SE 4472b

Information Security

Week 2-2

Some Formal Security Notions

Aleksander Essex

Western Engineering

Fall 2015

Formalizing Security

As we saw, classical ciphers leak information:

- Caeser/Vigenere leaks letter frequency information
- Enigma leaks information about what the plaintext *isn't*

We need some way to formalize what our goals are. What should it mean for a cipher to be secure?

Formalizing Security

- Information-theoretic security is a strong notion, but hard to achieve in practice. We want to be able to use short, fixed-length keys.
- In this *computational* setting, it is possible to crack an ciphertext by trying all the keys (a so-called *brute-force* attack).
- However, barring the ability for an attacker to brute-force the key, is there some way we can express security
  Similar to Shannon's proof that the ciphertext of one-time pad doesn't reveal any information about the message, and barring the ability of an attacker to brute force the key, can we form similar security notions?

The Big Idea

Here's an idea: what if we played a game.

- I flip a coin an pick one of two messages and encrypt it.
- I give you the ciphertext and you have to tell me which message picked.
- If you guess correctly, you win.
- If you guess incorrectly, I win.
- Just to make things interesting, I'll even let *you* pick the two messages.

If you guess randomly, you'll be right 50% of the time. So here's the question: can you win *more* than 50% of the time?

The Big Idea

Let's think about the implications of this game:

- If you guess randomly, you'll win 50% of the time.
- Is there a (cryptanalytic) strategy you can take to win *more* than 50% of the time?
- If you can win more than 50% of the time, we say you can *distinguish* ciphertexts. That means the cryptosystem is leaking information, and that's a bad thing.
- On the other hand, if you can't win more than 50% of the time, that's an indication maybe the cryptosystem isn't leaking information, at least not enough for you to have an advantage winning in winning the game.

Security Games

We can formalize different security levels by playing adversarial games:

- Game between two players: an adversary and a message holder
- Used for modeling what an adversary can learn about a message from its encryption.
- Cryptosystem is secure if ciphertexts are *indistinguishable*

Security Game

Here's the high level way the game is played:

1. Two players: A, B (in this setting A is the ``adversary'')
2. B starts by choosing a secret key
3. Game proceeds in phases:
    - Query: Depending on the game, A can make queries to B (more later)
    - Challenge: A chooses a pair of messages (of equal length) and sends them to B
    - Response: B chooses one message at random, encrypts it, and sends the ciphertext to A
    - Guess: A has to decide which message B chose. A ``wins'' if it correctly guesses correctly.
4. The game can be played again and again and the adversary's goal is to win more than 50% of the time.

Indistinguishably of Encryptions under Eavesdropping.

- In the most basic game, we are interested in what the adversary can learn simply from eavesdropping on the messages.
- In subsequent games, we will give the adversary more powers to ask questions of B to help it try to win the game.
- In this game, however, we are interested in determining whether a given cryptosystem produces ciphertexts that are indistinguishable under eavesdropping. We call this game EAV.
- If the adversary cannot win this game more than 50% of the time, we say a cryptosystem is IND-EAV secure. This is just a rough pass on a definition. We'll formalize it later.

The EAV Game

Public inputs: Encryption function $E()$ and key length s.

A	B
	Generate random key $k \in_r \{0,1\}^s$

The EAV Game

Public inputs: Encryption function $E()$ and key length s.

A

B

Generate random key $k \in_r \{0,1\}^s$

Send challenge messages $\xrightarrow{m_0, m_1}$

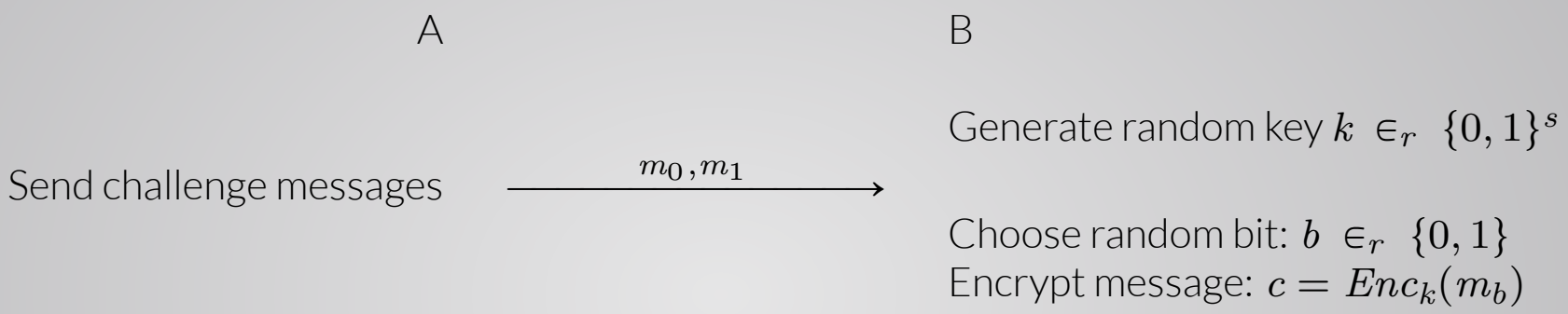
The EAV Game

Public inputs: Encryption function $E()$ and key length s.

A		B
		Generate random key $k \in_r \{0,1\}^s$
Send challenge messages	$\xrightarrow{m_0, m_1}$	
		Choose random bit: $b \in_r \{0,1\}$

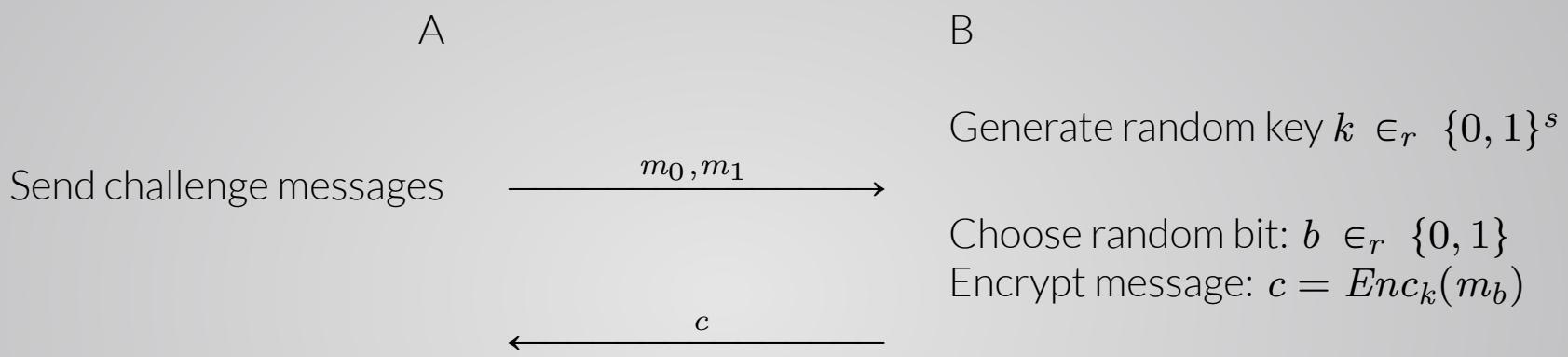
The EAV Game

Public inputs: Encryption function $E()$ and key length s.



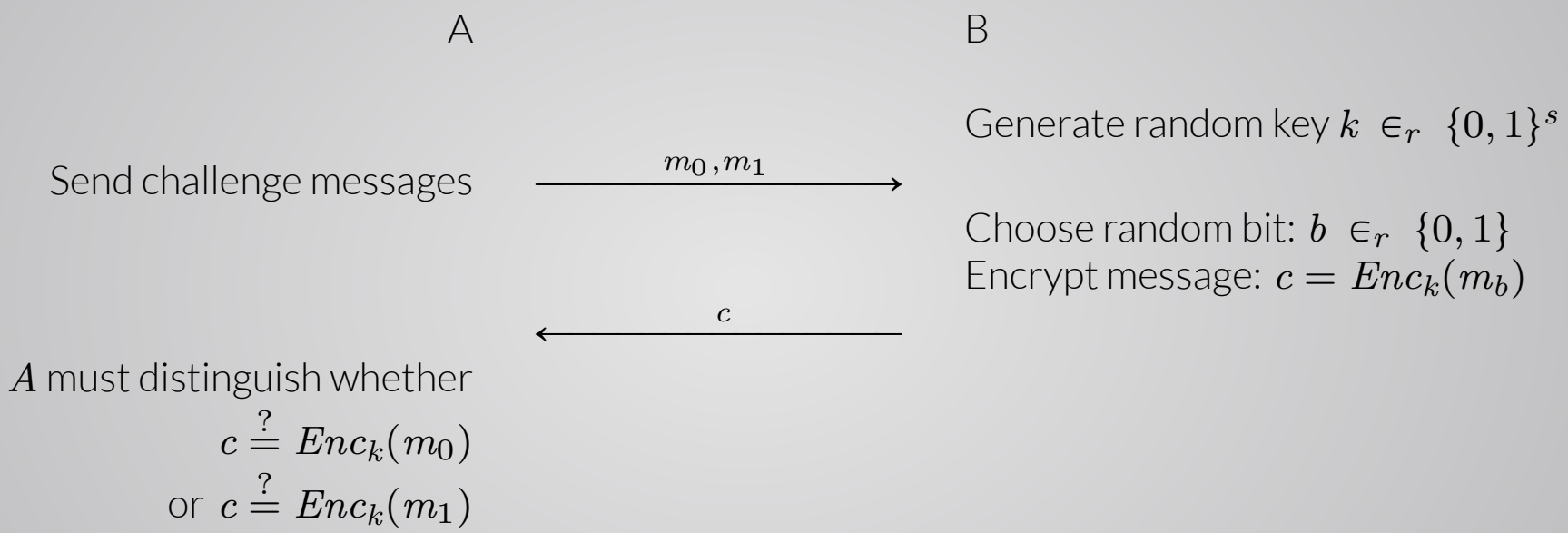
The EAV Game

Public inputs: Encryption function $E()$ and key length s.



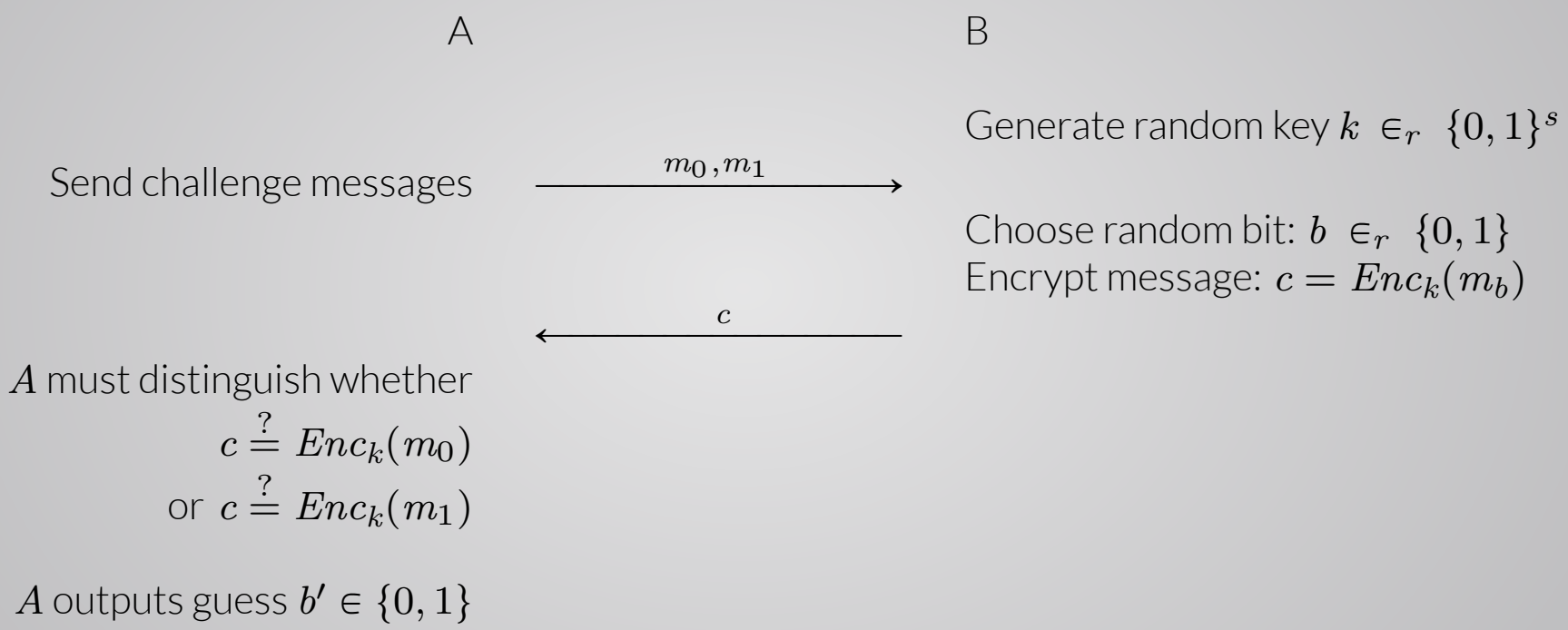
The EAV Game

Public inputs: Encryption function $E()$ and key length s.



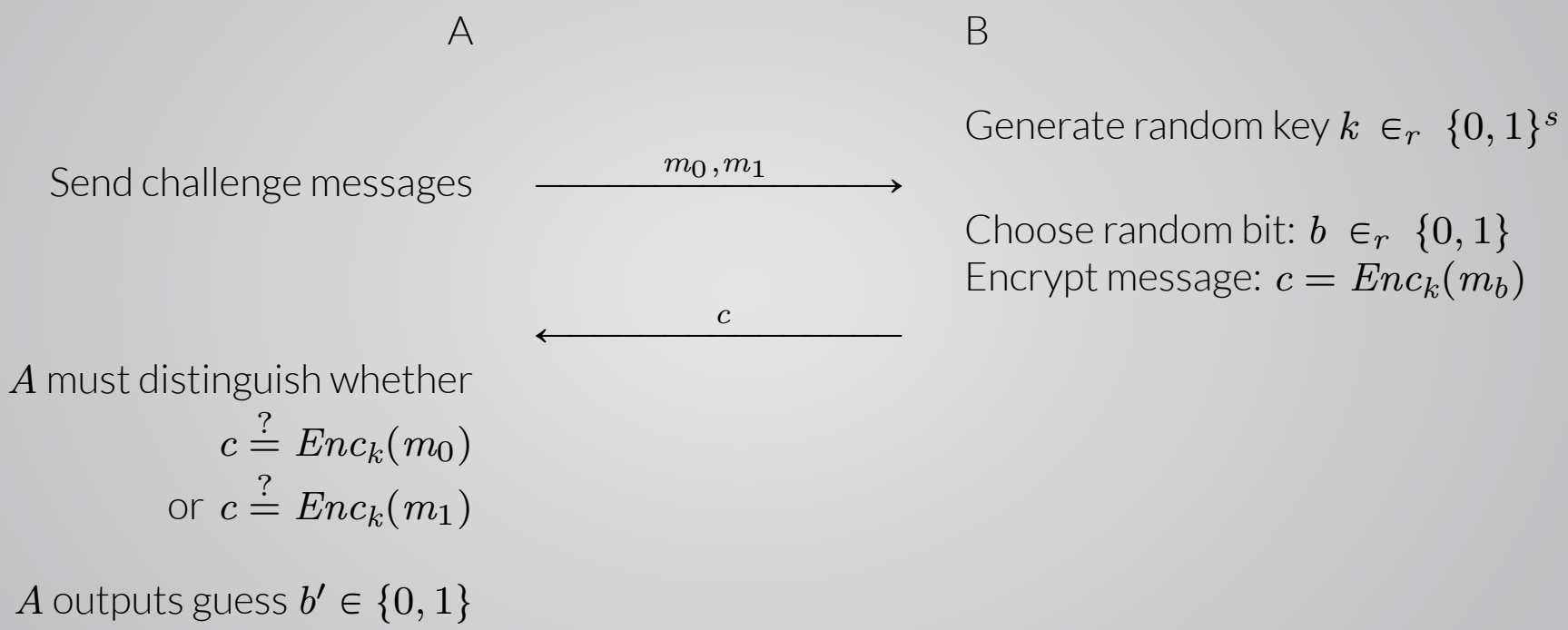
The EAV Game

Public inputs: Encryption function $E()$ and key length s.



The EAV Game

Public inputs: Encryption function $E()$ and key length s.



We say A wins if $b' = b$. The game can be replayed arbitrarily many times. We're interested in how often A wins.

The EAV Game

For each of the following ciphers, prove whether it is IND-EAV secure, or not:

1. Caesar cipher
2. Enigma machine
3. One-time pad

Remember: the adversary isn't trying to be correct every time, just with greater than negligible advantage (we'll define what we mean by *negligible* in the next lecture).