EE309 Advanced Programming Techniques for EE

Lecture 19: PRF, PRP, Hash, and PRNG

INSU YUN (윤인수)

School of Electrical Engineering, KAIST

[Slides from Introduction to Computer Security (18-487) at CMU]

The Landscape

Jargon in Cryptography

Good News: OTP has perfect secrecy

Thm: The One Time Pad is Perfectly Secure

Must show: $\Pr\left[E(k,m_0)=c\right]=\Pr\left[E(k,m_1)=c\right]$

where $|M| = \{0,1\}^m$

Proof:

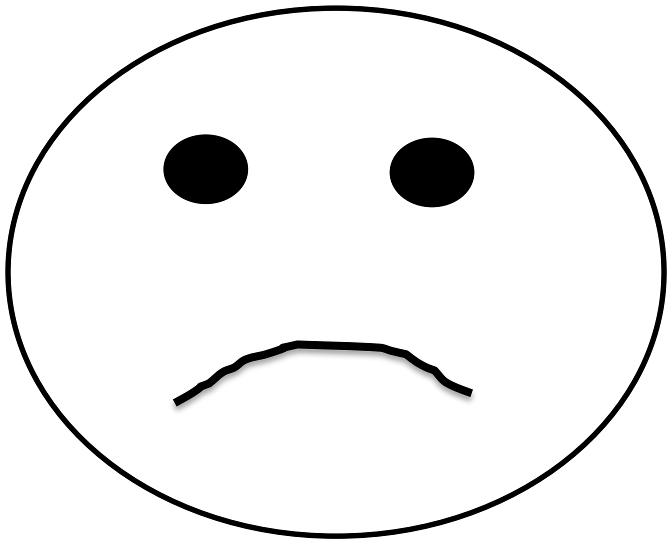
$$
\begin{aligned}
\Pr[E(k,m_0)=c] &= \Pr[k \oplus m_0 = c] && (1)\\
&= \frac{|k \in \{0,1\}^m : k \oplus m_0 = c|}{\{0,1\}^m} && (2)\\
&= \frac{1}{2^m} && (3)\\
\Pr[E(k,m_1)=c] &= \Pr[k \oplus m_1 = c] && (4)\\
&= \frac{|k \in \{0,1\}^m : k \oplus m_1 = c|}{\{0,1\}^m} && (5)\\
&= \frac{1}{2^m} && (6)
\end{aligned}
$$

Information-Theoretic Secrecy

Therefore, $\Pr[E(k,m_0)=c]=\Pr[E(k,m_1)=c]$

The “Bad News” Theorem

<u>Theorem</u>: Perfect secrecy requires |K| >= |M|



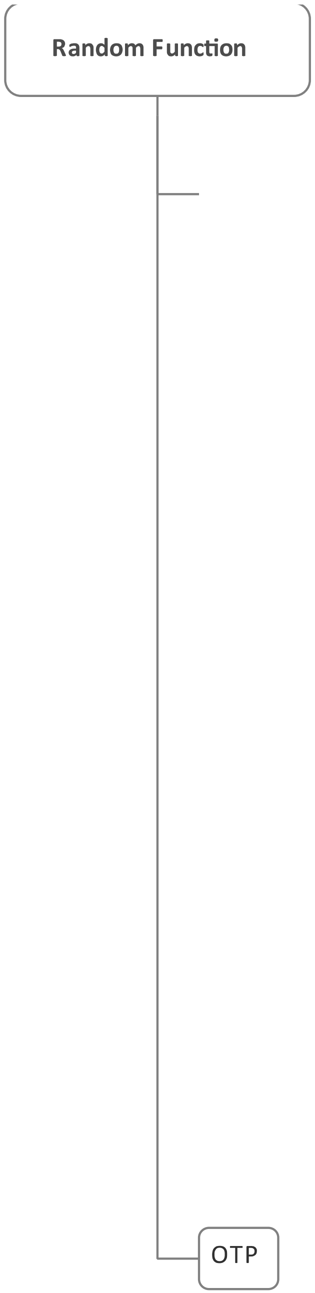
Kerckhoffs' Principle

The system must be *practically*, if not mathematically, indecipherable

- Security is only preserved against efficient adversaries running in (probabilistic) polynomial time (PPT) and space
- Adversaries can succeed with some small probability (that is small enough it is hopefully not a concern)
  - Ex: Probability of guessing a password

"A scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability"

The Landscape



Pseudorandom Number Generators

Amplify small amount of randomness to large "pseudo-random" number with a *pseudo-random number generator* (PRNG)

$$\text{Let } S : \{0,1\}^s \text{ and } K : \{0,1\}^k$$
$$G : S \rightarrow K \text{ where } k \gg s$$

One Way Functions

<u>Defn</u>: A function *f* is one-way if:

1. *f* can be computed in polynomial time
2. No polynomial time adversary ***A*** can invert with more than negligible probability

$$\Pr[f(\mathbf{A}(f(x))) = f(x)] < \epsilon$$

<u>Note:</u> mathematically, a function is one-way if it is not one-to-one. Here we mean something stronger.

Candidate One-Way Functions

- Factorization. Let $N=p*q$, where $|p| = |q| = |N|/2$. We believe factoring N is hard.

- Discrete Log. Let p be a prime, x be a number between 0 and p. Given g^x mod p, it is believed hard to recover x.

The relationship

$$\text{PRNG exist} \Leftrightarrow \text{OWF exist}$$

Thinking About Functions

A function is just a mapping from inputs to outputs:

f_1

x	$f_1(x)$
1	4
2	13
3	12
4	1
5	7

f_2

x	$f_2(x)$
1	1
2	2
3	3
4	4
5	5

f_3

x	$f_2(x)$
1	12
2	3
3	7
4	8
5	10

...

~~Which function is *not* random?~~

Thinking About Functions

A function is just a mapping from inputs to outputs:

f_1

x	$f_1(x)$
1	4
2	13
3	12
4	1
5	7

f_2

x	$f_2(x)$
1	1
2	2
3	3
4	4
5	5

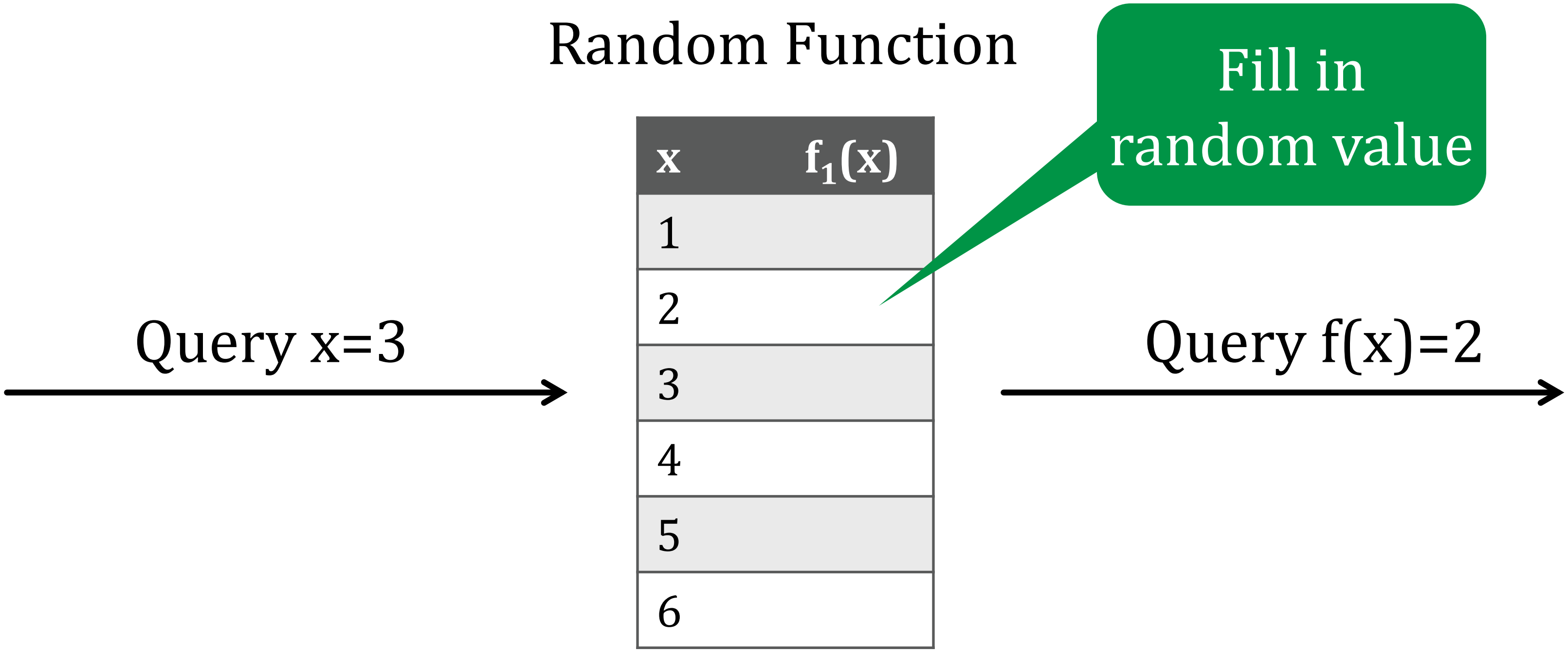
f_3

x	$f_2(x)$
1	12
2	3
3	7
4	8
5	10

...

What is random is the way we *pick* a function

Game-based Interpretation



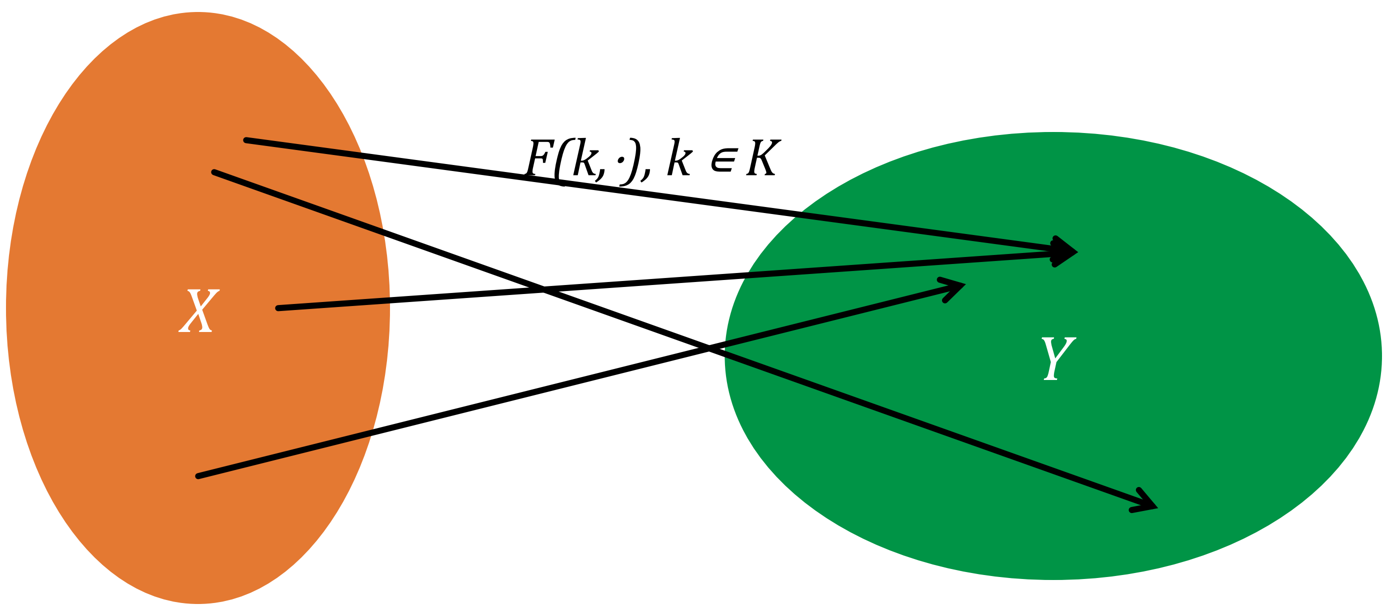
Note asking x=1, 2, 3, ... gives us our OTP randomness.

PRFs

Pseudo Random Function (PRF) defined over (K,X,Y):

$$F : K \times X \to Y$$

such that there exists an "efficient" algorithm to evaluate $F(k,x)$



Pseudorandom functions are not to be confused with pseudorandom generators (PRGs). The guarantee of a PRG is that a single output appears random if the input was chosen at random. On the other hand, the guarantee of a PRF is that all its outputs appear random, regardless of how the corresponding inputs were chosen, as long as the function was drawn at random from the PRF family.

- wikipedia

PRNG exist $\Leftrightarrow$ OWF exist $\Leftrightarrow$ PRF exists

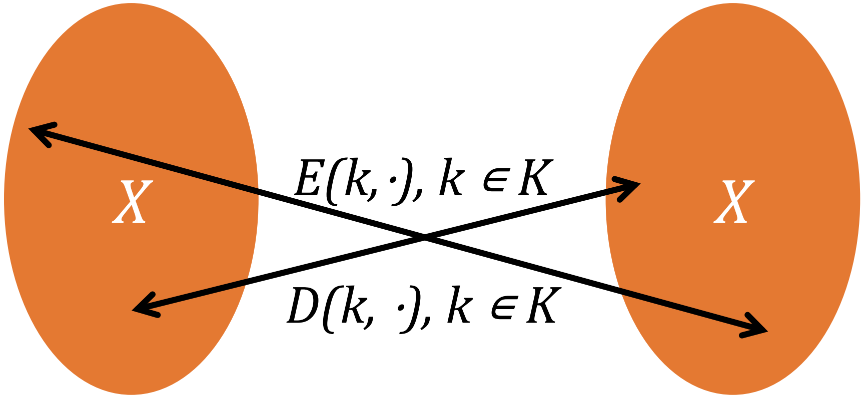
Abstractly: PRPs

Pseudo Random <u>Permutation</u> (PRP) defined over (K,X)

$$E : K \times X \rightarrow X$$

such that:

1. Exists "efficient" deterministic algorithm to evaluate *E(k,x)*
2. The function *E(k, ·)* is one-to-one
3. Exists "efficient" inversion algorithm *D(k,y)*



Running example

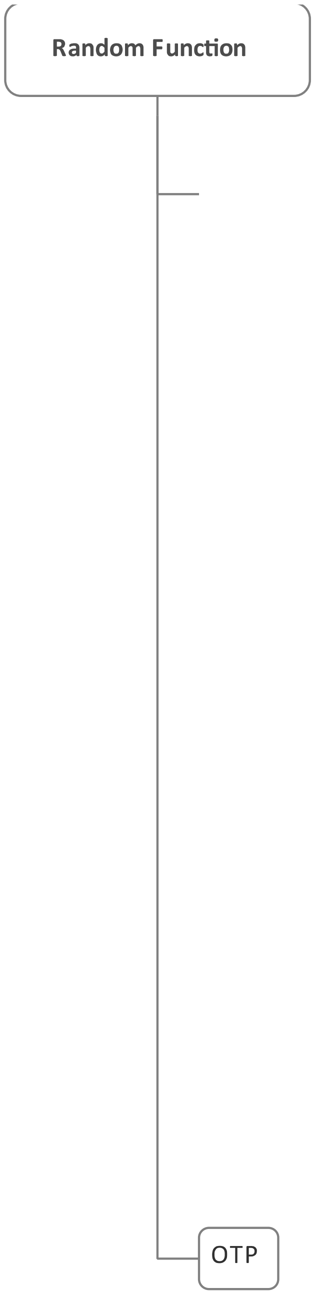
- Example PRPs: 3DES, AES, …

$$\text{AES: } K \times X \to X \quad \text{where} \quad K = X = \{0,1\}^{128}$$

$$\text{3DES: } K \times X \to X \quad \text{where} \quad X = \{0,1\}^{64}, K = \{0,1\}^{168}$$

- Functionally, *any* PRP is also a PRF.
  - PRP is a PRF when $X = Y$ and is efficiently invertible

The Landscape



Security and Indistinguishability

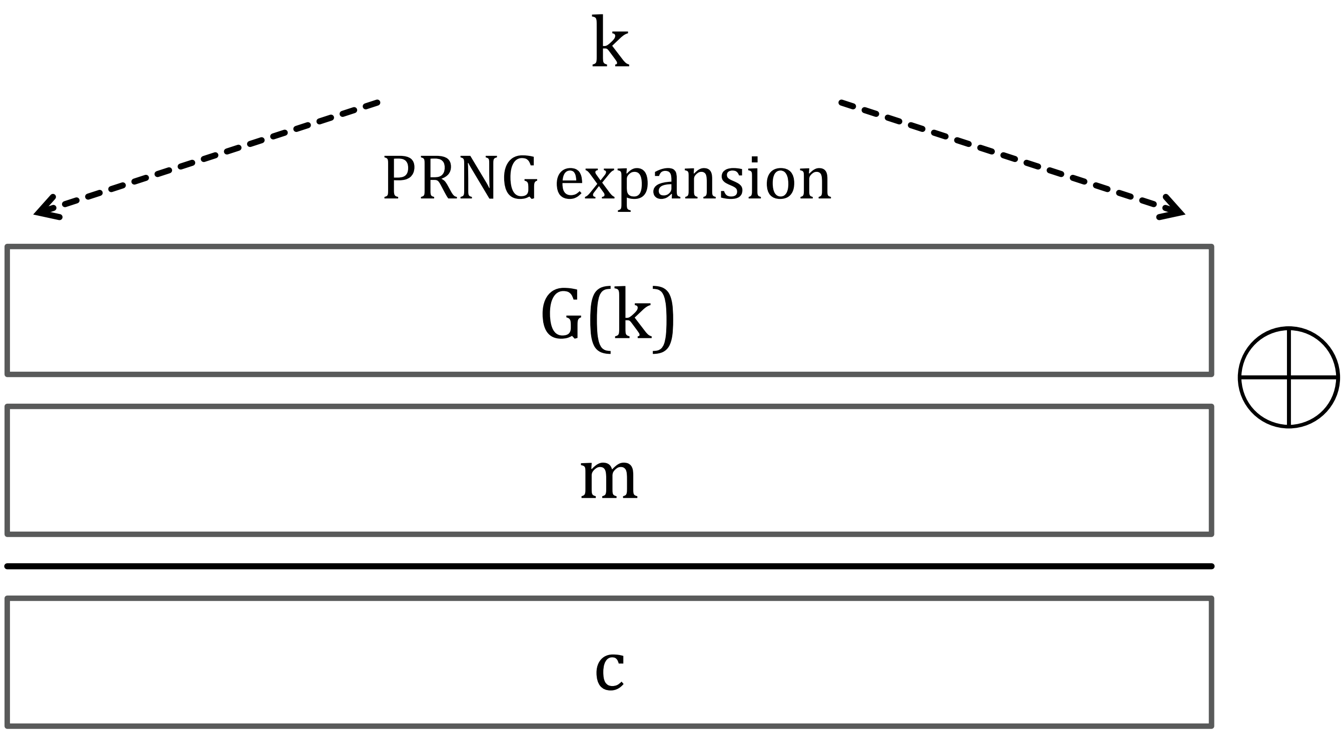
Kerckhoffs' Principle

The system must be ***practically***, if not mathematically, indecipherable

- Security is only preserved against efficient adversaries running in polynomial time and space
- Adversaries can succeed with some small probability (that is small enough it is hopefully not a concern)
  - Ex: Probability of guessing a password

"A scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability"

A Practical OTP



$$c = E(k, m) = m \oplus G(k)$$

$$D(k, c) = c \oplus G(k)$$

Question

Can a PRNG-based pad have perfect secrecy?

1. Yes, if the PRNG is secure
2. No, there are no ciphers with perfect secrecy
3. No, the key size is shorter than the message

PRG Security

Recall PRNG:

Let $S : \{0,1\}^s$ and $K : \{0,1\}^k$

$G : S \to K$ where $k \gg s$

One requirement: Output of PRG is unpredictable (mimics a perfect source of randomness)

It should be impossible for any Alg to predict bit i+1 given the first i bits:

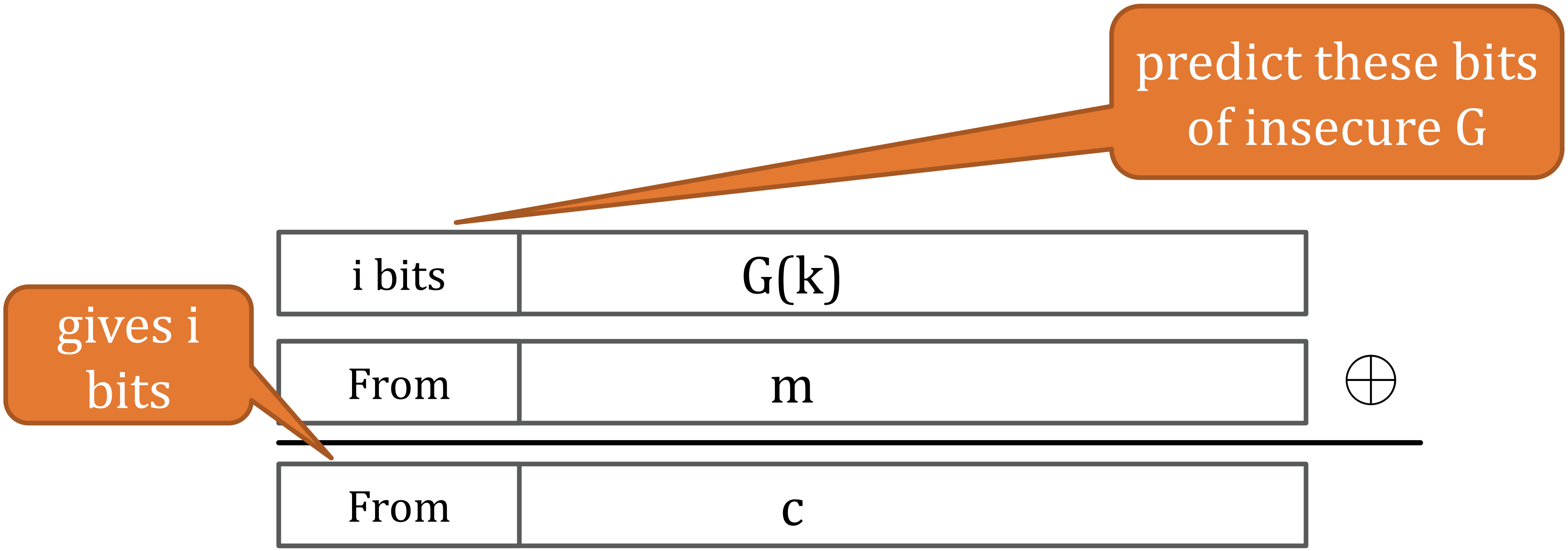
$$\exists i. G(k)|_{1,\ldots,i} \xrightarrow{\text{Alg}} G(k)|_{i+1,\ldots,n}$$

Even predicting 1 bit is insecure

Example

Suppose PRG is predictable:

$$\exists i. G(k)|_{1,\ldots,i} \xrightarrow{\text{Alg}} G(k)|_{i+1,\ldots,n}$$



Adversarial Indistinguishability Game

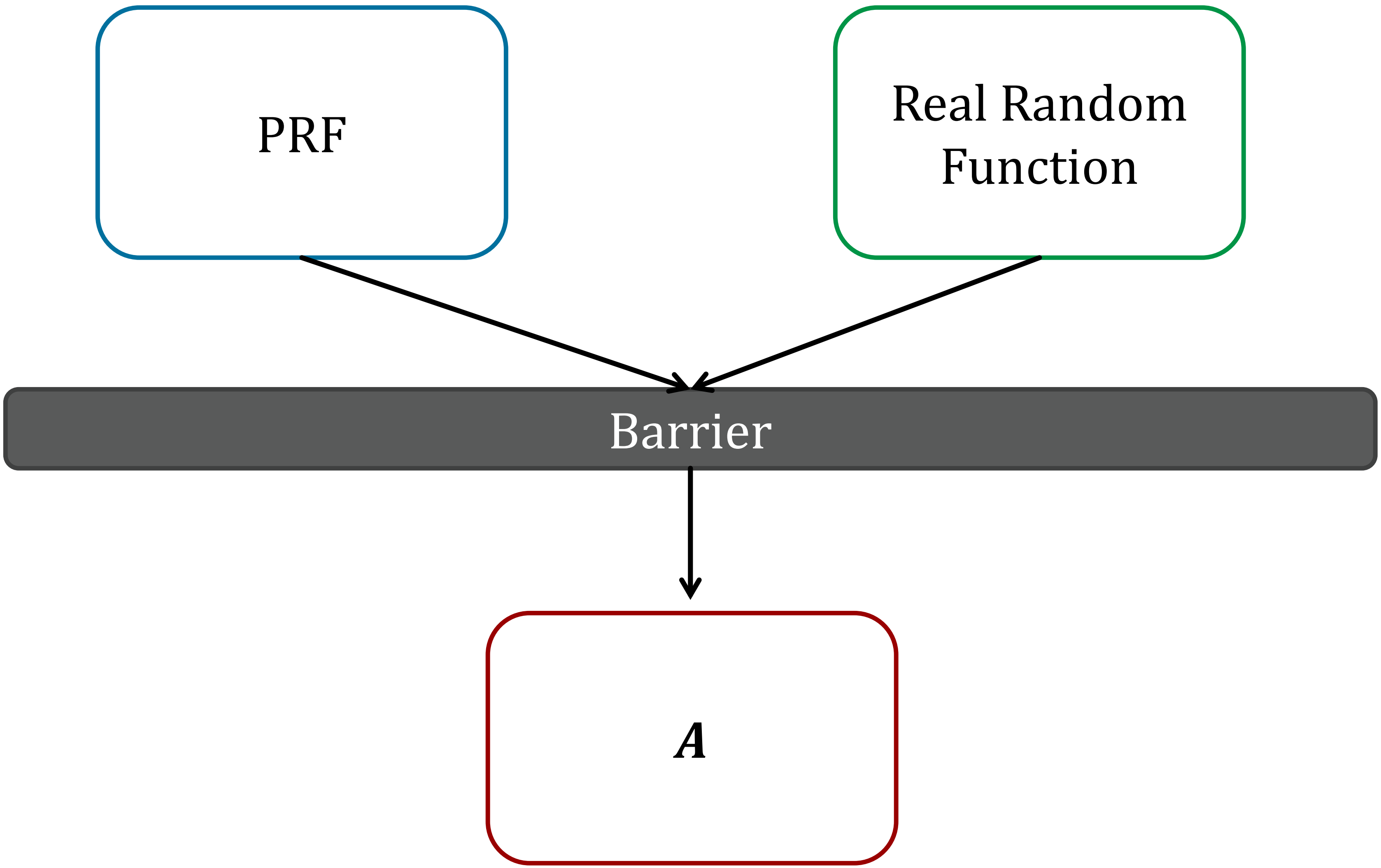
E

Challenger:
I have a secure PRF. It's just like real randomness!

A

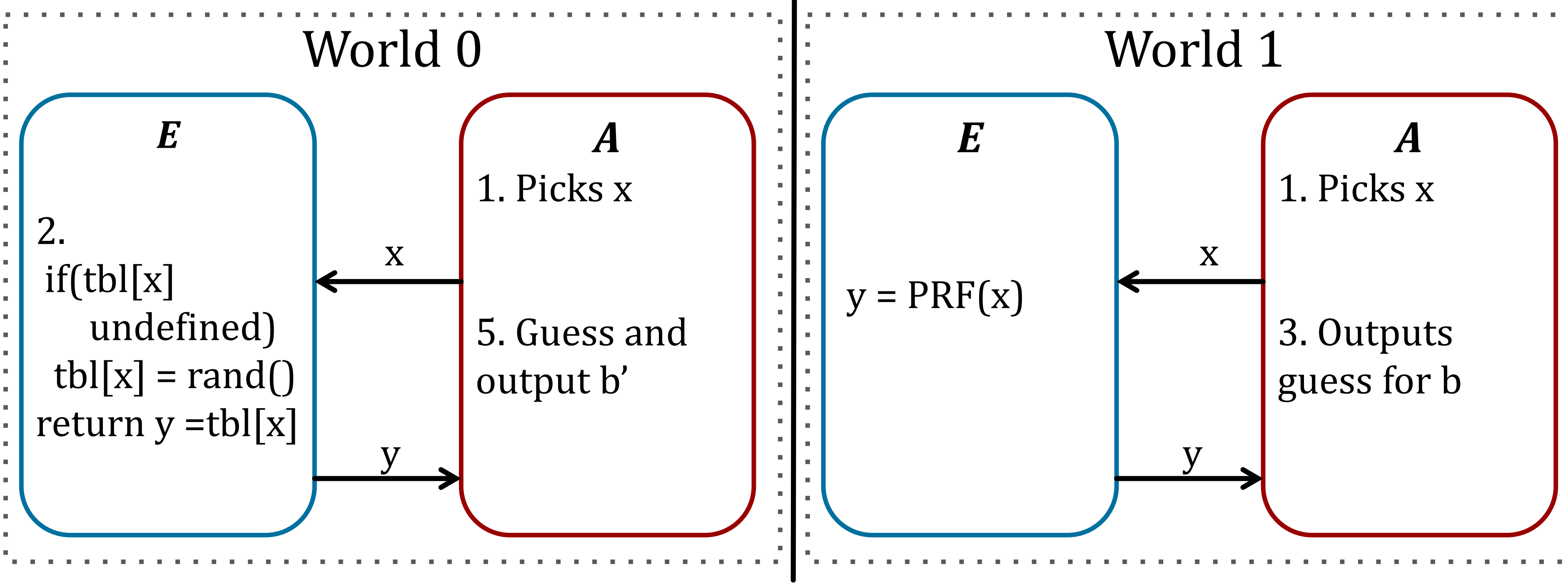
I am *<u>any</u>* adversary. You can't fool me.

Secure PRF: The Intuition



PRF Security Game

(A *behavioral* model)



A doesn't know which world he is in, but wants to figure it out.

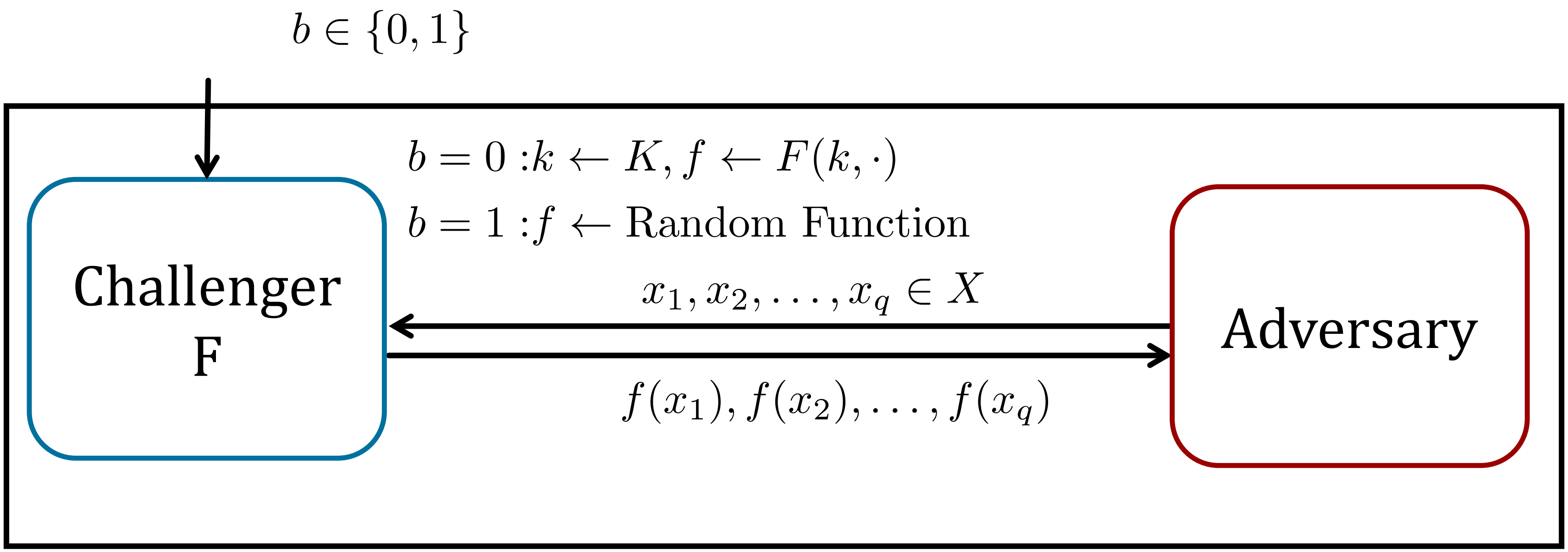
For b=0,1: W_b := [event that $\boldsymbol{A}(W_b)$ =1]

Always 1

$\mathrm{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}] := |\ \Pr[W_0] - \Pr[W_1]\ | \in [0,1]$

Secure PRF: An Alternate Interpretation

For *b = 0,1* define experiment *EXP(b)* as:



Def: PRF is a secure PRF if for all efficient ***A***:

$$\mathbf{Adv}_{PRF}[A, F] := |\Pr[Exp(0) = 1] - \Pr[Exp(1) = 1]| < \epsilon$$

Quiz

Let $F : K \times X \to \{0,1\}^{128}$ be a secure PRF.

Is the following G a secure PRF?

$$G(k,x) = \begin{cases} 0^{128} & \text{if } x = 0 \\ F(k,x) & \text{otherwise} \end{cases}$$

- ● No, it is easy to distinguish G from a random function
- ○ Yes, an attack on G would also break F
- ○ It depends on F

Semantic Security of Ciphers

What is a secure cipher?

Attackers goal: recover one plaintext (for now)

Attempt #1: Attacker cannot recover key

Insufficient: $E(k,m) = m$

Attempt #2: Attacker cannot recover all of plaintext

Insufficient: $E(k,m_0 \,||\, m_1) = m_0 \,||\, E(k,m_1)$

Recall Shannon's Intuition:
c should reveal no information about *m*

Adversarial Indistinguishability Game

E

Challenger:
I have a
secure cipher E

A

I am *any*
adversary. I can
break your crypto.

Semantic Security Motivation

E ***A***

m_0, m_1 ←

1. ***A*** sends m_0, m_1 s.t. $|m_0|=|m_1|$ to the challenger

2. Challenger encrypts one at random. Sends back c.

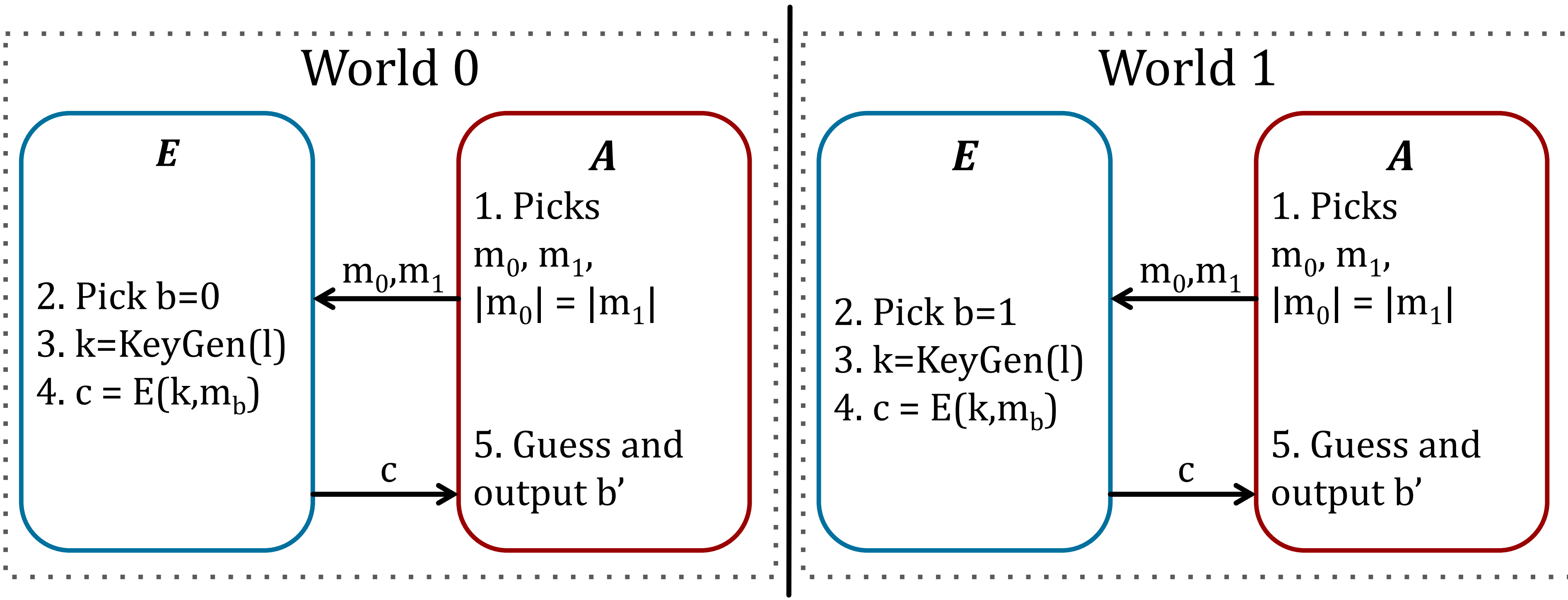
c →

3. ***A*** tries to guess which message was encrypted.

4. Challenger wins of ***A*** is no better than guessing

<u>Semantically secure</u>

Semantic Security Game

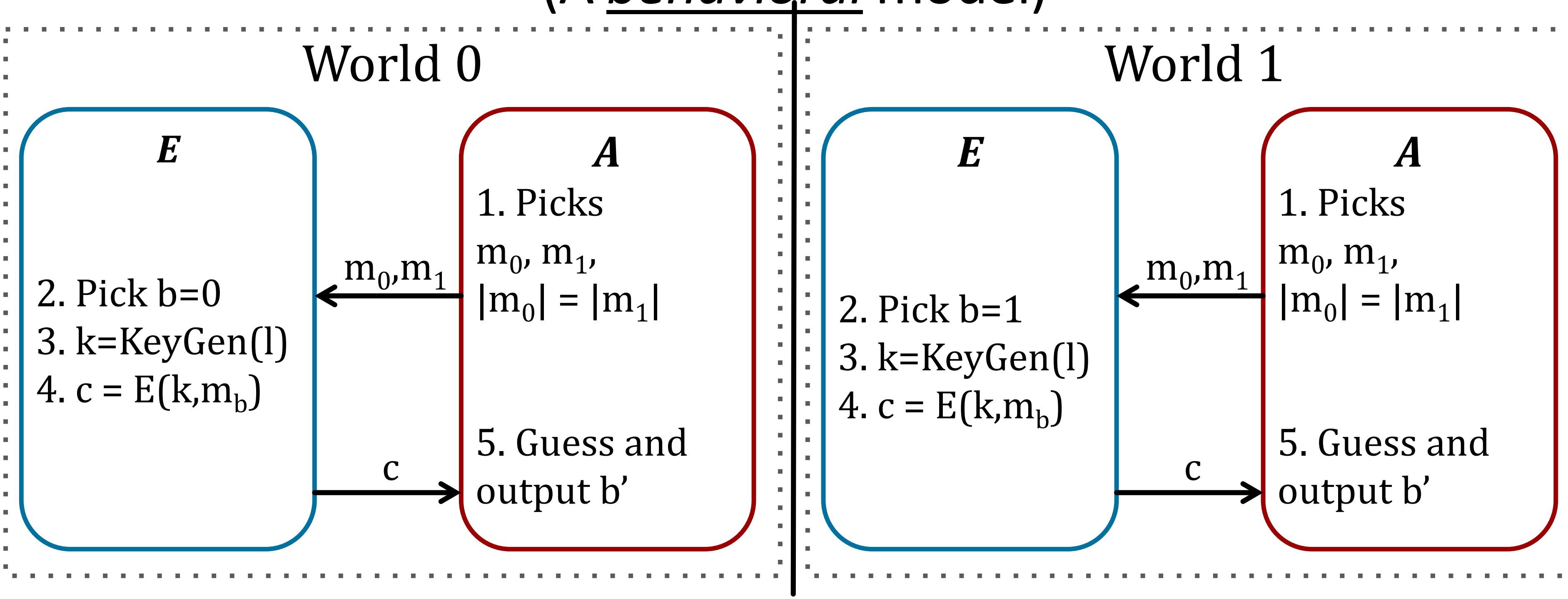


A doesn't know which world he is in, but wants to figure it out.

Semantic security is a behavioral model getting at any ***A*** behaving the same in either world when ***E*** is secure.

Semantic Security Game

(A ***behavioral*** model)



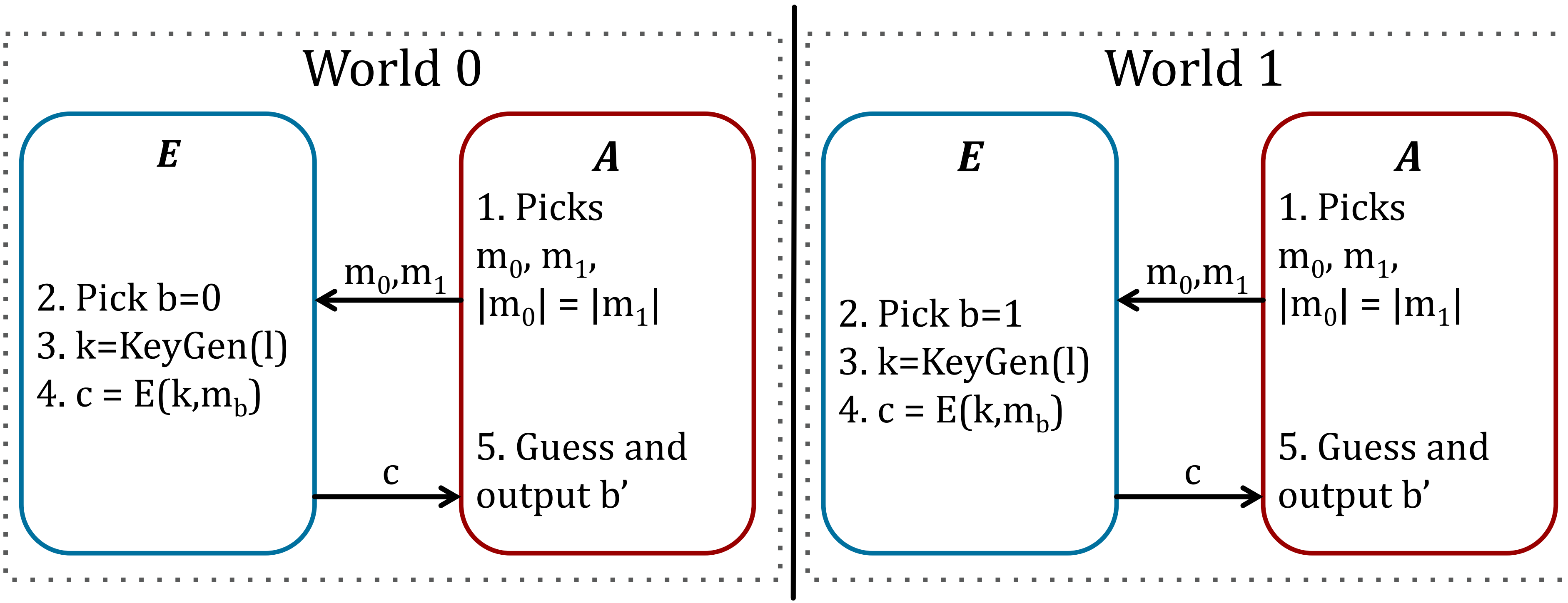
A doesn’t know which world he is in, but wants to figure it out.

For b=0,1: W_b := [event that $\boldsymbol{A}(W_b)$ =1]

Always 1

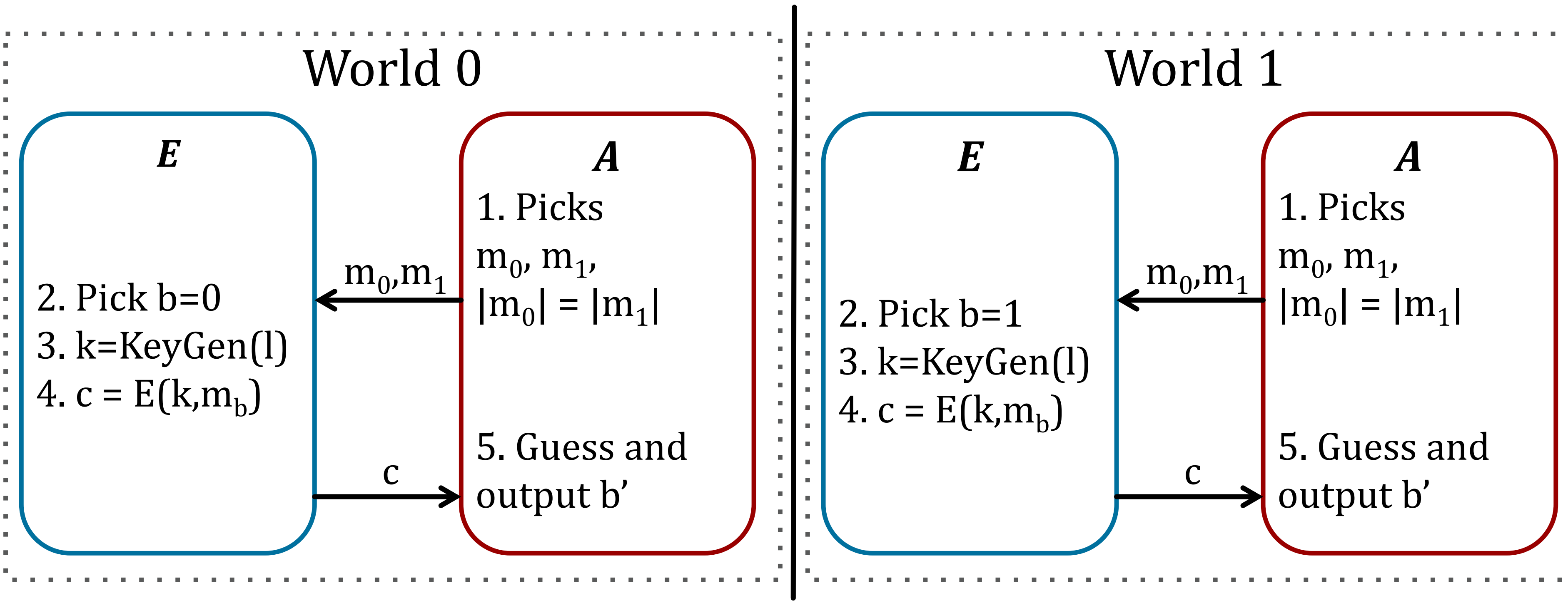
$\mathrm{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}] := |\Pr[W_0] - \Pr[W_1]| \in [0,1]$

Example 1: Guessing



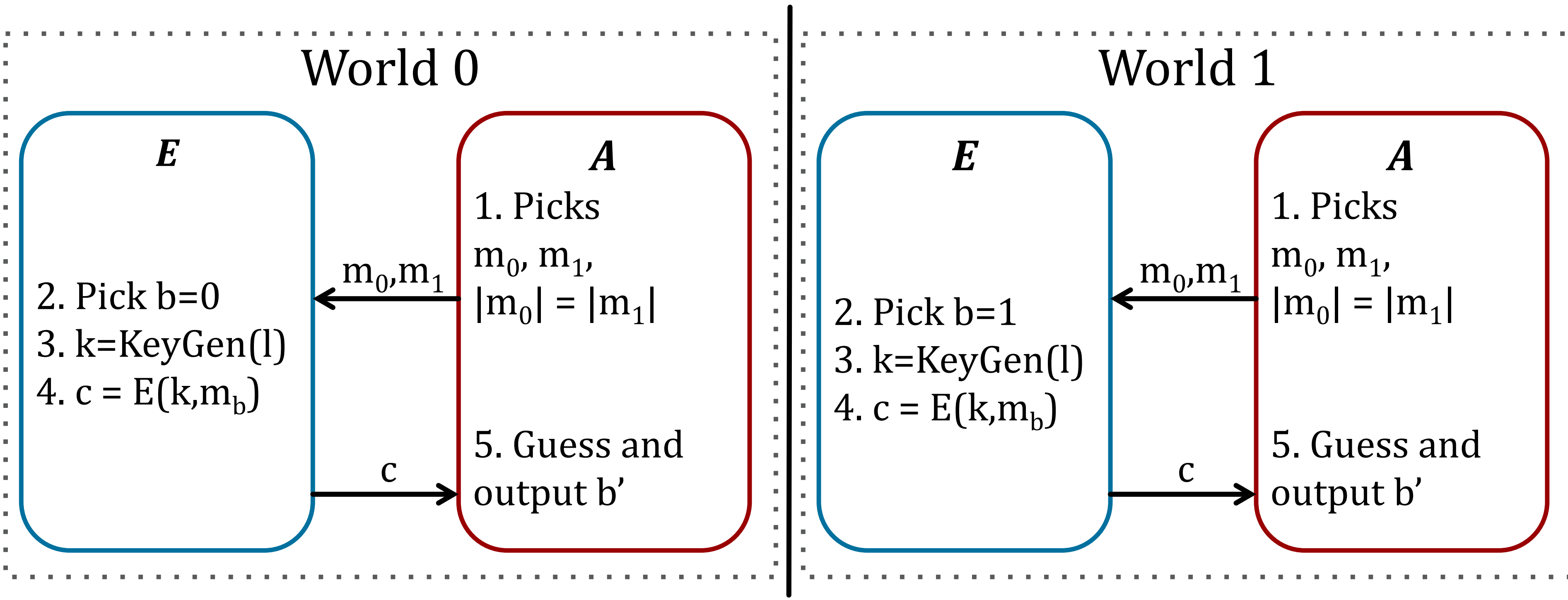
A guesses. W_b := [event that ***A***(W_b) =1]. So
W_0 = .5, and W_1 = .5
$\text{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}]$:= | .5 – .5 | = 0

Example 1: *A* is right 75% of time



A guesses. W_b := [event that ***A***(W_b) =1]. So
W_0 = .25, and W_1 = .75
$\mathrm{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}]$:= | .25 – .75 | = .5

Example 1: ***A*** is right 25% of time



A guesses. W_b := [event that ***A***(W_b) =1]. So
W_0 = .75, and W_1 = .25
$\mathrm{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}]$:= | .75 – .25 | = .5

Note for W_0, ***A*** is wrong more often than right. ***A*** should switch guesses.

Semantic Security

Given:

For b=0,1: W_b := [event that $\boldsymbol{A}(W_b)$ =1]

$\mathrm{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}]$:= | Pr[W_0] − Pr[W_1] | $\in$ [0,1]

Defn:

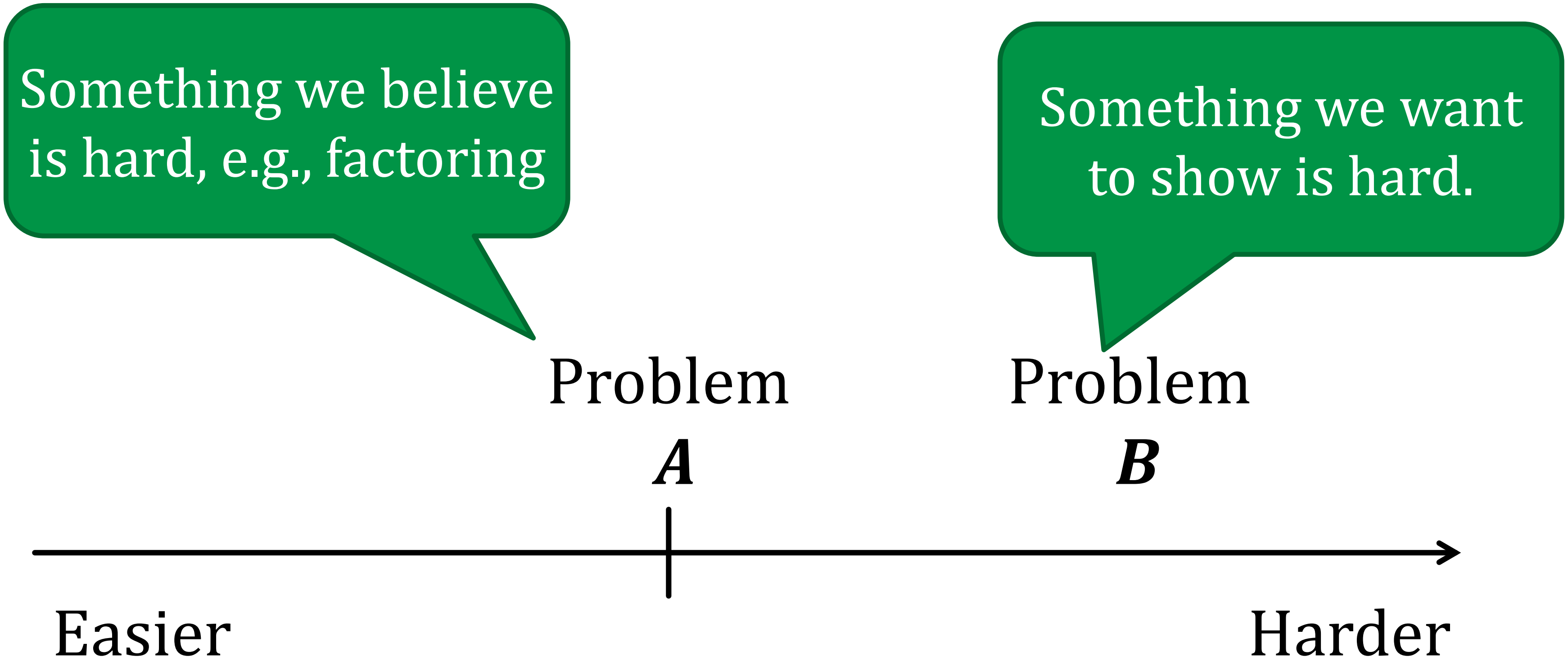
E is *semantically secure* if for all efficient ***A***:

$\mathrm{Adv}_{SS}[\boldsymbol{A}, \boldsymbol{E}]$ is negligible.

$\Rightarrow$ for all explicit m_0 , $m_1 \in M$:

$\{ E(k,m_0) \} \approx_p \{ E(k,m_1) \}$

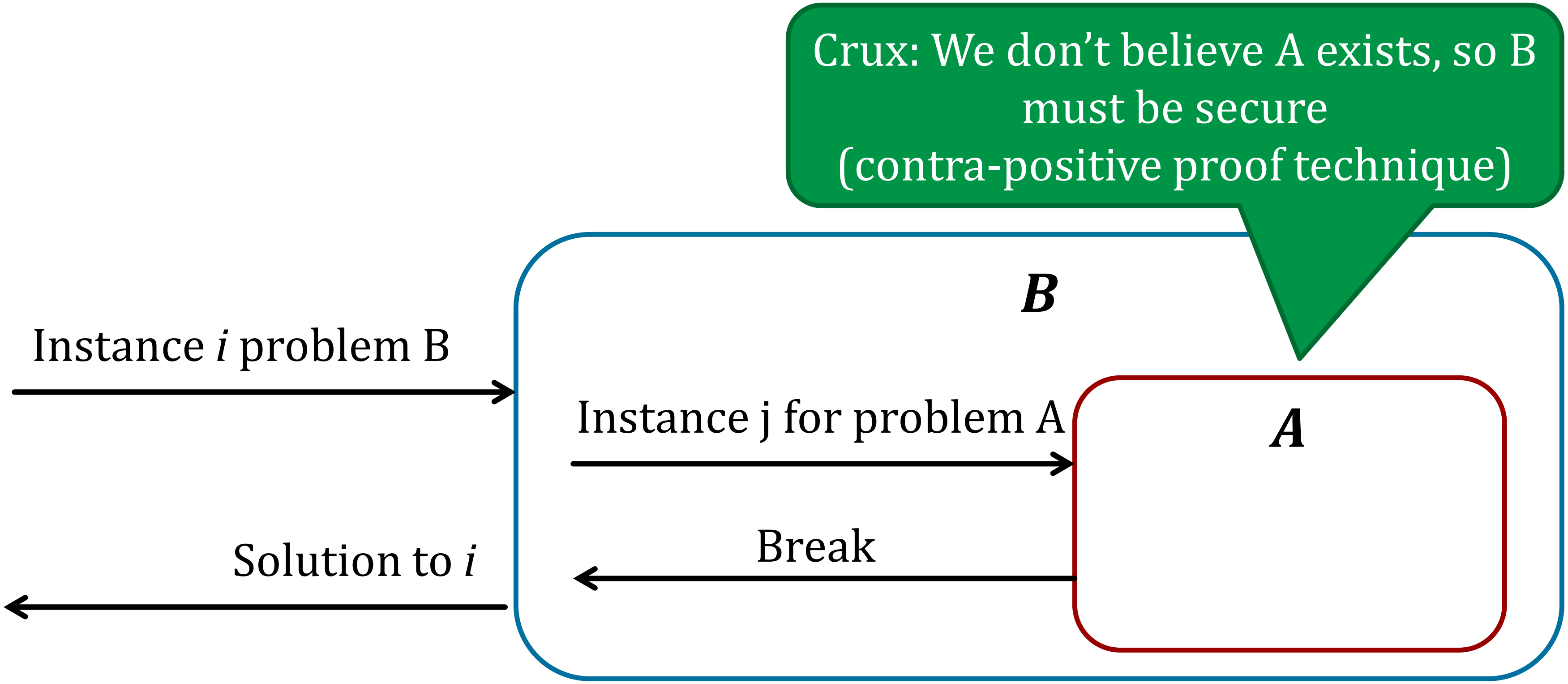
This is what it means to be secure against eavesdroppers. No partial information is leaked


Something we believe is hard, e.g., factoring
Something we want to show is hard.
Problem A
Problem B
Easier
Harder

Proving Security

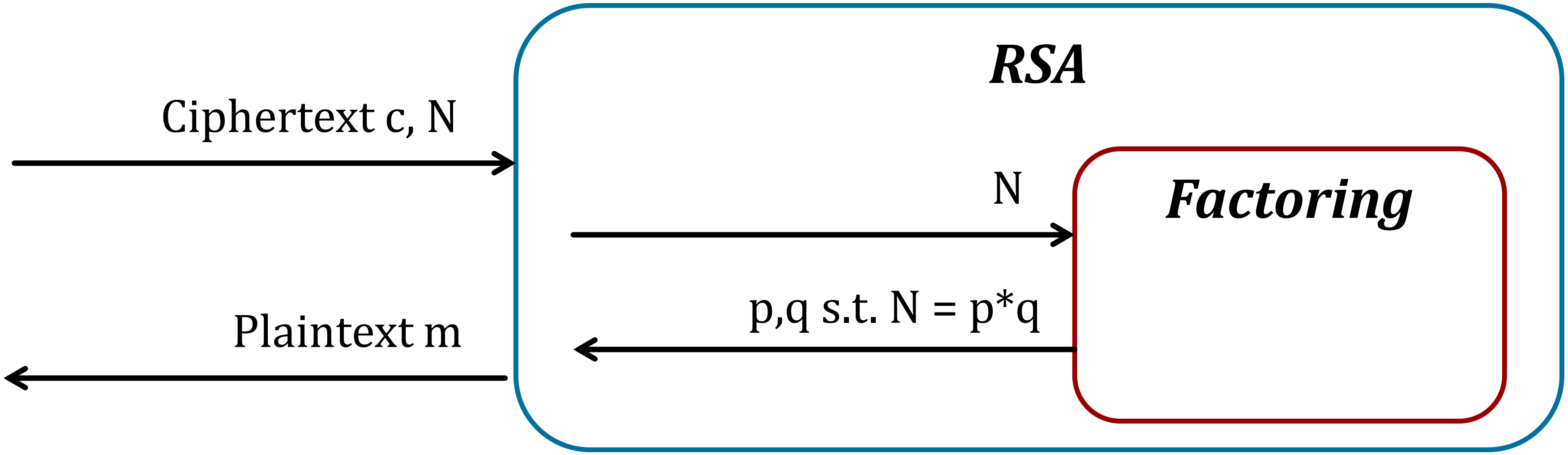
Security Reductions

Reduction: Problem ***A*** is at least as hard as ***B*** if an algorithm for solving ***A*** efficiently (if it existed) could also be used as a subroutine to solve problem ***B*** efficiently.



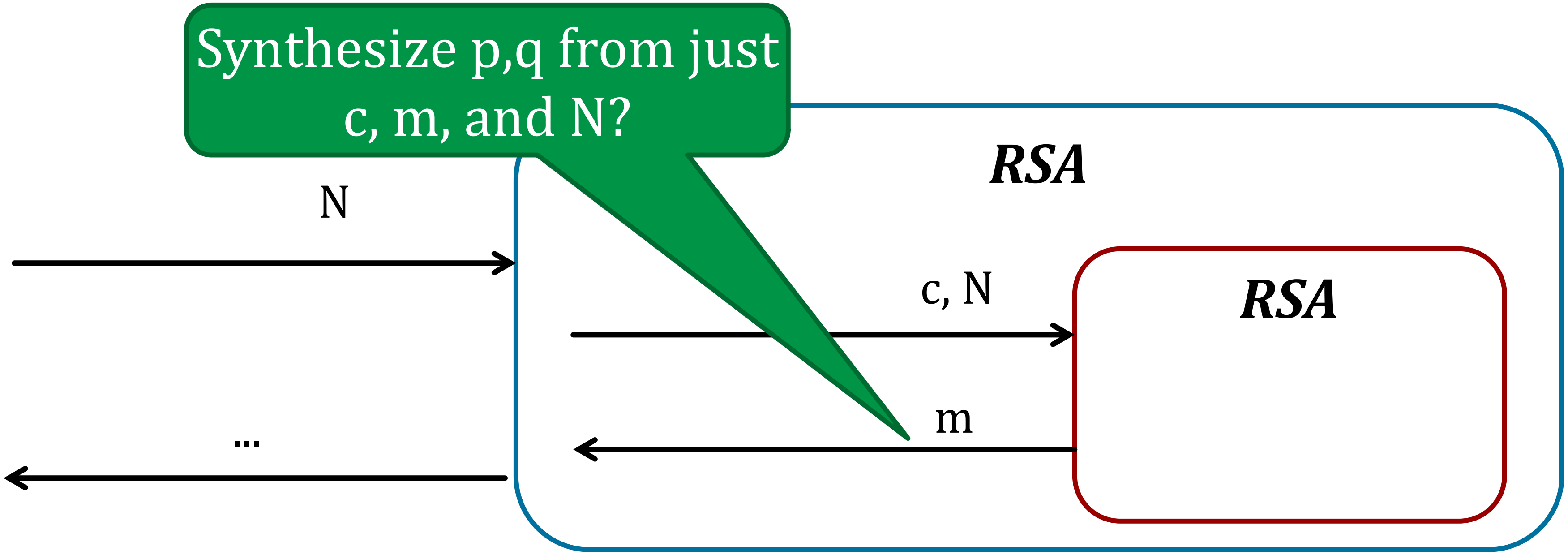
Example

<u>*Reduction*</u>: Problem ***Factoring (A)*** is at least as hard as ***RSA (B)*** if an algorithm for solving ***Factoring (A)*** efficiently (if it existed) could also be used as a subroutine to solve problem ***RSA (B)*** efficiently.

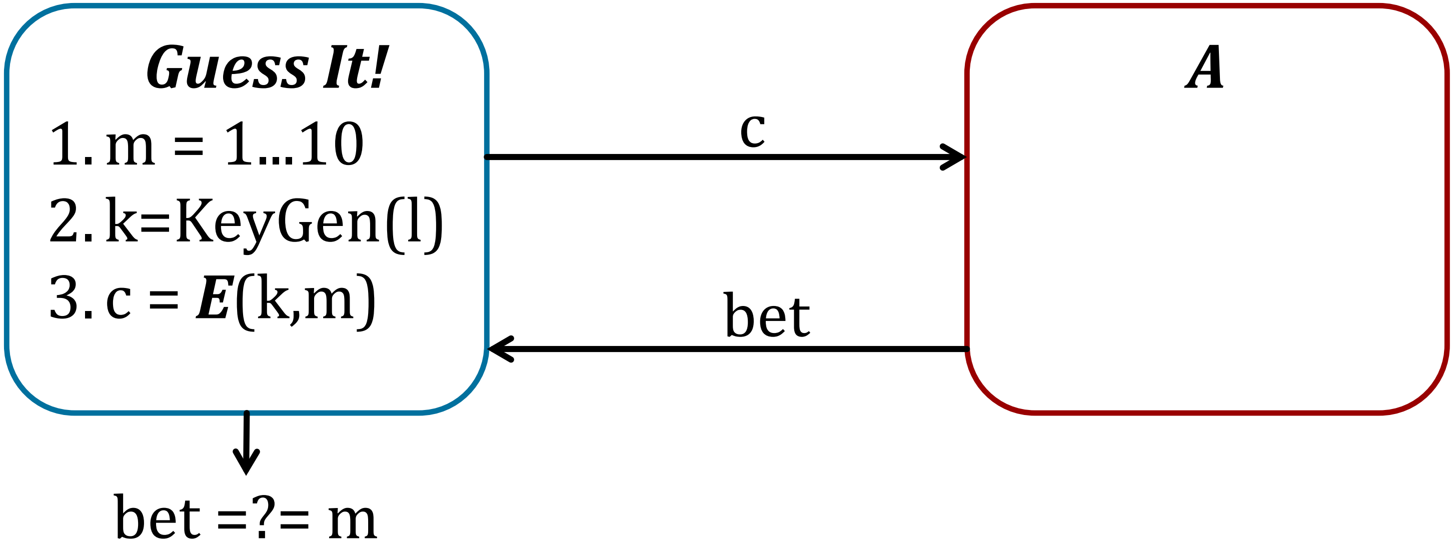


What's *unknown*...

Reduction: Problem ***RSA (A)*** is at least as hard as ***Factoring (B)*** if an algorithm for solving ***RSA (A)*** efficiently (if it existed) could also be used as a subroutine to solve problem ***Factoring (B)*** efficiently.



Games and Reductions



Suppose ***A*** is in a guessing game. Guess It! uses ***E*** to encrypt. How can we prove, in this setting, that ***E*** is secure?

Reduction: If ***A*** does better than 1/10, we break ***E*** in the semantic security game. Showing security of ***E*** reduces to showing if ***A*** exists, it could break the semantic security game. (Equivalently, if ***E*** is semantically secure, then the probability ***A*** wins is at most 10%.)

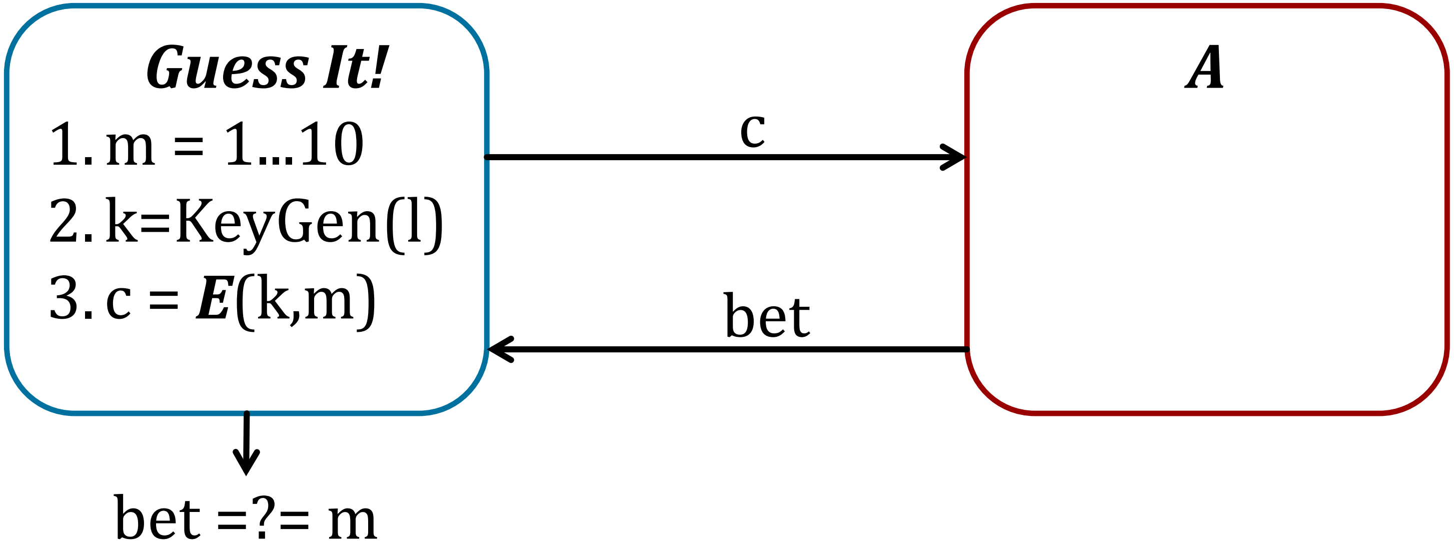
Note: The "type" of A is A: c -> bet

Idea

<u>*Reduction*</u>: We build an adversary $\boldsymbol{B}$ that uses $\boldsymbol{A}$ as a subroutine. Our adversary $\boldsymbol{B}$ has the property if $\boldsymbol{A}$ wins at Guess It! with probability significantly greater than 10%, $\boldsymbol{B}$ will have a non-negligible advantage in our semantic security game.

- If $\boldsymbol{E}$ secure, Guess It! is secure.
- Equivalently, if Guess It! insecure, $\boldsymbol{E}$ is insecure

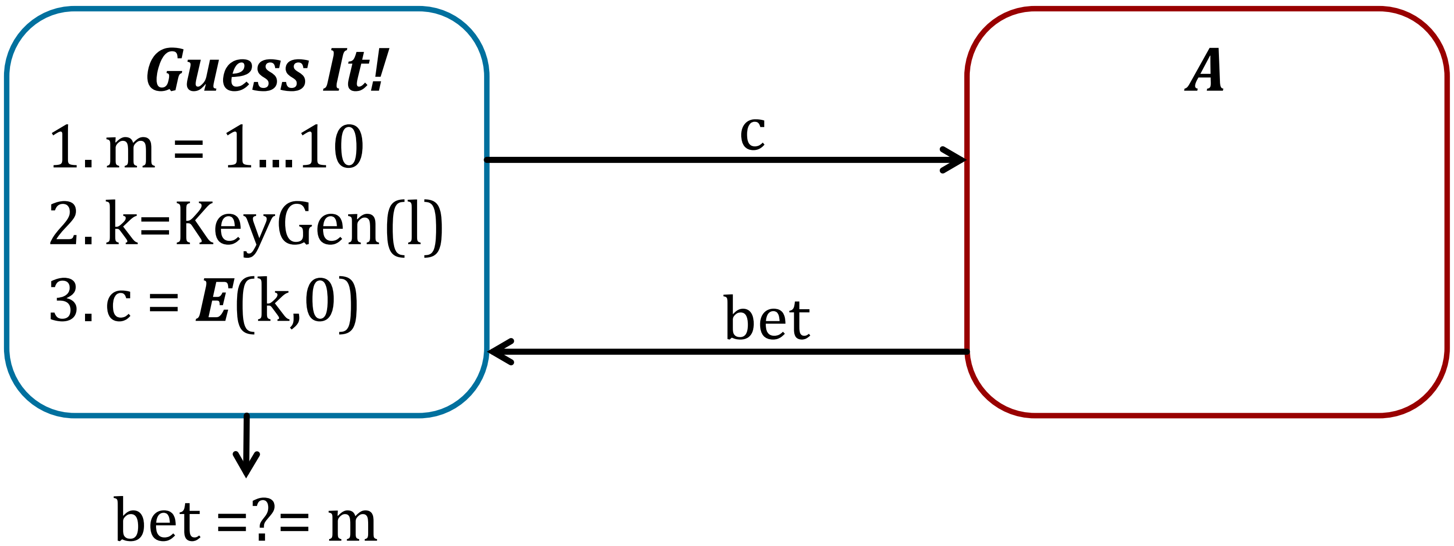
The Real Version



In the *real* version, ***A*** always gets an encryption of the real message.

– Pr[A wins in *real* version] = p_0

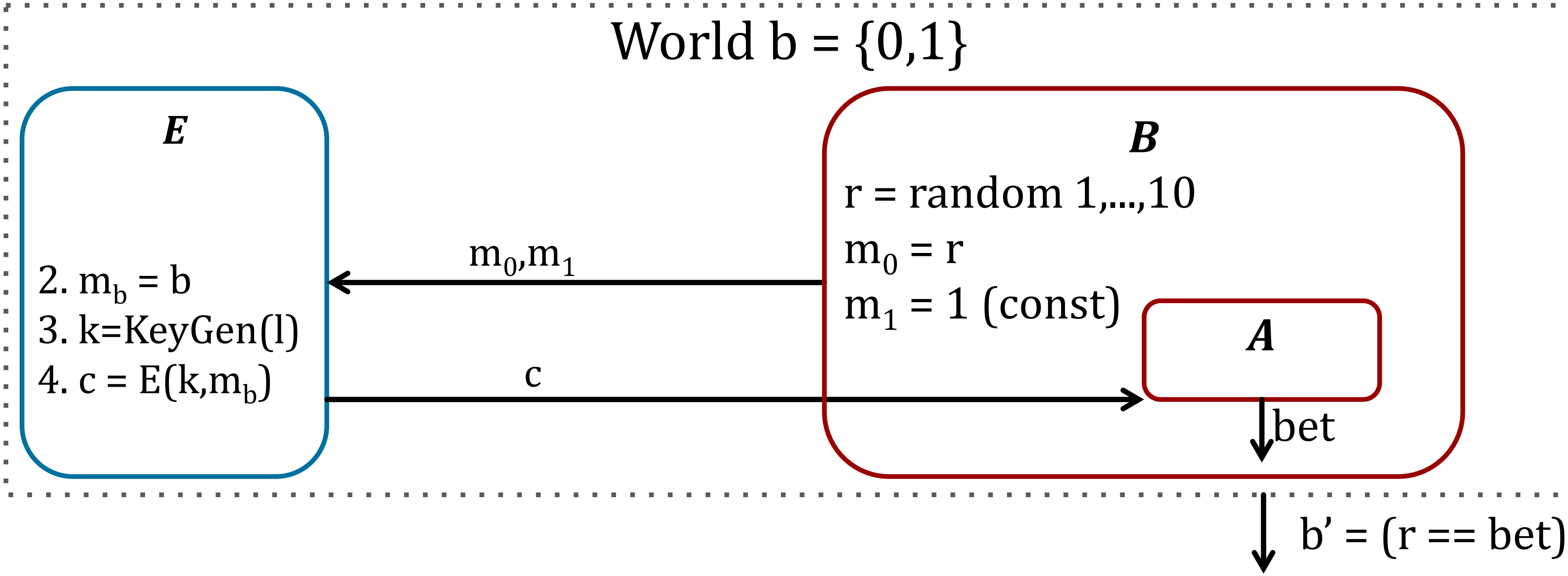
Idealized Version



In the ideal version, ***A*** always gets an encryption of a constant, say 1. (A still only wins if it gets *m* correct.)

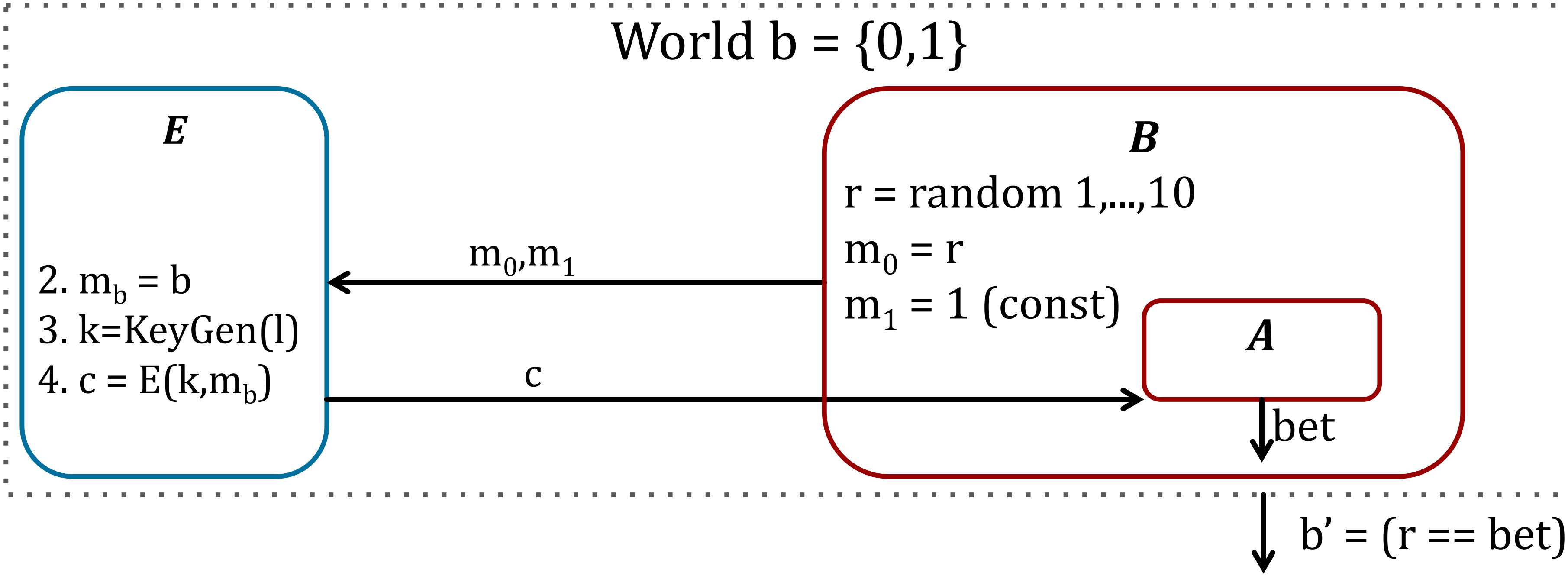
– Pr[A wins in Idealized Version] = p_1 = 1/10

Reduction



- If B is in world 0, then Pr[b' = 1] = p_0
  - B can guess r==bet with prob. p_0.
- If B is in world 1, then Pr[b' = 1] = p_1 = 1/10
- For b=0,1: W_b := [event that ***B***(W_b) =1]
  $\text{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}] = |\,\Pr[\,W_0\,] - \Pr[\,W_1\,]\,|$
  $= |p_0 - p_1|$

Reduction



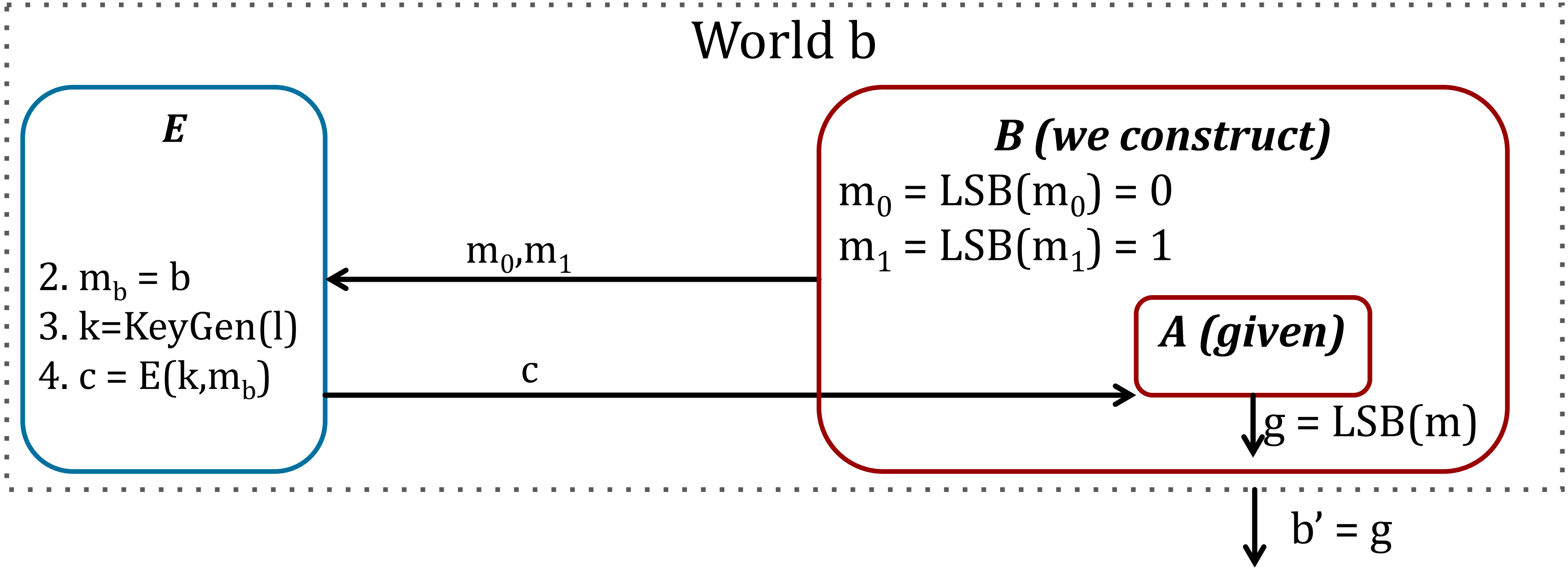
- If B is in world 0, then Pr[b' = 1] = p_0
  - B can guess r==bet with prob. p_0.
- If B is in world 1, then Pr[b' = 1] = p_1 = 1/10
- For b=0,1: W_b := [event that ***B***(W_b) =1]
  $\mathrm{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}]$ = | Pr[W_0] − Pr[W_1] |
  = $|p_0 - p_1|$

Suppose 33% correct

33%-%10 = 23% Advantage

Reduction Example 2

Suppose efficient A can always deduce LSB of PT from CT.
Then E = (E,D) is not semantically secure.



$$\text{Adv}_{SS}[\boldsymbol{A},\boldsymbol{E}] = |\Pr[W_0] - \Pr[W_1]| = |0 - 1| = 1$$

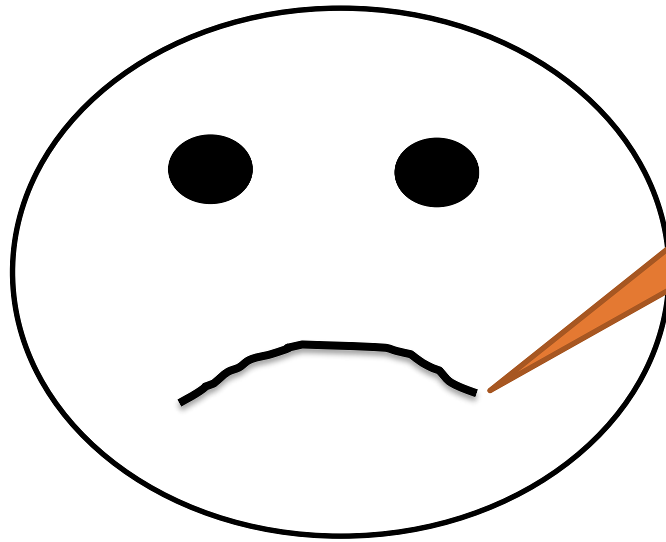
Questions?

END

Thought

The “Bad News” Theorem

<u>Theorem</u>: Perfect secrecy requires |K| >= |M|

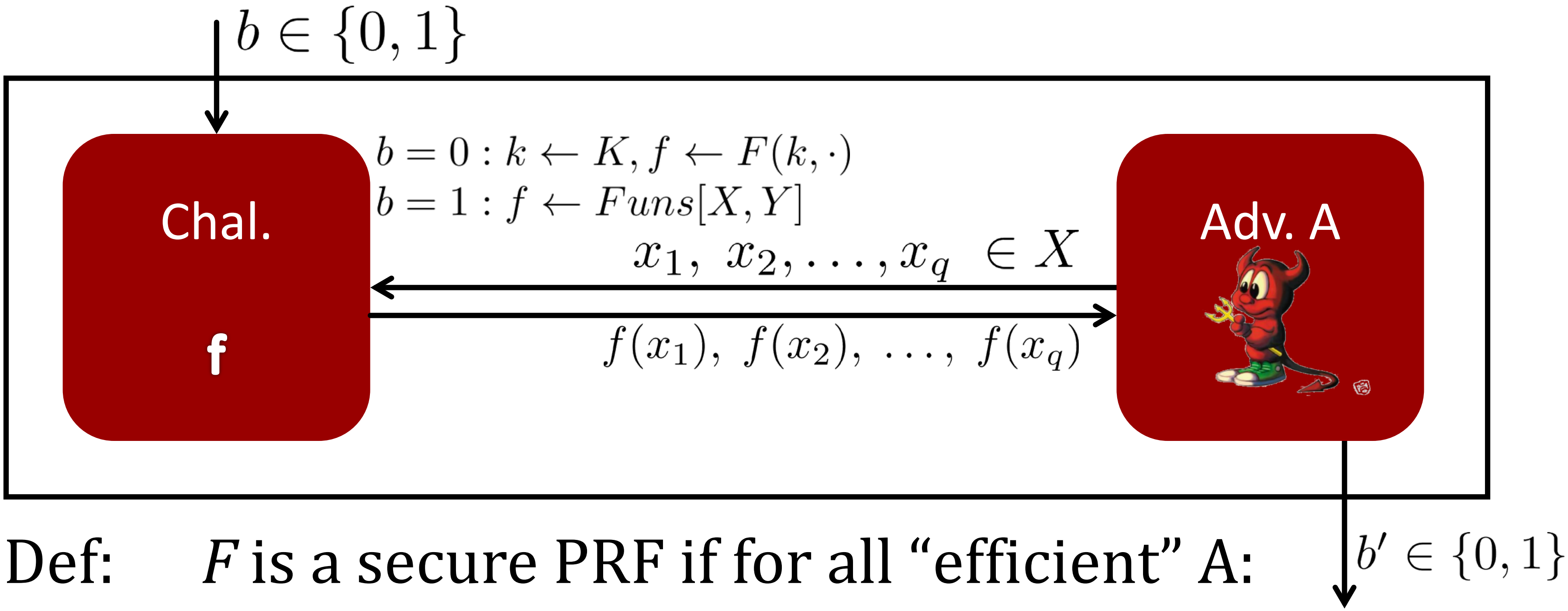


And what about integrity and authenticity?

In practice, we usually shoot for *<u>computational security</u>*.

Secure PRF: Definition

- For *b = 0,1* define experiment *EXP(b)* as:



- Def: *F* is a secure PRF if for all "efficient" A:

$$Adv_{PRF}[A,F] := |Pr[EXP(0)=1] - Pr[EXP(1)=1]|$$

is "negligible".

Quiz

Let $F : K \times X \to \{0,1\}^{128}$ be a secure PRF.

Is the following G a secure PRF?

$$G(k,x) = \begin{cases} 0^{128} & \text{if } x = 0 \\ F(k,x) & \text{otherwise} \end{cases}$$

- ● No, it is easy to distinguish G from a random function
- ○ Yes, an attack on G would also break F
- ○ It depends on F

Secure PRPs (secure block cipher)

- Let $E : K \times X \to Y$ be a PRP$(X = Y)$

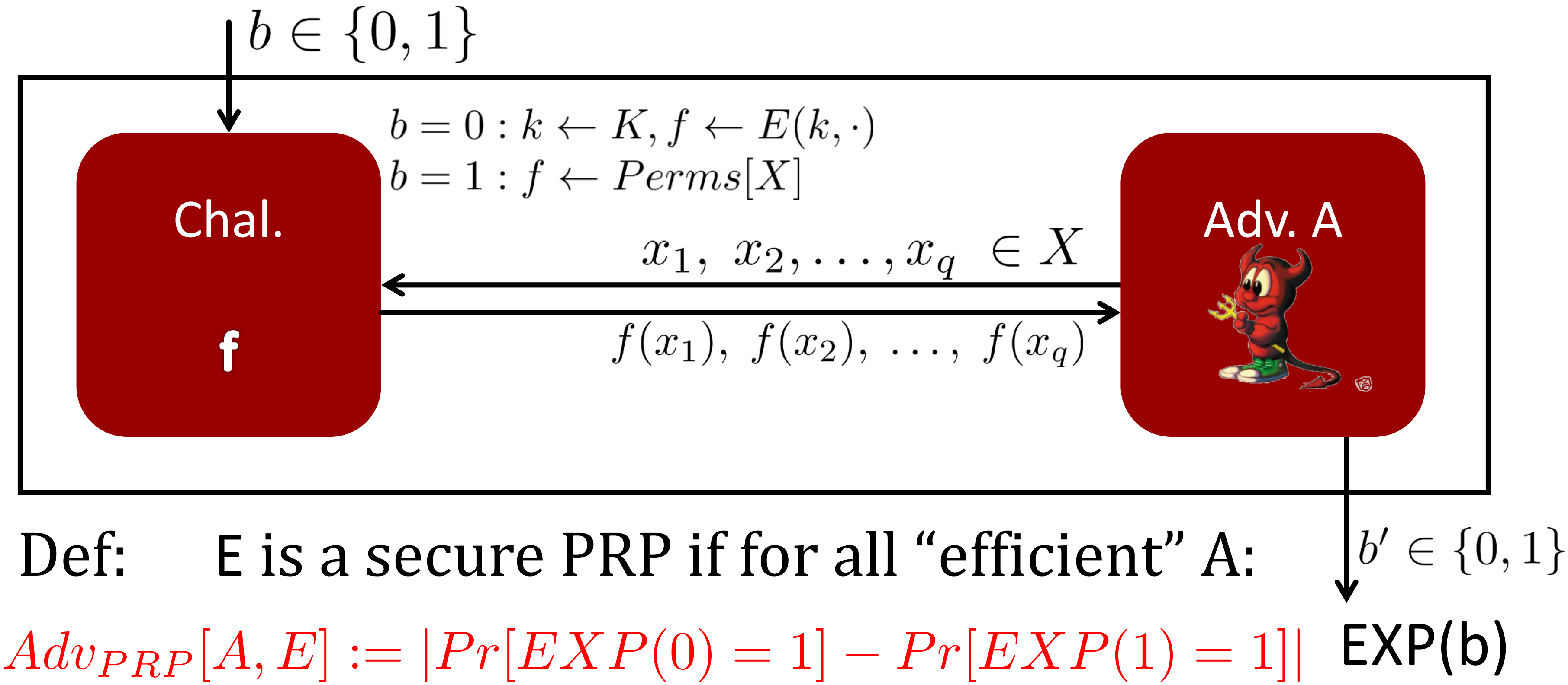
$$\begin{cases} Perms[X] : \text{the set of all } \underline{\textbf{one-to-one}} \text{ functions from } X \text{ to } Y \\ S_F = \{E(k, \cdot) \quad \text{s.t.} \quad k \in K\} \subseteq Perms[X] \end{cases}$$

- <u>Intuition:</u> a PRP is **secure** if

  A random function in *Perms[X]* is indistinguishable from a random function in S_F

Secure PRP: (secure block cipher)

- For b = 0,1 define experiment EXP(b) as:



- Def: E is a secure PRP if for all "efficient" A:

$$Adv_{PRP}[A,E] := |Pr[EXP(0)=1] - Pr[EXP(1)=1]|$$

is "negligible".

Modern Notions: Indistinguishability and Semantic Security