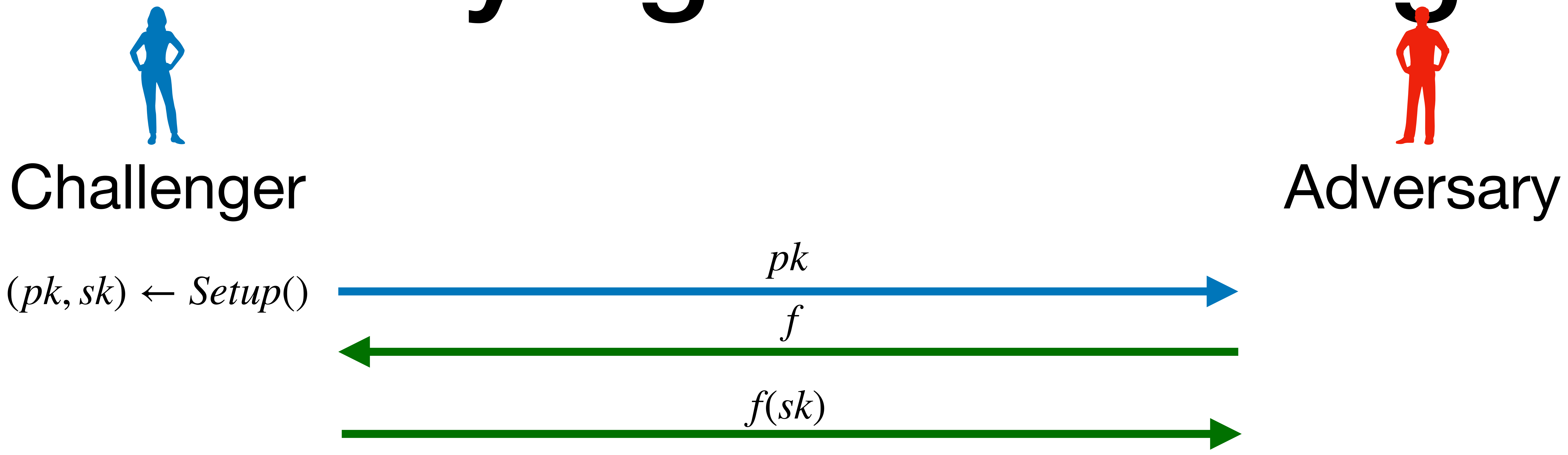
Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

Security against Leakage



Challenger

Adversary

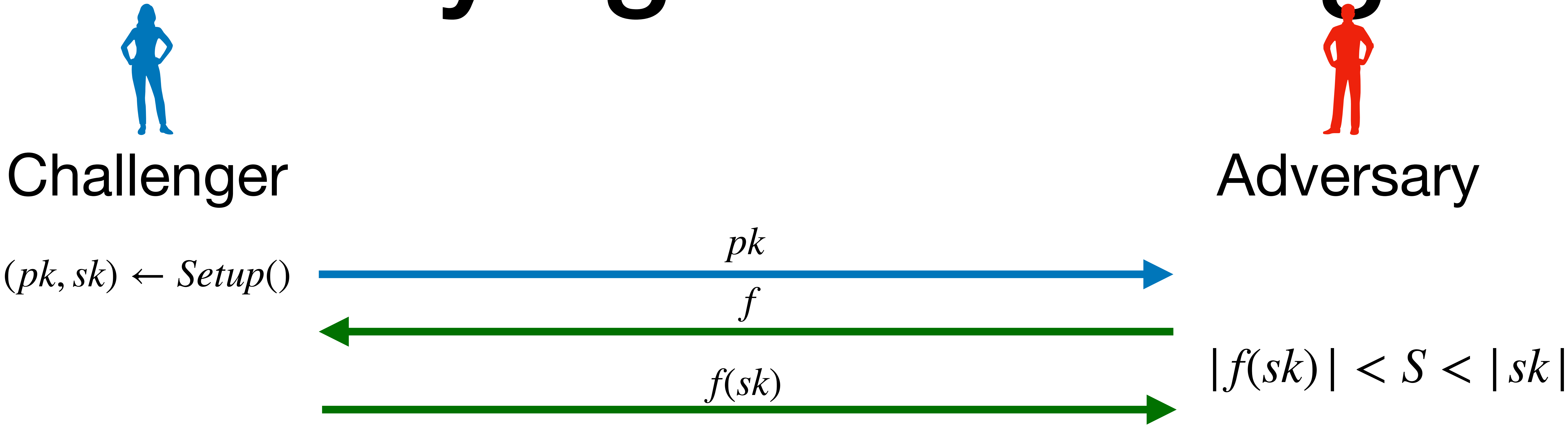
$(pk, sk) \leftarrow Setup()$

pk

f

$f(sk)$

Security against Leakage



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

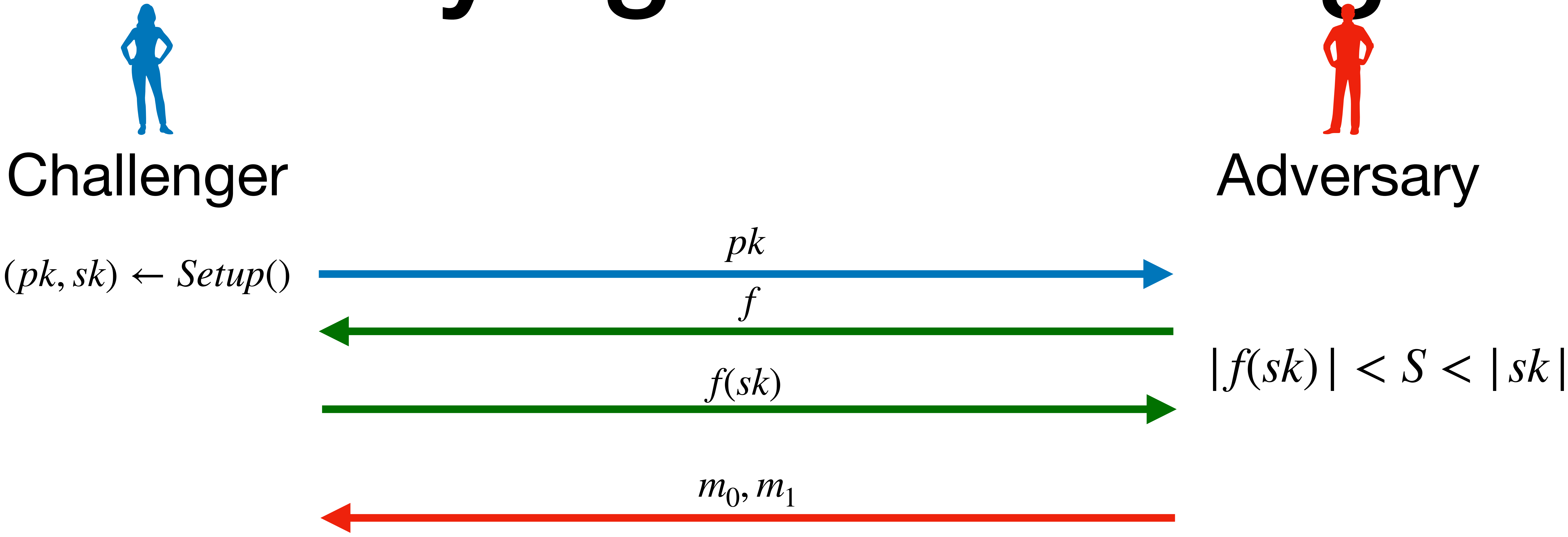
pk

f

$f(sk)$

$|f(sk)| < S < |sk|$

Security against Leakage



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

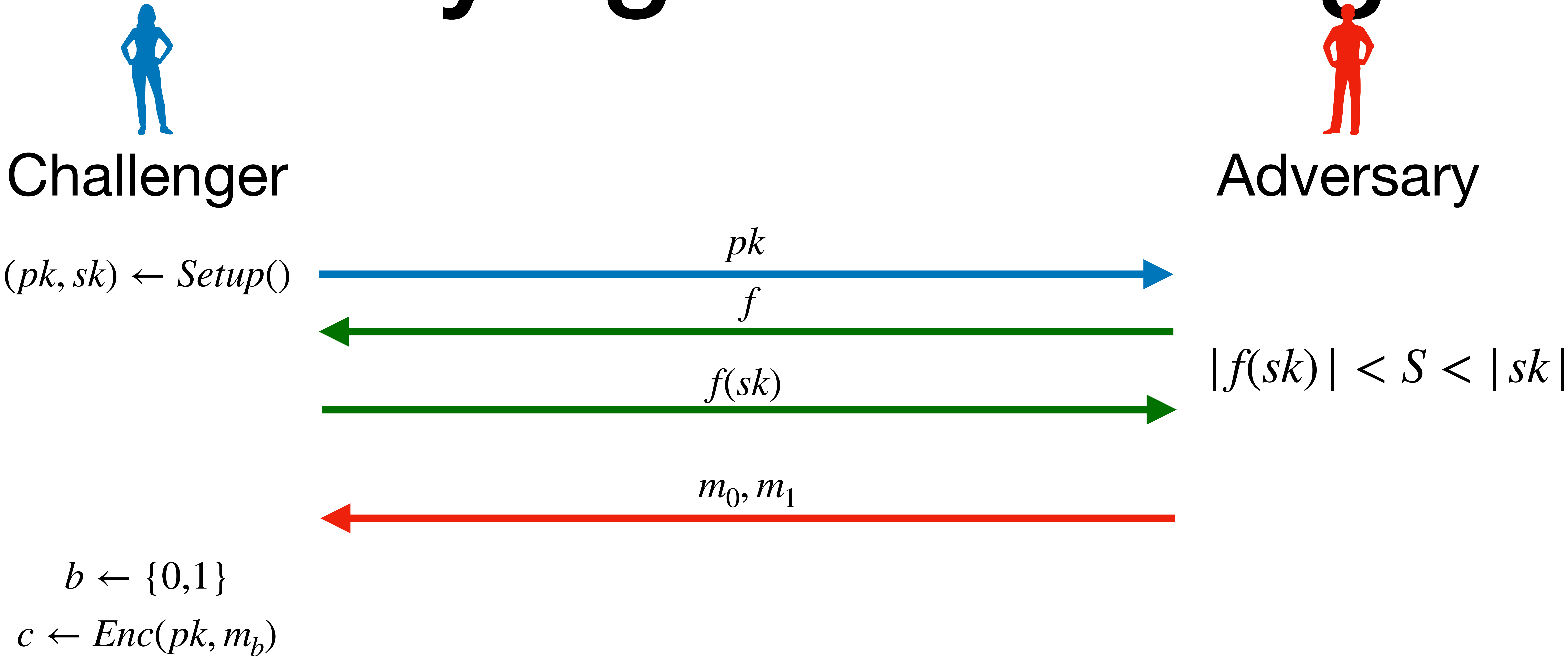
f

$f(sk)$

$|f(sk)| < S < |sk|$

m_0, m_1

Security against Leakage



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$f(sk)$

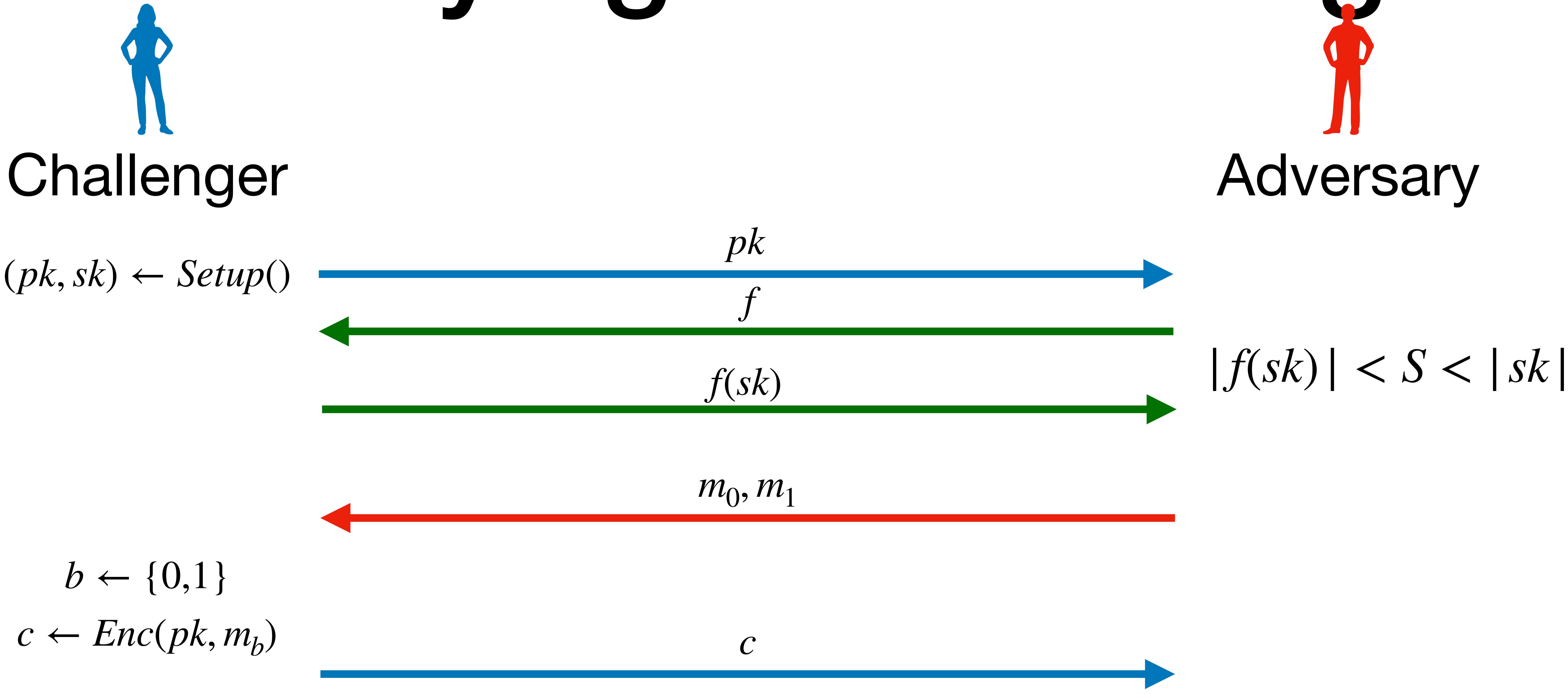
$|f(sk)| < S < |sk|$

m_0, m_1

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

$c \leftarrow Enc(pk, m_b)$

Security against Leakage



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$f(sk)$

$|f(sk)| < S < |sk|$

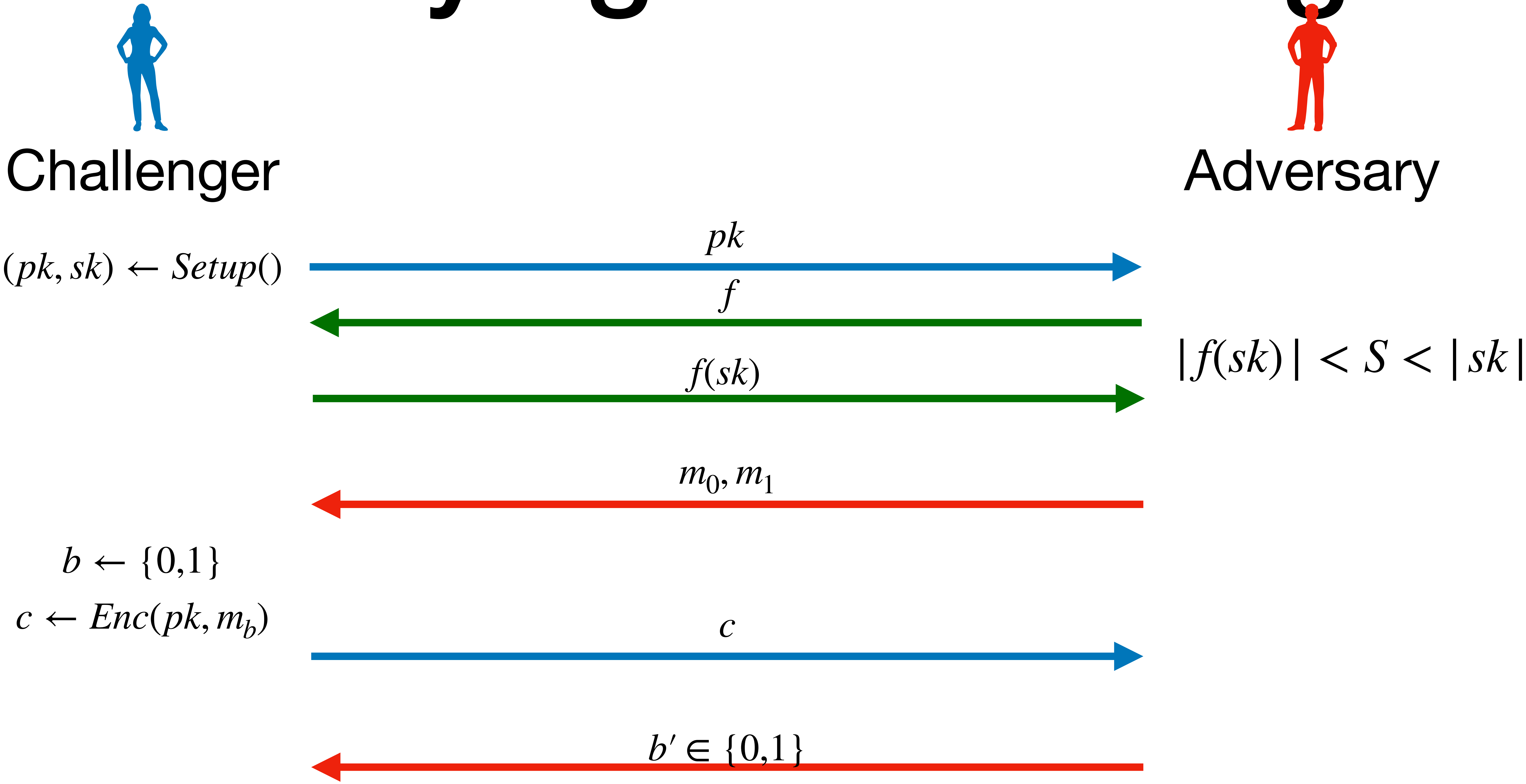
m_0, m_1

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

$c \leftarrow Enc(pk, m_b)$

c

Security against Leakage



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$f(sk)$

$|f(sk)| < S < |sk|$

m_0, m_1

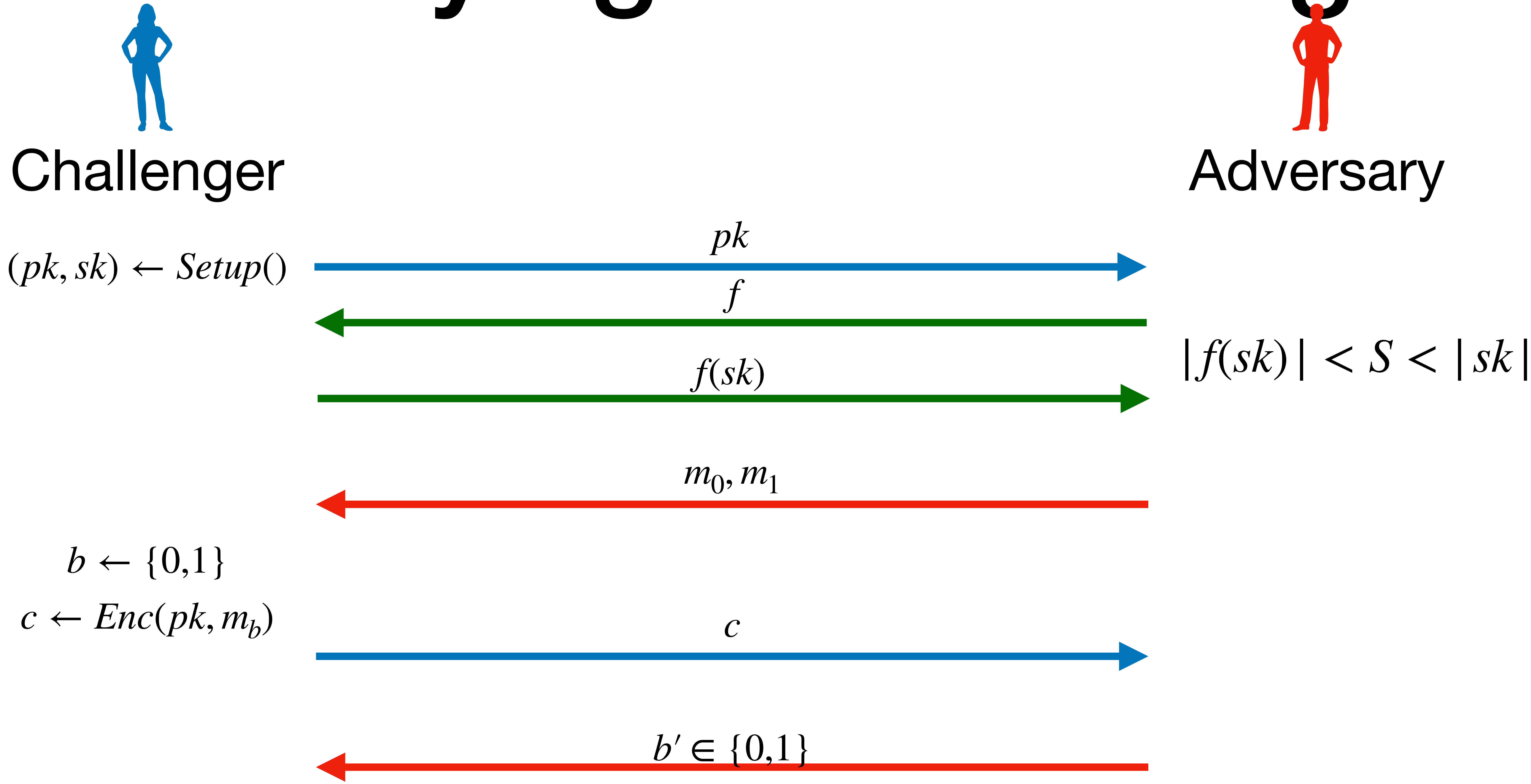
$b \leftarrow \{0,1\}$

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

Security against Leakage



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$f(sk)$

$|f(sk)| < S < |sk|$

m_0, m_1

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

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

Adversary wins if $b = b'$

Leakage Resilient Schemes

Leakage Resilient Schemes

- Canetti et al.-00 and Dodis et al.-01 gave construction where f returns bits of sk.

Leakage Resilient Schemes

- Canetti et al.-00 and Dodis et al.-01 gave construction where f returns bits of sk.

- Dziembowski-06, Di Crescenzo et al.-06, Akavia et al.-09, etc. considered arbitrary function f.

Leakage Resilient Schemes

- Canetti et al.-00 and Dodis et al.-01 gave construction where f returns bits of sk.

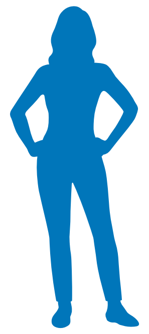
- Dziembowski-06, Di Crescenzo et al.-06, Akavia et al.-09, etc. considered arbitrary function f.

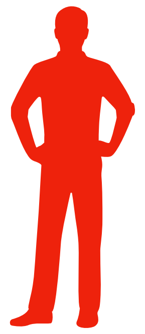
- Other works include Dodis et al.-09, Brakerski et al.-10, Dodis et al.-10, Faonio et al.-15 and many more.

Key-Dependent Message Security

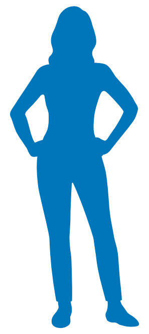
KDM Security

KDM Security

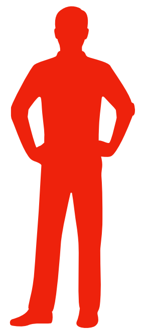




KDM Security



Challenger



Adversary

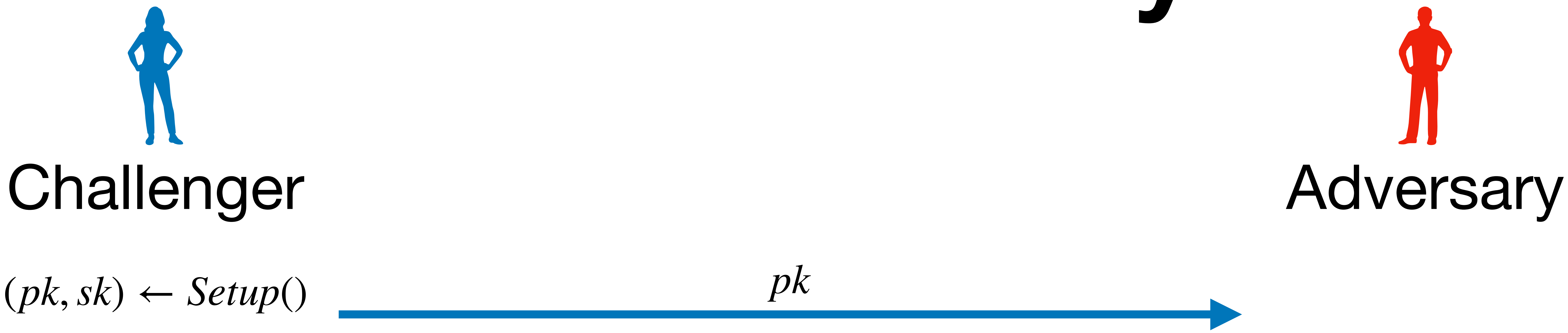
KDM Security

Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

KDM Security



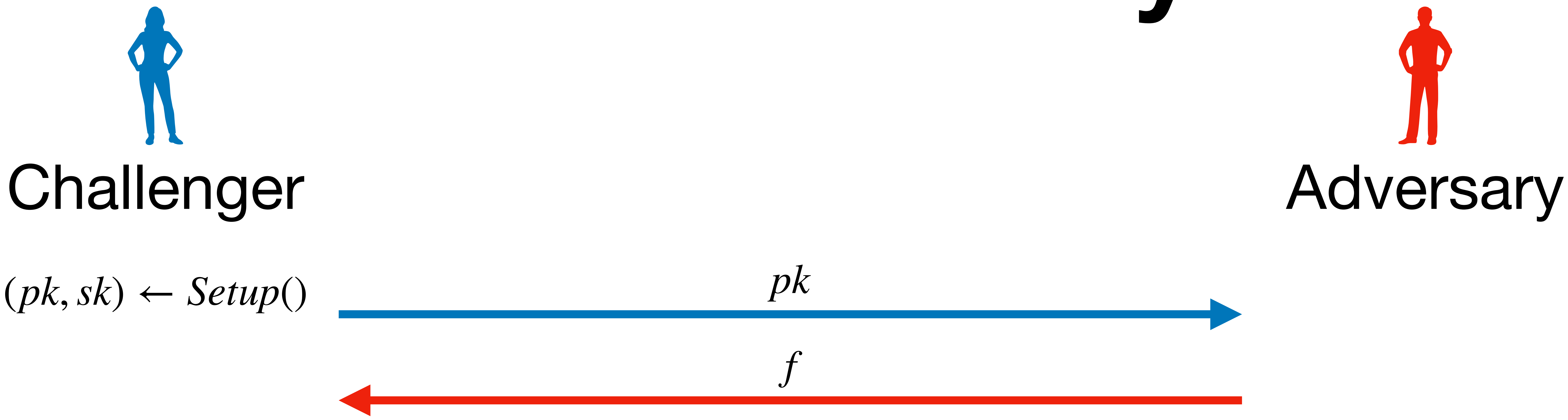
Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

KDM Security



Challenger

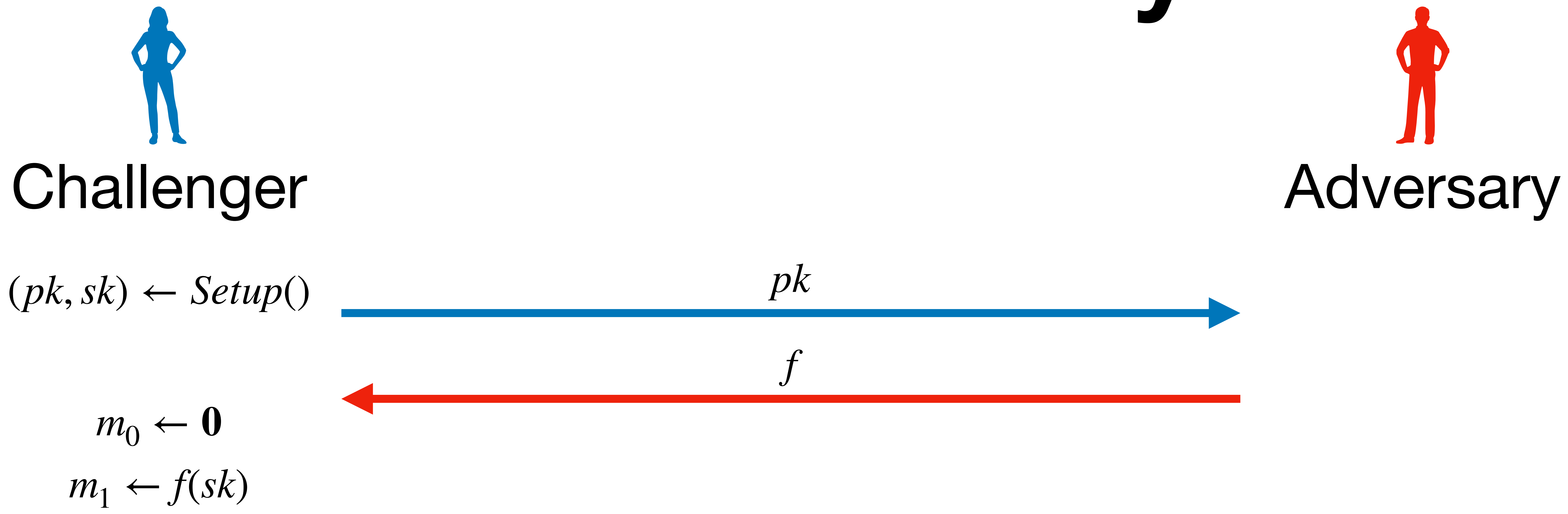
Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

KDM Security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

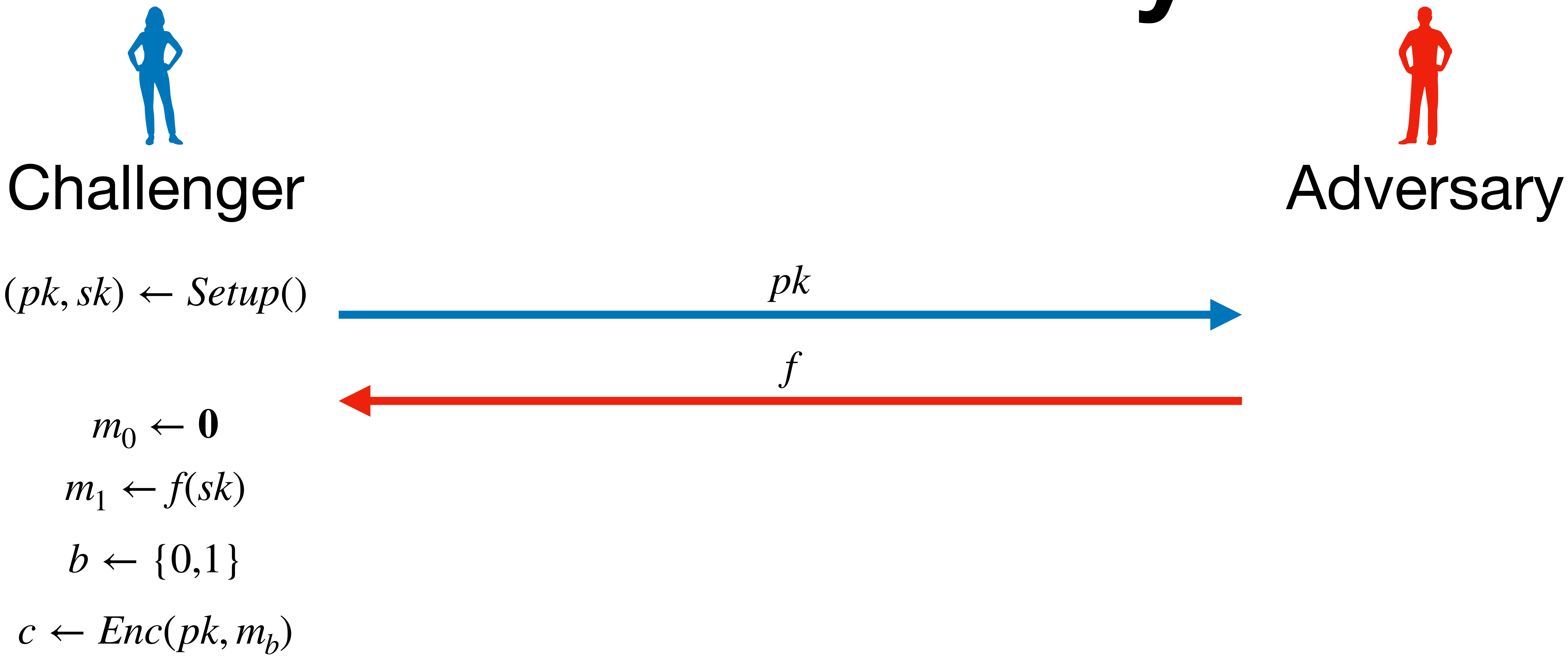
pk

f

$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

KDM Security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

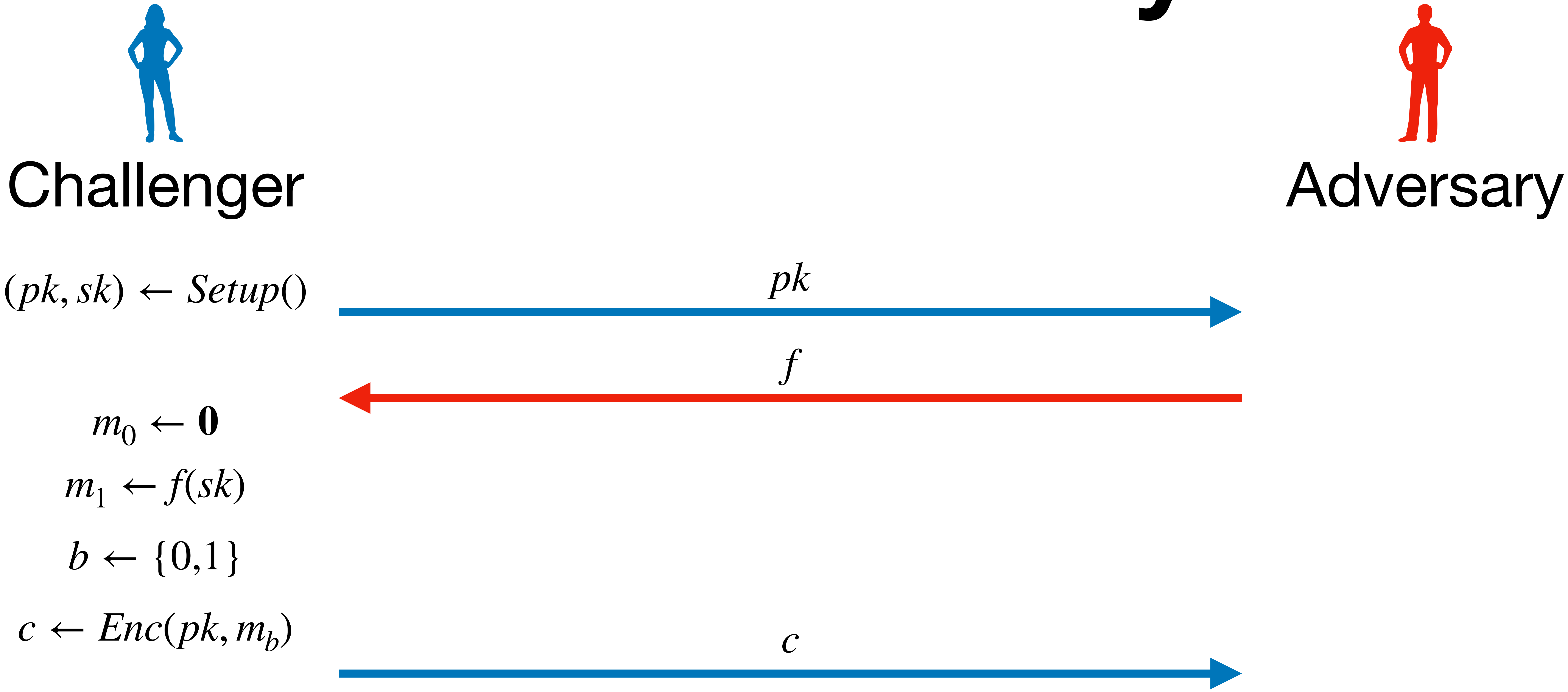
$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

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

$c \leftarrow Enc(pk, m_b)$

KDM Security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$m_0 \leftarrow \mathbf{0}$

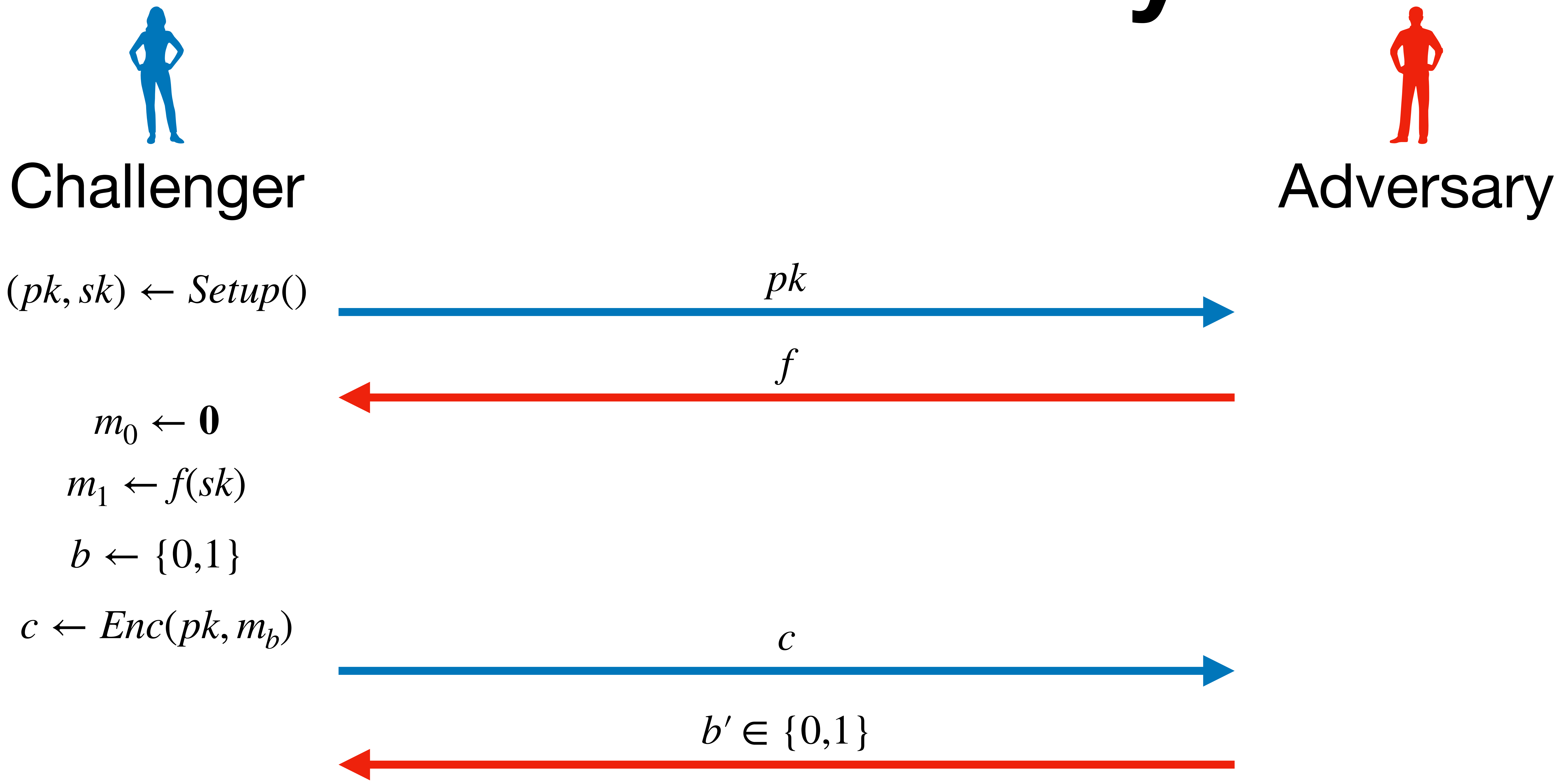
$m_1 \leftarrow f(sk)$

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

$c \leftarrow Enc(pk, m_b)$

c

KDM Security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

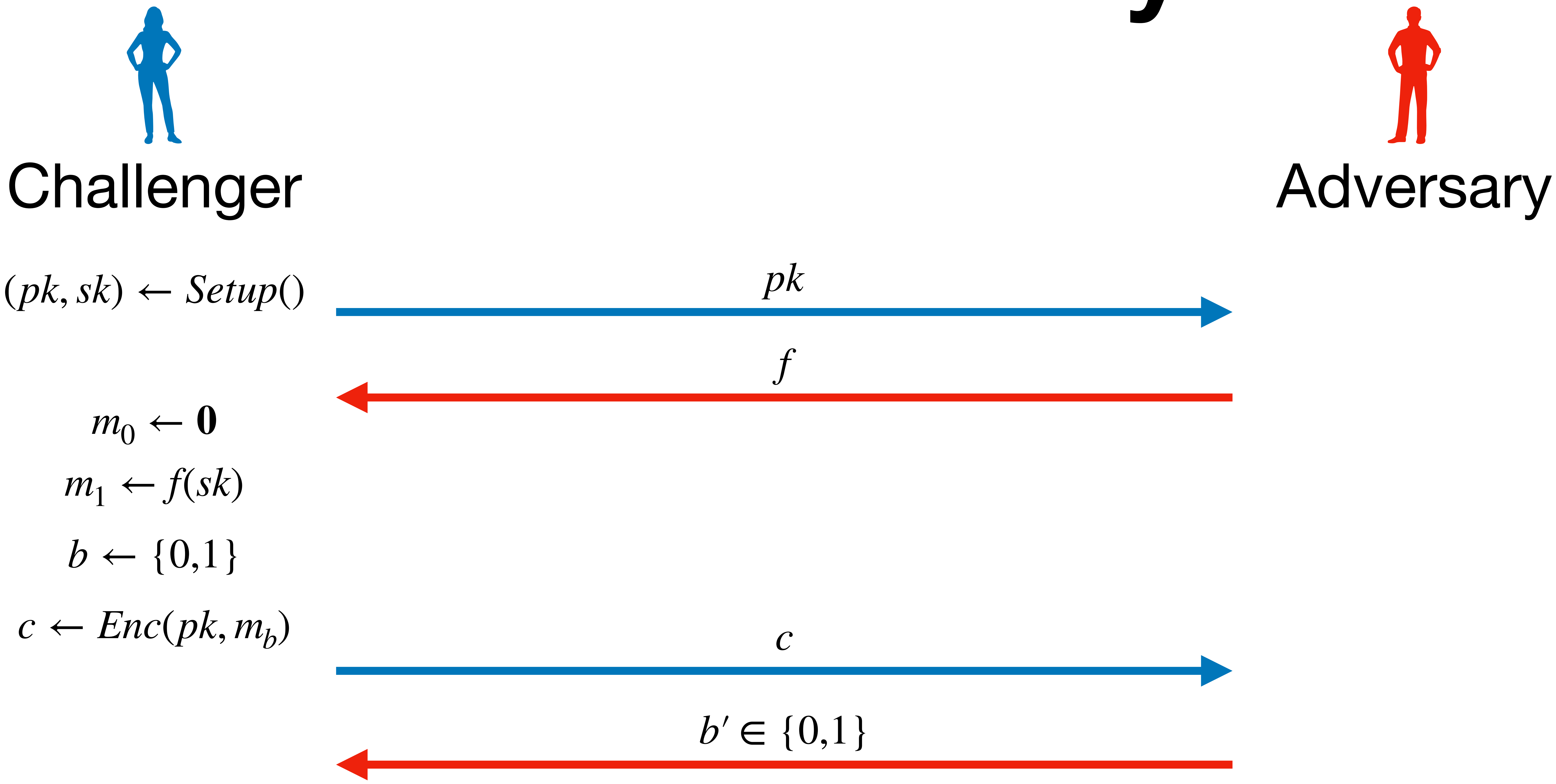
$b \leftarrow \{0,1\}$

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

KDM Security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

f

$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

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

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

Adversary wins if $b = b'$

Function Classes

Function Classes

- **Circular:** $f_i(x_1, \ldots, x_n) = x_i$.

Function Classes

- **Circular:** $f_i(x_1, \ldots, x_n) = x_i$.
- **Projection:** if each of its output bits depends on at most a single input bit.

Function Classes

- **Circular**: $f_i(x_1, \ldots, x_n) = x_i$.
- **Projection**: if each of its output bits depends on at most a single input bit.
- **Affine**: can be represented as $f(x) = Ax + b$ where A is a matrix and b is a vector.

Function Classes

- **Circular**: $f_i(x_1, \ldots, x_n) = x_i$.
- **Projection**: if each of its output bits depends on at most a single input bit.
- **Affine**: can be represented as $f(x) = Ax + b$ where A is a matrix and b is a vector.
- **Circuits** of a-priori bounded size s: described by a circuit of size s.

KDM Schemes

KDM Schemes

- Black, Rogaway,Shrimpton-03 formalised KDM security.

KDM Schemes

- Black, Rogaway,Shrimpton-03 formalised KDM security.
- Boneh, Halevi, Hamburg, Ostrovsky-08 developed the first KDM-secure PKE scheme from DDH assumption.

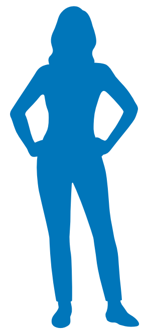
KDM Schemes

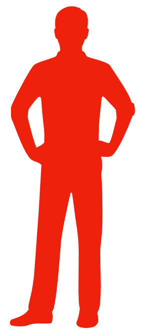
- Black, Rogaway,Shrimpton-03 formalised KDM security.
- Boneh, Halevi, Hamburg, Ostrovsky-08 developed the first KDM-secure PKE scheme from DDH assumption.
- Applebaum, Cash, Peikert, Sahai-09 gave construction for KDM-secure PKE from LWE.

Leakage-Resilient Key Dependent Message Secuity

LR-KDM security

LR-KDM security



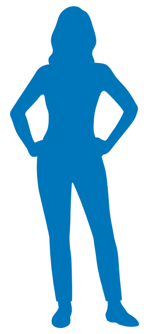


LR-KDM security

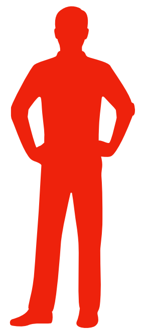
Challenger

Adversary

LR-KDM security



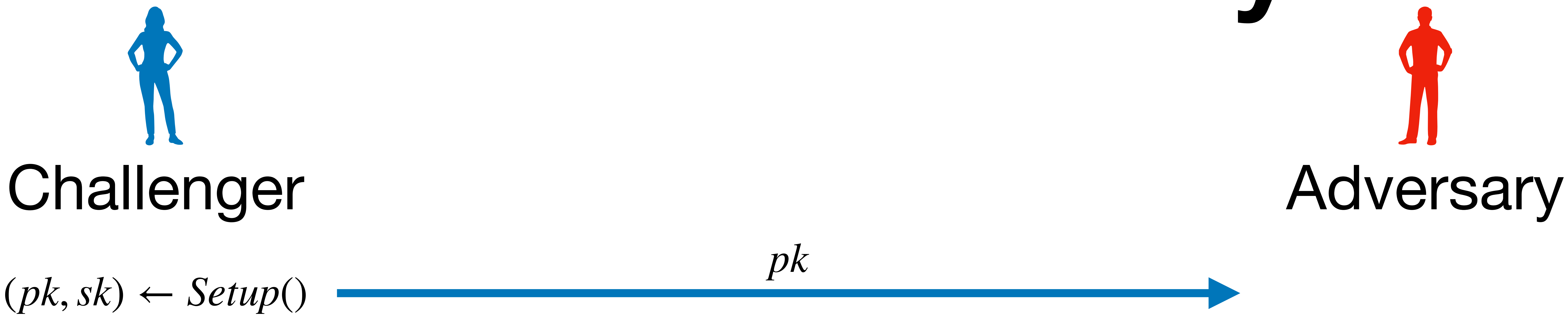
Challenger



Adversary

$(pk, sk) \leftarrow Setup()$

LR-KDM security

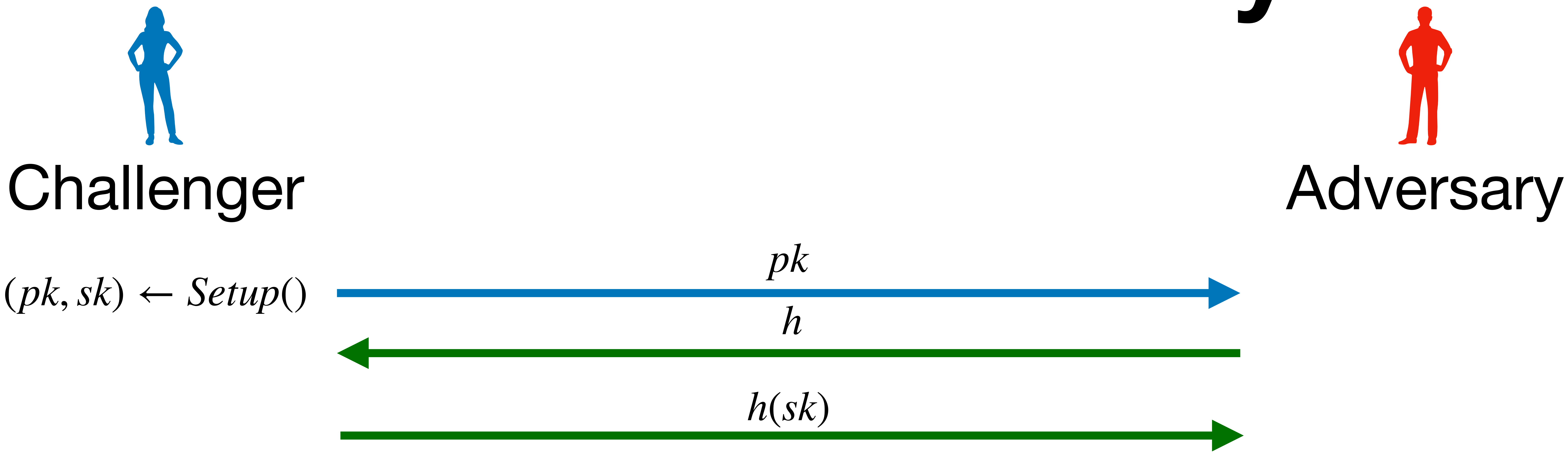


Challenger

Adversary

$(pk, sk) \leftarrow Setup()$ $\xrightarrow{pk}$

LR-KDM security



Challenger

Adversary

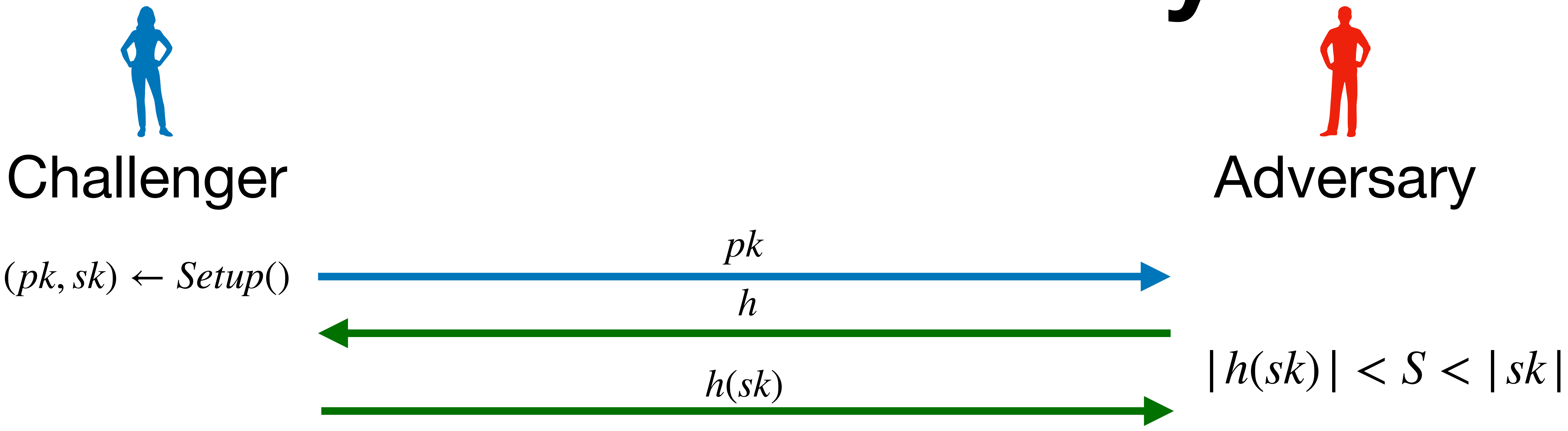
$(pk, sk) \leftarrow Setup()$

pk

h

$h(sk)$

LR-KDM security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

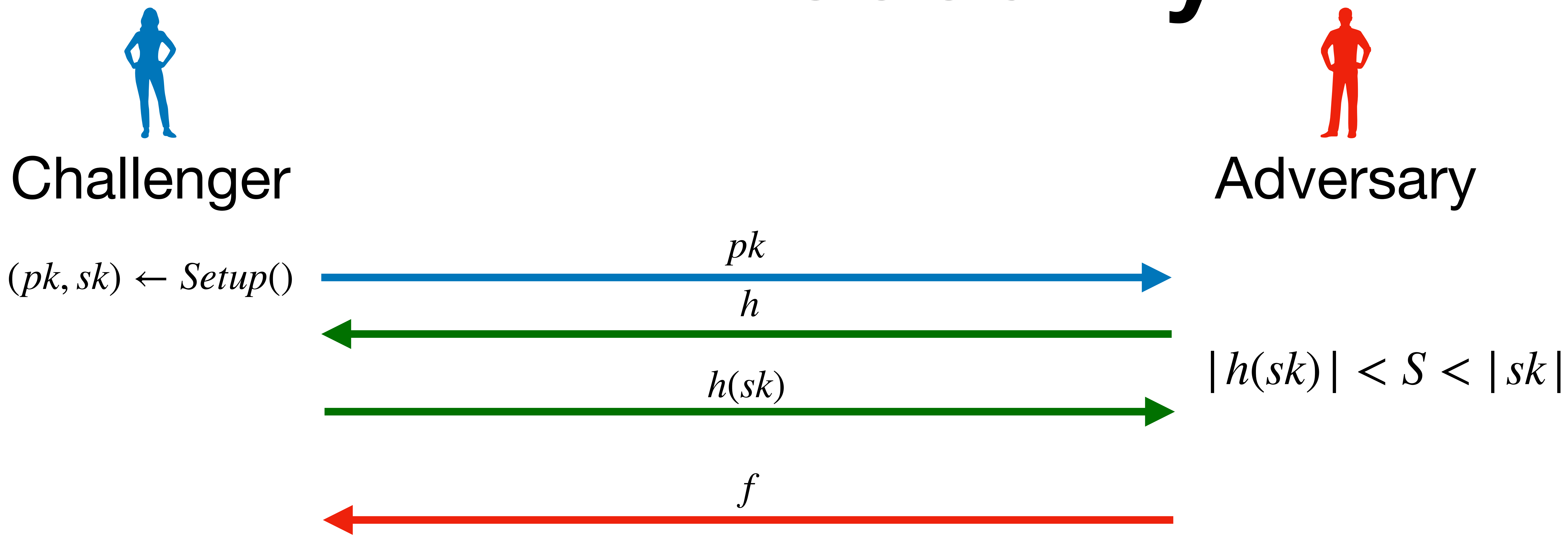
pk

h

$h(sk)$

$|h(sk)| < S < |sk|$

LR-KDM security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

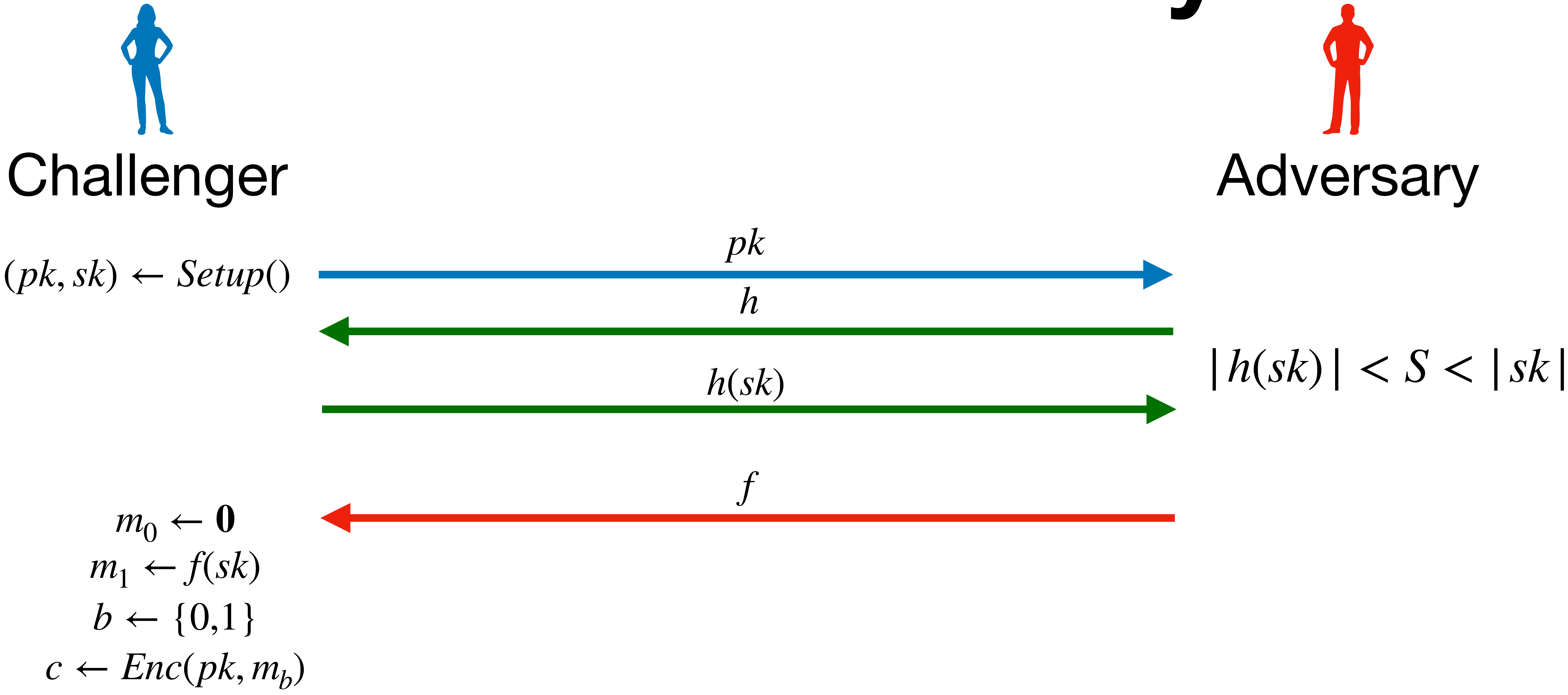
h

$h(sk)$

$|h(sk)| < S < |sk|$

f

LR-KDM security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

h

$h(sk)$

$|h(sk)| < S < |sk|$

f

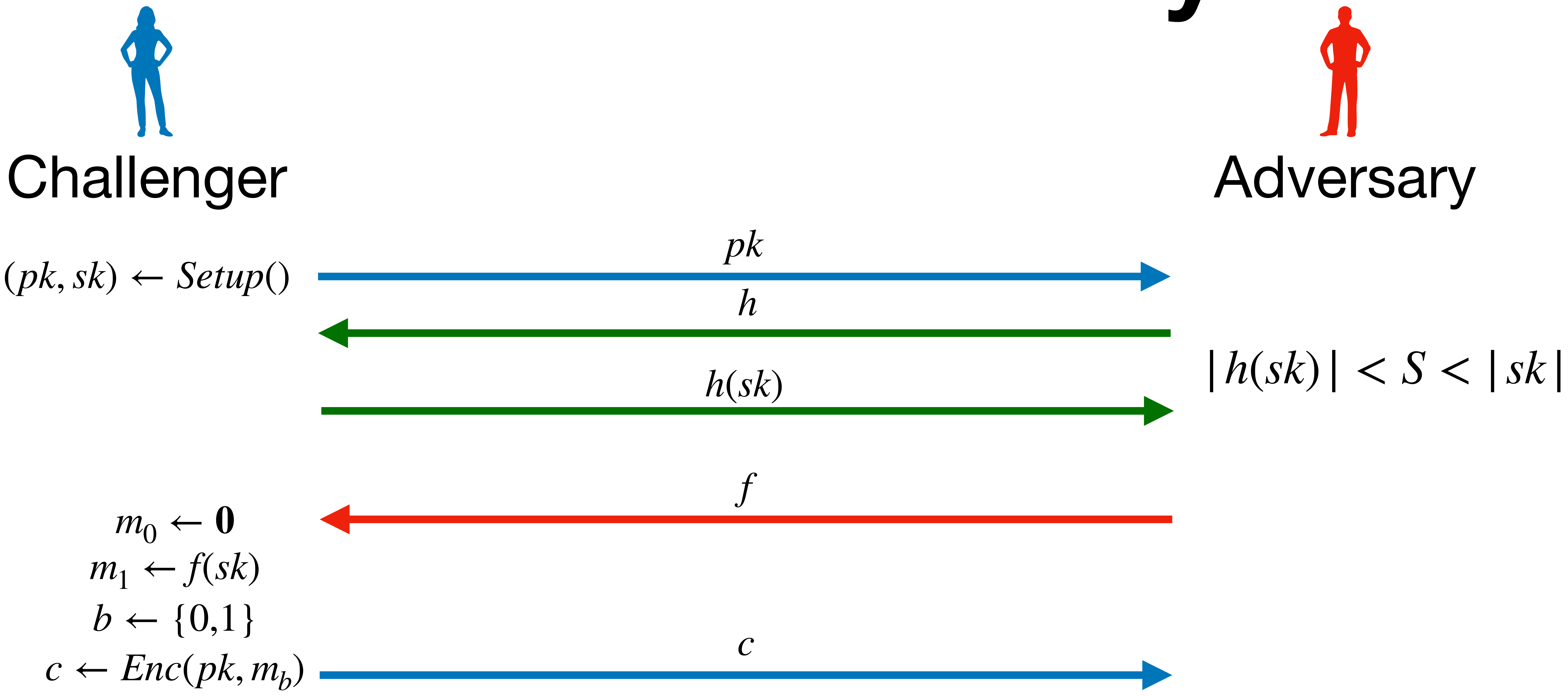
$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

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

$c \leftarrow Enc(pk, m_b)$

LR-KDM security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$ $\xrightarrow{pk}$

$\xleftarrow{h}$

$\xrightarrow{h(sk)}$ $|h(sk)| < S < |sk|$

$\xleftarrow{f}$

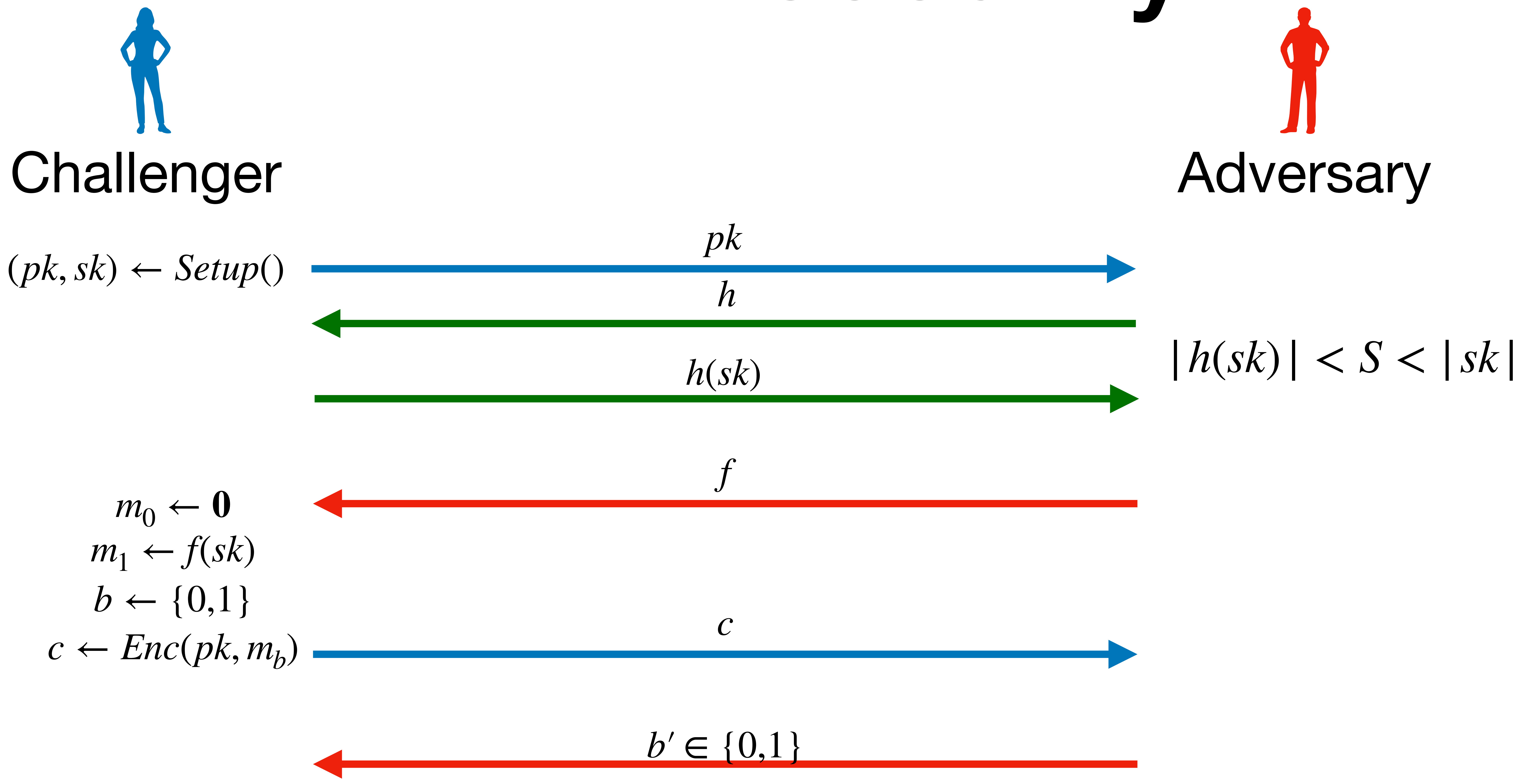
$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

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

$c \leftarrow Enc(pk, m_b)$ $\xrightarrow{c}$

LR-KDM security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

h

$h(sk)$

$|h(sk)| < S < |sk|$

f

$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

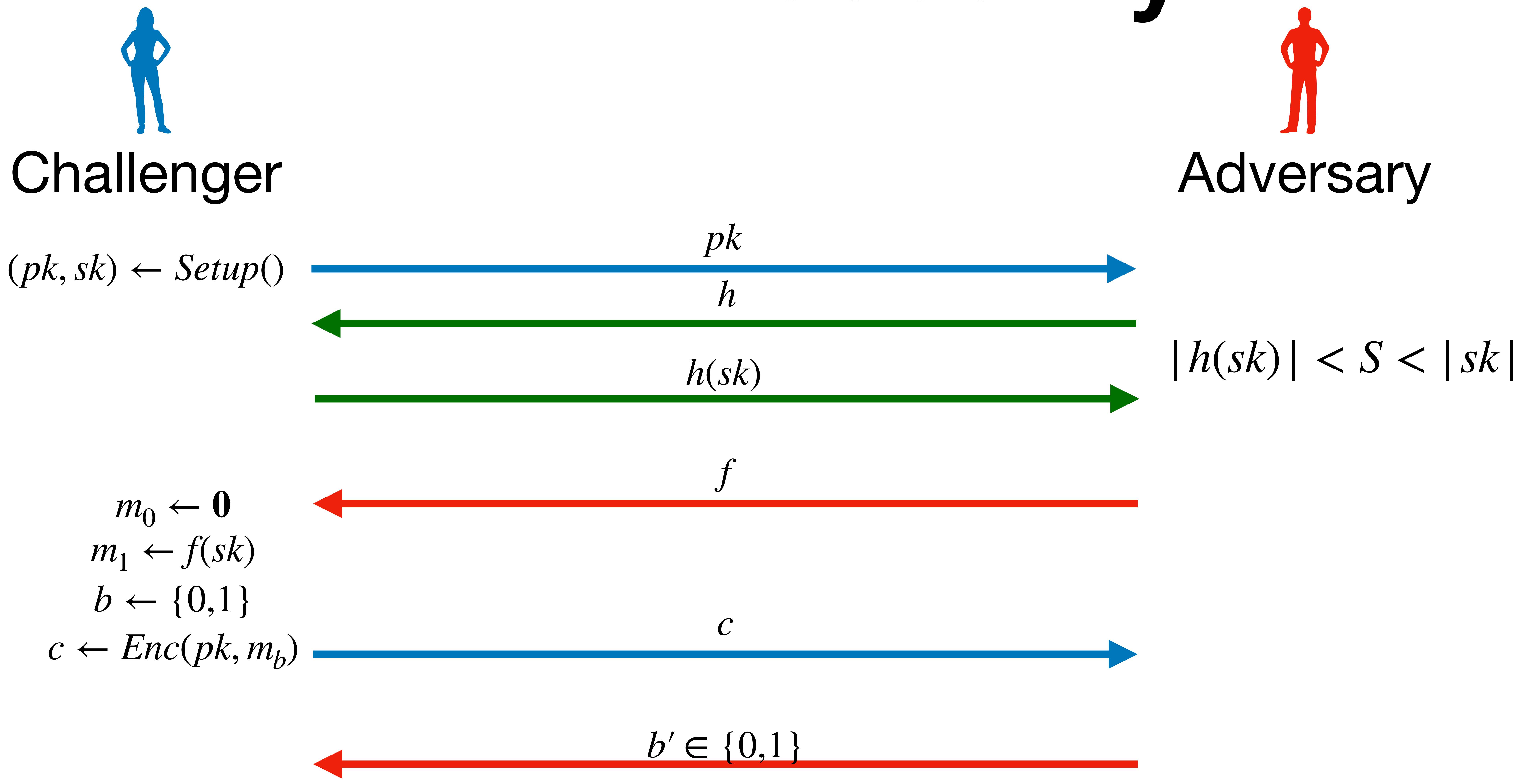
$b \leftarrow \{0,1\}$

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

LR-KDM security



Challenger

Adversary

$(pk, sk) \leftarrow Setup()$

pk

h

$h(sk)$

$|h(sk)| < S < |sk|$

f

$m_0 \leftarrow \mathbf{0}$

$m_1 \leftarrow f(sk)$

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

$c \leftarrow Enc(pk, m_b)$

c

$b' \in \{0,1\}$

Adversary wins if $b = b'$

Prior Works

Prior Works

- Naor and Segev-09 showed that BHHO construction is LR.

Prior Works

- Naor and Segev-09 showed that BHHO construction is LR.

- Brakerski and Goldwasser-10 constructed schemes that are LR and KDM scheme from QR and DCR assumptions.

Prior Works

- Naor and Segev-09 showed that BHHO construction is LR.
- Brakerski and Goldwasser-10 constructed schemes that are LR and KDM scheme from QR and DCR assumptions.
- Hajiabadi, Kapron, Srinivasan-16 developed a scheme that are LR and KDM secure schemes using homomorphic hash proof systems.

Prior Works

- Naor and Segev-09 showed that BHHO construction is LR.
- Brakerski and Goldwasser-10 constructed schemes that are LR and KDM scheme from QR and DCR assumptions.
- Hajiabadi, Kapron, Srinivasan-16 developed a scheme that are LR and KDM secure schemes using homomorphic hash proof systems.
- Brakerski, Lombardi, Segev, Vaikuntanathan-18 used batch encryption to construct scheme that are LR and KDM secure schemes based on DDH, LPN and other standard assumptions.

Prior Works

- Naor and Segev-09 showed that BHHO construction is LR.
- Brakerski and Goldwasser-10 constructed schemes that are LR and KDM scheme from QR and DCR assumptions.
- Hajiabadi, Kapron, Srinivasan-16 developed a scheme that are LR and KDM secure schemes using homomorphic hash proof systems.
- Brakerski, Lombardi, Segev, Vaikuntanathan-18 used batch encryption to construct scheme that are LR and KDM secure schemes based on DDH, LPN and other standard assumptions.
- Dodis, Karthikeyan, Wichs-21 defined CS+LR Security which is stronger than LR-KDM and used it to construct updatable PKE schemes.

Separation Result

Result

Result

There exists schemes that are LR and KDM secure,
but isn’t LR-KDM secure.

Construction

Construction

- Let SKE’ be LR and circular-KDM.

Construction

- Let SKE’ be LR and circular-KDM.
- PRF be a pseudorandom function.

Construction

- Let SKE' be LR and circular-KDM.
- PRF be a pseudorandom function.
- $Setup$: Run $ske.sk \leftarrow SKE'.Setup()$ and generate PRF key k. Output $sk = (k, ske.sk)$

Construction

- Let SKE’ be LR and circular-KDM.
- PRF be a pseudorandom function.
- $Setup$: Run $ske.sk \leftarrow SKE'.Setup()$ and generate PRF key k. Output $sk = (k, ske.sk)$
- $Enc(sk, m)$: If $m = ske.sk$, set $c_0 = PRF(k,1)$. Else, $c_0 = PRF(k,0)$. Generate $c_1 \leftarrow SKE'.Enc(ske.sk, m)$. Output $ct = (c_0, c_1)$.

Construction

- Let SKE' be LR and circular-KDM.
- PRF be a pseudorandom function.
- $Setup$: Run $ske.sk \leftarrow SKE'.Setup()$ and generate PRF key k. Output $sk = (k, ske.sk)$
- $Enc(sk, m)$: If $m = ske.sk$, set $c_0 = PRF(k,1)$. Else, $c_0 = PRF(k,0)$. Generate $c_1 \leftarrow SKE'.Enc(ske.sk, m)$. Output $ct = (c_0, c_1)$.
- $Dec(sk, ct)$: Output $SKE'.Dec(ske.sk, c_1)$.

LR and KDM security

LR and KDM security

- If adversary A breaks LR security, the LR security of SKE’ is broken.

LR and KDM security

- If adversary A breaks LR security, the LR security of SKE’ is broken.
    - Reduction B on receiving h from A, generates k and relays $h(k, \cdot\,)$ to challenger.

LR and KDM security

- If adversary A breaks LR security, the LR security of SKE' is broken.
    - Reduction B on receiving h from A, generates k and relays $h(k, \cdot)$ to challenger.
    - It generate $c_0 = PRF(k,0)$.

LR and KDM security

- If adversary A breaks LR security, the LR security of SKE' is broken.
  - Reduction B on receiving h from A, generates k and relays $h(k, \cdot)$ to challenger.
  - It generate $c_0 = PRF(k,0)$.
- If adversary A breaks f-KDM security, the KDM security of SKE' is broken.

LR and KDM security

- If adversary A breaks LR security, the LR security of SKE' is broken.
  - Reduction B on receiving h from A, generates k and relays $h(k, \cdot)$ to challenger.
  - It generate $c_0 = PRF(k,0)$.
- If adversary A breaks f-KDM security, the KDM security of SKE' is broken.
  - Here, $f(x, y) = y$.

LR and KDM security

- If adversary A breaks LR security, the LR security of SKE’ is broken.
  - Reduction B on receiving h from A, generates k and relays $h(k, \cdot\,)$ to challenger.
  - It generate $c_0 = PRF(k,0)$.
- If adversary A breaks f-KDM security, the KDM security of SKE’ is broken.
  - Here, $f(x, y) = y$.
  - B generates a random c_0.

Not LR-KDM secure

Not LR-KDM secure

- Adversary can leak the entire k in the leakage phase.

Not LR-KDM secure

- Adversary can leak the entire k in the leakage phase.

- Using k, it checks whether $c_0 = PRF(k, 0)$ or not.

Constructions and Amplifications

Constructions

Constructions

- Wee-16 showed that homomorphic HPS gives KDM secure schemes.

Constructions

- Wee-16 showed that homomorphic HPS gives KDM secure schemes.
  - We defined LR homomorphic HPS and constructed LR-KDM secure schemes.

Constructions

- Wee-16 showed that homomorphic HPS gives KDM secure schemes.
  - We defined LR homomorphic HPS and constructed LR-KDM secure schemes.
- We showed that batch encryption schemes are also LR-KDM secure.

Amplifications

Amplifications

- Waters and Wichs-23 showed that PKE + (existence) circular-KDM SKE gives circuit-KDM PKE.

Amplifications

- Waters and Wichs-23 showed that PKE + (existence) circular-KDM SKE gives circuit-KDM PKE.

- Applebaum-14 showed projection-KDM PKE + garbled circuits implies circuit-KDM PKE.

Amplifications

- Waters and Wichs-23 showed that PKE + (existence) circular-KDM SKE gives circuit-KDM PKE.

- Applebaum-14 showed projection-KDM PKE + garbled circuits implies circuit-KDM PKE.

- We showed these can be used in the LR-KDM setting.

Future Works

Future Works

- Multi-Key LR-KDM security where adversary interacts with multiple pairs of public-secret keys.

Future Works

- Multi-Key LR-KDM security where adversary interacts with multiple pairs of public-secret keys.

- LR-KDM security under Chosen-Ciphertext Attacks.

Future Works

- Multi-Key LR-KDM security where adversary interacts with multiple pairs of public-secret keys.

- LR-KDM security under Chosen-Ciphertext Attacks.

- LR-KDM in advanced primitives such as IBE and ABE.

Thank You!