LDACS1310: Introduction to Applied Cryptography

Lecture 2: symmetric encryption

T. Peters, ***F.-X. Standaert***

UCLouvain Crypto Group, fstandae@uclouvain.be

Reminder



https://perso.uclouvain.be/fstandae/LDACS1310/

And questions from last lecture?

Assumptions dictionary

- Pseudo-random generator: $\mathrm{G} : \{0,1\}^n \to \{0,1\}^l : k \to \mathrm{G}(k), l > n$

$$\left|\Pr\left(\mathrm{Dist}\left(\mathrm{G}(k)\right) = 1\right) - \Pr(\mathrm{Dist}(R) = 1)\right| \leq \mathrm{negl}(n)$$

❑ Question: # of l-bit bitstrings: , # of n-bit keys:

Assumptions dictionary

- Pseudo-random generator: $\mathrm{G} : \{0,1\}^n \to \{0,1\}^l : k \to \mathrm{G}(k), l > n$

$$\left|\Pr\left(\mathrm{Dist}\left(\mathrm{G}(k)\right) = 1\right) - \Pr(\mathrm{Dist}(R) = 1)\right| \leq \mathrm{negl}(n)$$

❑ Question: # of l-bit bitstrings: 2^l, # of n-bit keys: 2^n

Assumptions dictionary

- Pseudo-random generator: $\mathrm{G} : \{0,1\}^n \to \{0,1\}^l : k \to \mathrm{G}(k), l > n$

$$\left|\Pr\left(\mathrm{Dist}\left(\mathrm{G}(k)\right) = 1\right) - \Pr(\mathrm{Dist}(R) = 1)\right| \leq \mathrm{negl}(n)$$

- PR function: $\mathrm{F} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^o : \; x \to \mathrm{F}_k(x)$

$$\left|\Pr\left(\mathrm{Dist}^{\mathrm{F}k(\cdot)} = 1\right) - \Pr\left(\mathrm{Dist}^{f(\cdot)} = 1\right)\right| \leq \mathrm{negl}(n)$$

❑ Question: # of n-bit keys: , # of m-bit to o-bit functions:

Assumptions dictionary

- Pseudo-random generator: $\mathrm{G} : \{0,1\}^n \to \{0,1\}^l : k \to \mathrm{G}(k), l > n$

$$\left|\Pr\left(\mathrm{Dist}\left(\mathrm{G}(k)\right) = 1\right) - \Pr(\mathrm{Dist}(R) = 1)\right| \leq \mathrm{negl}(n)$$

- PR function: $\mathrm{F} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^o : \; x \to \mathrm{F}_k(x)$

$$\left|\Pr\left(\mathrm{Dist}^{\mathrm{F}k(\cdot)} = 1\right) - \Pr\left(\mathrm{Dist}^{f(\cdot)} = 1\right)\right| \leq \mathrm{negl}(n)$$

❑ Question: # of n-bit keys: 2^n, # of m-bit to o-bit functions: $2^{o \cdot 2^m}$

Assumptions dictionary

- Pseudo-random generator: $\mathrm{G} : \{0,1\}^n \to \{0,1\}^l : k \to \mathrm{G}(k), l > n$

$$\left|\Pr\left(\mathrm{Dist}\left(\mathrm{G}(k)\right) = 1\right) - \Pr(\mathrm{Dist}(R) = 1)\right| \leq \mathrm{negl}(n)$$

- PR function: $\mathrm{F} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^o : \; x \to \mathrm{F}_k(x)$

$$\left|\Pr\left(\mathrm{Dist}^{\mathrm{F}k(\cdot)} = 1\right) - \Pr\left(\mathrm{Dist}^{f(\cdot)} = 1\right)\right| \leq \mathrm{negl}(n)$$

- PR permutation: $\mathrm{P} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m : \; x \to \mathrm{P}_k(x)$

$$\left|\Pr\left(\mathrm{Dist}^{\mathrm{P}k(\cdot)} = 1\right) - \Pr\left(\mathrm{Dist}^{p(\cdot)} = 1\right)\right| \leq \mathrm{negl}(n)$$

❑ Question: # of n-bit keys: , # of m-bit permutations:

Assumptions dictionary

- Pseudo-random generator: $\mathrm{G} : \{0,1\}^n \to \{0,1\}^l : k \to \mathrm{G}(k), l > n$

$$\left|\Pr\left(\mathrm{Dist}\left(\mathrm{G}(k)\right) = 1\right) - \Pr(\mathrm{Dist}(R) = 1)\right| \leq \mathrm{negl}(n)$$

- PR function: $\mathrm{F} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^o : \; x \to \mathrm{F}_k(x)$

$$\left|\Pr\left(\mathrm{Dist}^{\mathrm{F}k(\cdot)} = 1\right) - \Pr\left(\mathrm{Dist}^{f(\cdot)} = 1\right)\right| \leq \mathrm{negl}(n)$$

- PR permutation: $\mathrm{P} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m : \; x \to \mathrm{P}_k(x)$

$$\left|\Pr\left(\mathrm{Dist}^{\mathrm{P}k(\cdot)} = 1\right) - \Pr\left(\mathrm{Dist}^{p(\cdot)} = 1\right)\right| \leq \mathrm{negl}(n)$$

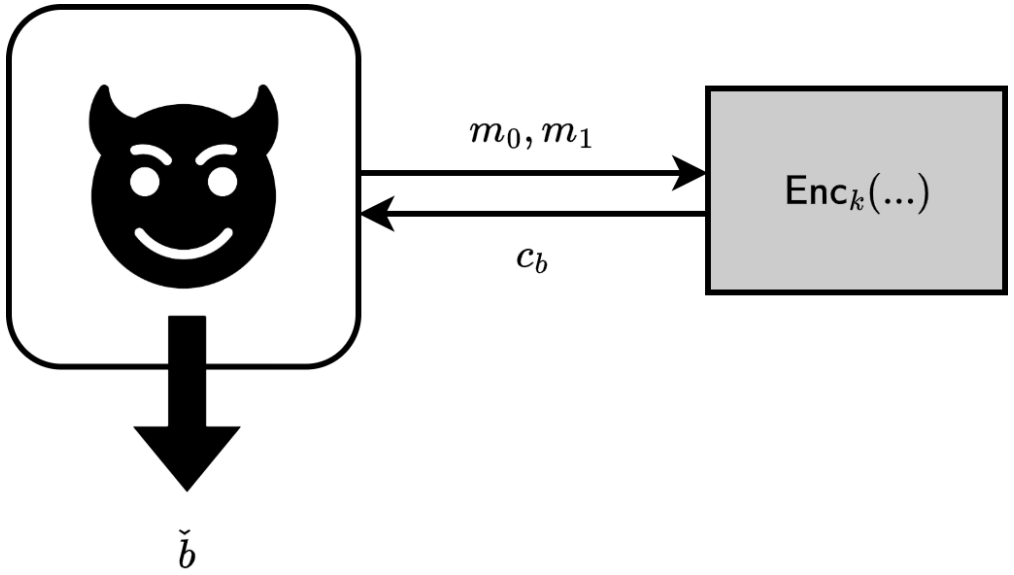
❑ Question: # of n-bit keys: 2^n, # of m-bit permutations: $(2^m)!$

Eavesdropper security (I)

- How to define confidentiality in our threat model? {…}

Eavesdropper security (I)

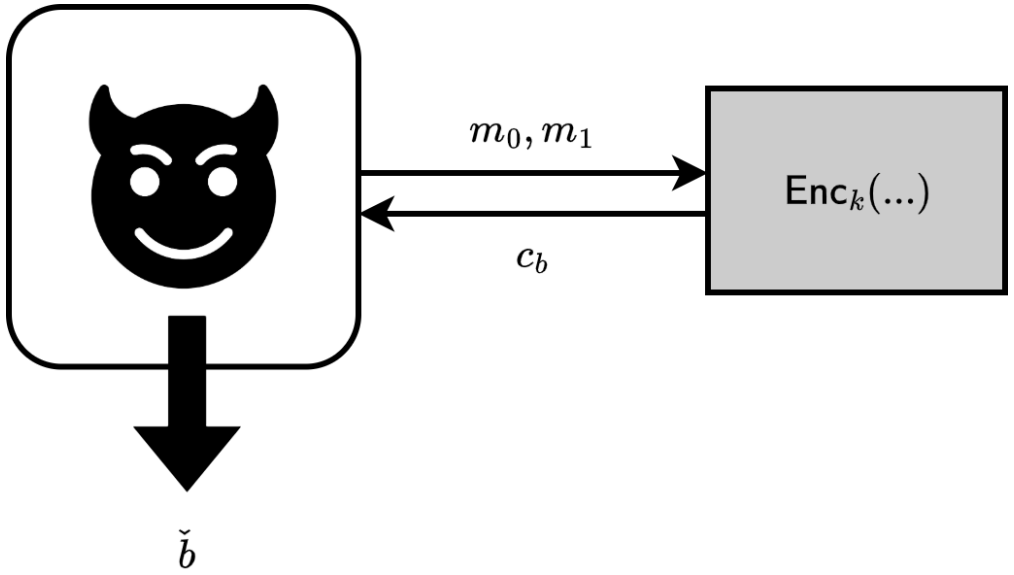
- How to define confidentiality in our threat model? {…}
- Eavesdropper experiment (single-message attacks) {…}



How long can the messages be?

Eavesdropper security (I)

- How to define confidentiality in our threat model? {…}
- Eavesdropper experiment (single-message attacks) {…}



How long can the messages be?
$l = \mathrm{poly}(n)$

- An encryption scheme $\mathrm{SE} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is secure against eavesdropping if $\forall$ polytime Adv, $\Pr(\mathrm{Exp}^{\mathrm{eav}}_{\mathrm{SE}}(n) = 1) \leq \frac{1}{2} + \mathrm{negl}(n)$

Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$

❑ Question: if using the one-time pad, $\Pr(\mathrm{Exp}^{\mathrm{eav}}_{\mathrm{SE}}(n) = 1) =$

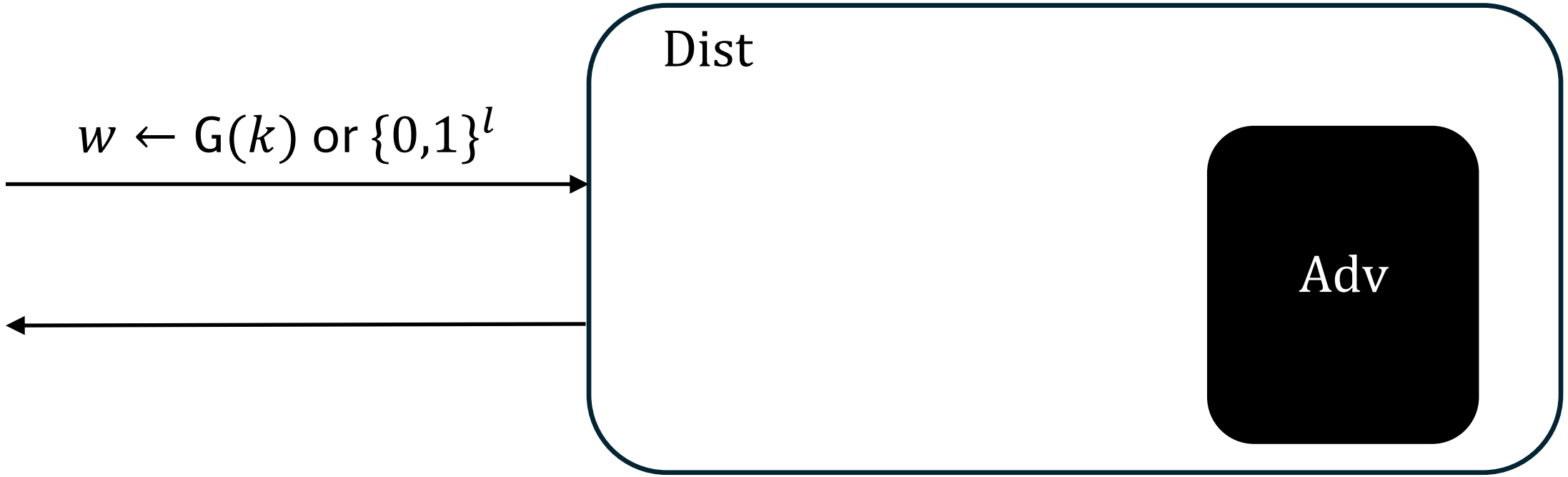
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$

❑ Question: if using the one-time pad, $\Pr(\mathrm{Exp}^{\mathrm{eav}}_{\mathrm{SE}}(n) = 1) = \frac{1}{2}$

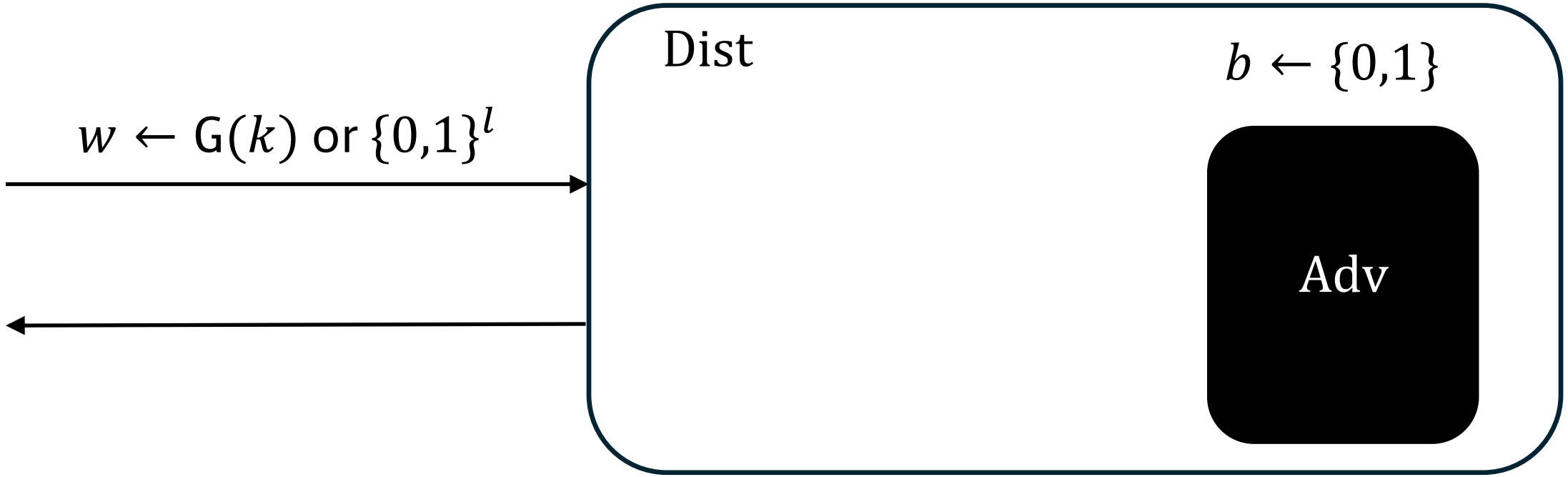
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



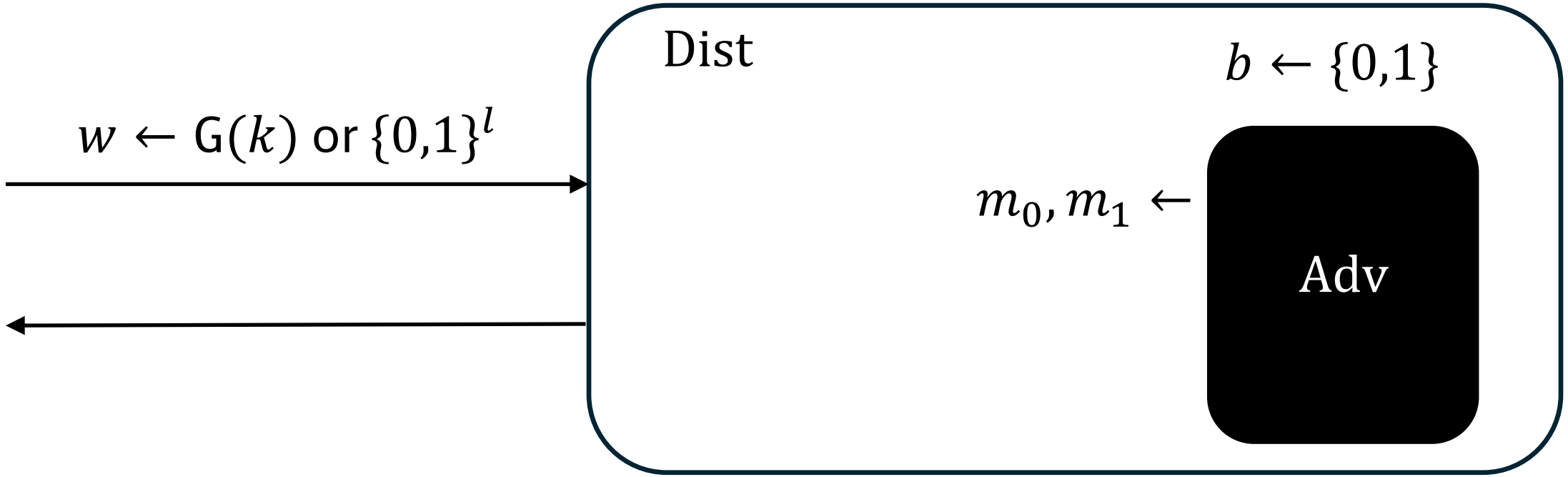
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



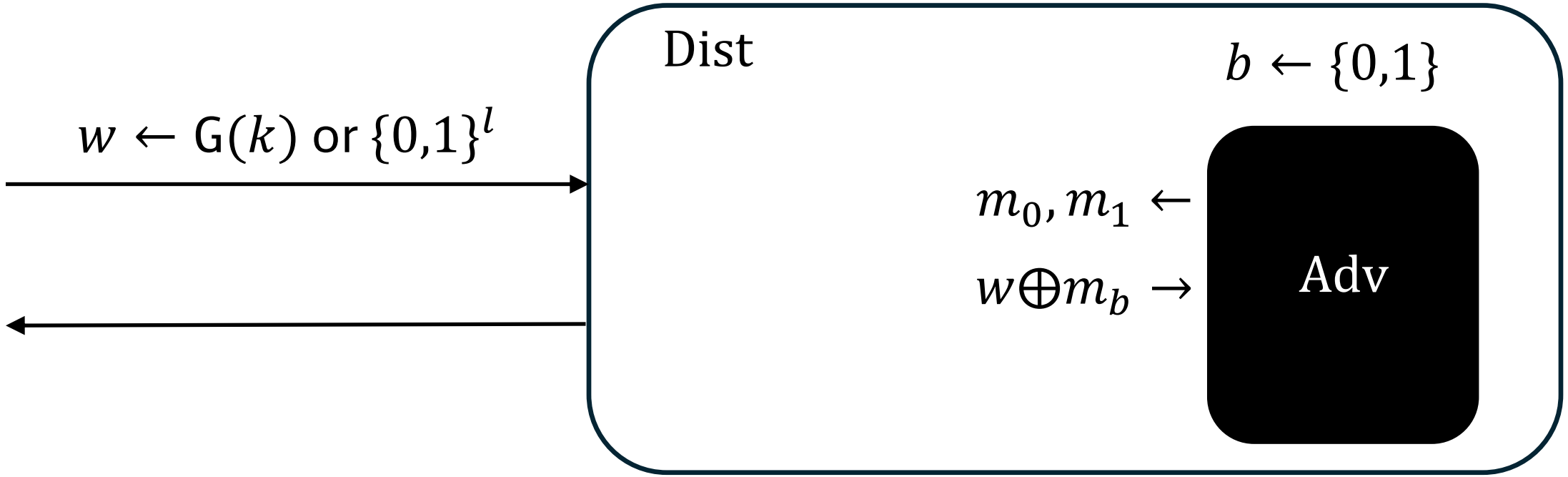
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



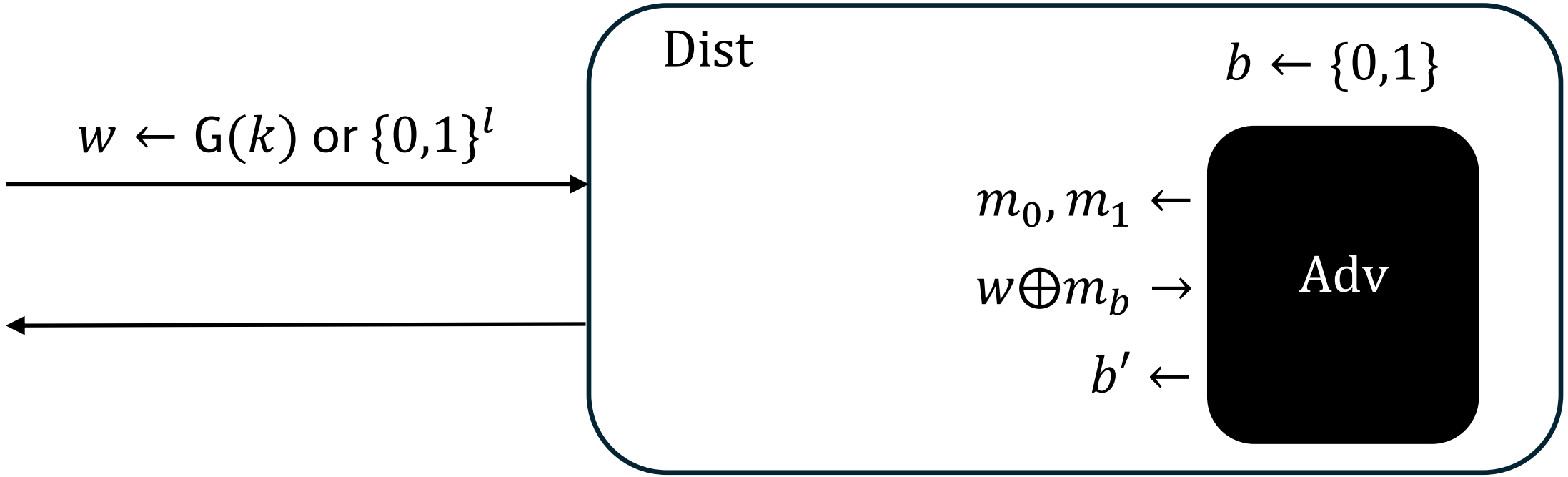
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



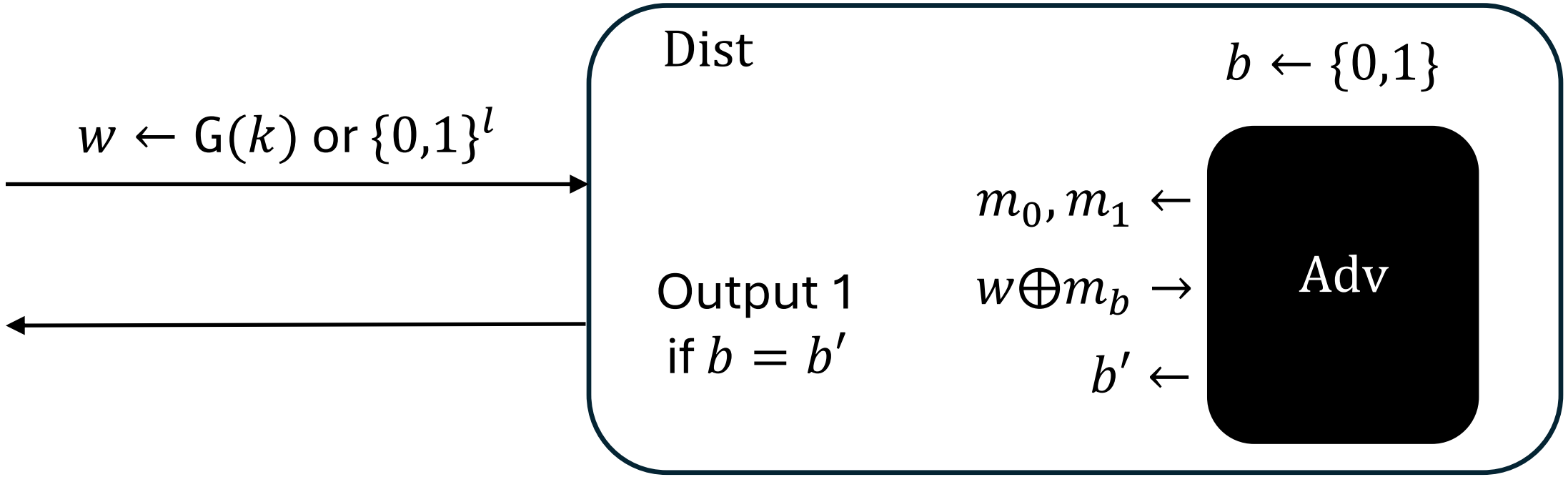
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



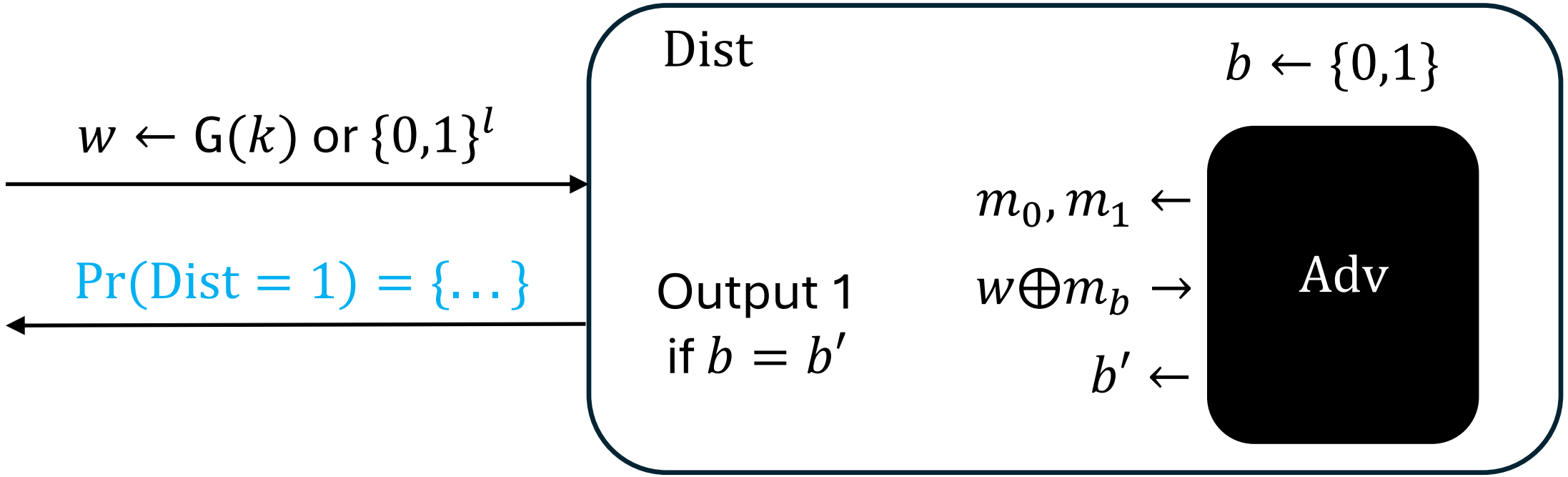
Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



Eavesdropper security (II)

- High-level design idea: emulate one-time pad with a PRG
  - $c = \mathrm{Enc}_k(m) = m \oplus \mathrm{G}(k), m = \mathrm{Dec}_k(c) = c \oplus \mathrm{G}(k)$
- Security reduction: if Adv. Breaks SE, she must distinguish G



Concrete security and limitations

- If G is a (t, ϵ_G) PRG, SE is $(t - c, \epsilon_G)$ secure against eavesdropping
  - With c a constant capturing the "additional" work of Dist (vs. Adv)

❑ Question: as designers, do we hope c to be small or big?

Concrete security and limitations

- If G is a (t, ϵ_G) PRG, SE is $(t - c, \epsilon_G)$ secure against eavesdropping
  - With c a constant capturing the "additional" work of Dist (vs. Adv)

❑ Question: as designers, do we hope c to be small or big?
Small (e.g., think about additional work is exhaustive key search)
Concretely, one hopes small ε values for large t values

Concrete security and limitations

- If G is a (t, ϵ_G) PRG, SE is $(t - c, \epsilon_G)$ secure against eavesdropping
  - With c a constant capturing the "additional" work of Dist (vs. Adv)

❑ Question: as designers, do we hope c to be small or big?
Small (e.g., think about additional work is exhaustive key search)
Concretely, one hopes small ε values for large t values

- Limitations of eavesdropper security?

Concrete security and limitations

- If G is a (t, ϵ_G) PRG, SE is $(t - c, \epsilon_G)$ secure against eavesdropping
  - With c a constant capturing the "additional" work of Dist (vs. Adv)

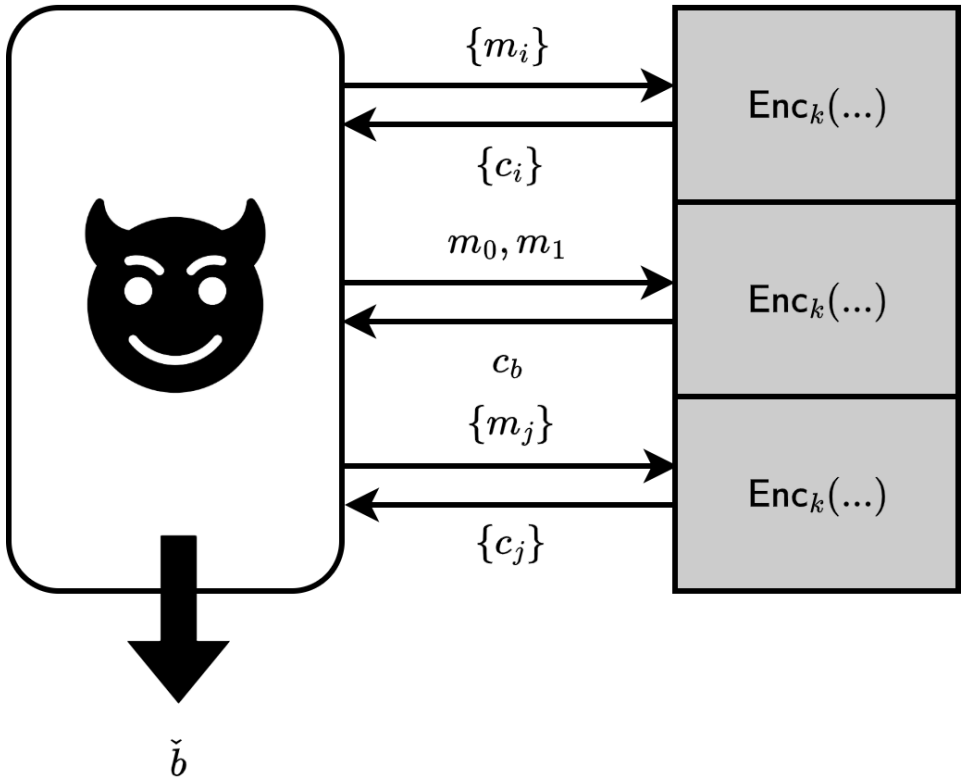
❑ Question: as designers, do we hope c to be small or big?
Small (e.g., think about additional work is exhaustive key search)
Concretely, one hopes small ε values for large t values

- Limitations of eavesdropper security
  1. Not unconditional: needs a secure PRG (cryptanalysis)
  2. Insecure against multi-message attacks (initialization issue)

  ⇒ We need a stronger security definition to cover this case

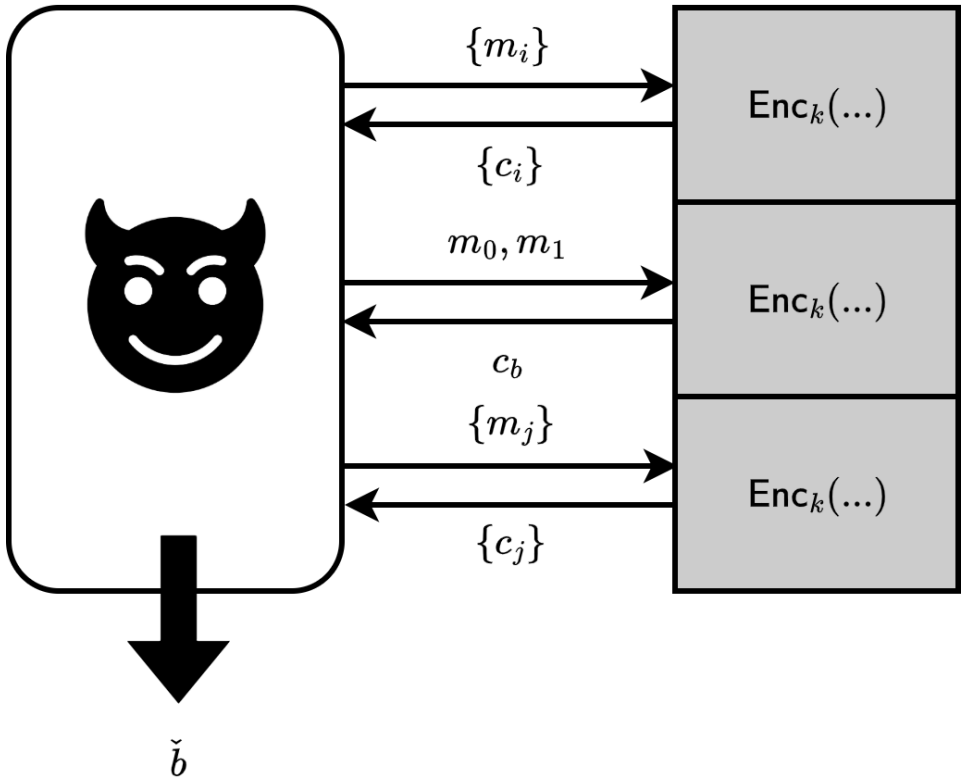
Security against Chosen Plaintext Attacks (CPA)

- CPA security {…}. How many queries can be made?



Security against Chosen Plaintext Attacks (CPA)

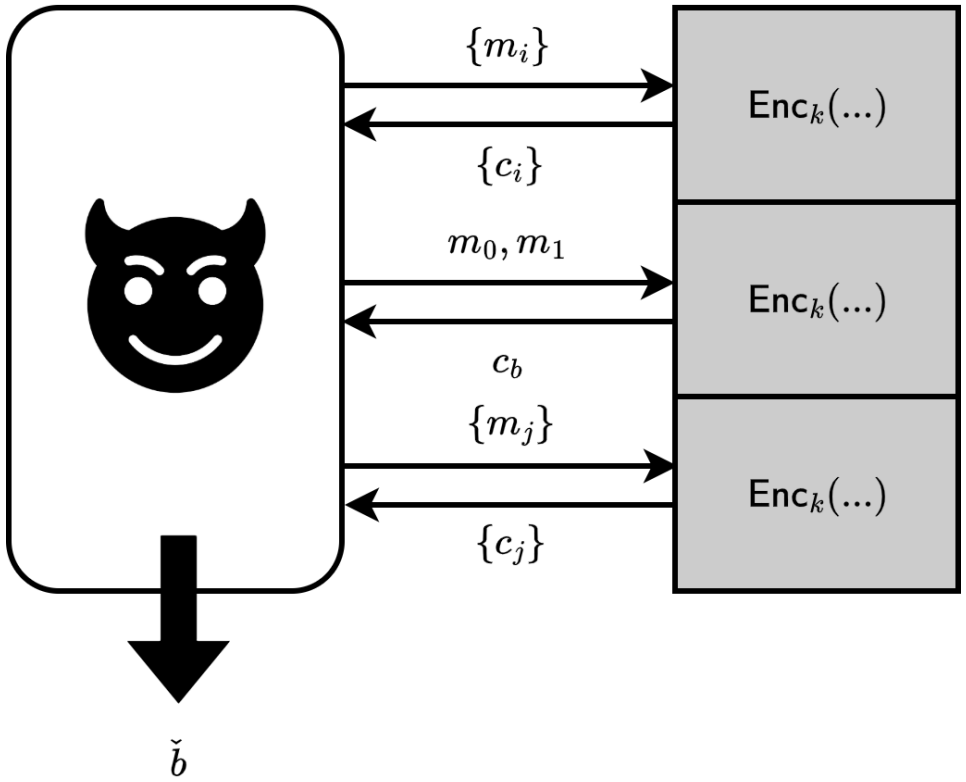
- CPA security {…}. How many queries can be made? $q = \text{poly}(n)$



- An encryption scheme $\text{SE} = (\text{Gen}, \text{Enc}, \text{Dec})$ is secure against CPA attacks if ∀ polytime Adv, $\Pr\left(\text{Exp}^{\text{cpa}}_{\text{SE}}(n) = 1\right) \leq \frac{1}{2} + \text{negl}(n)$

Security against Chosen Plaintext Attacks (CPA)

- CPA security {…}. How many queries can be made? $q = \text{poly}(n)$

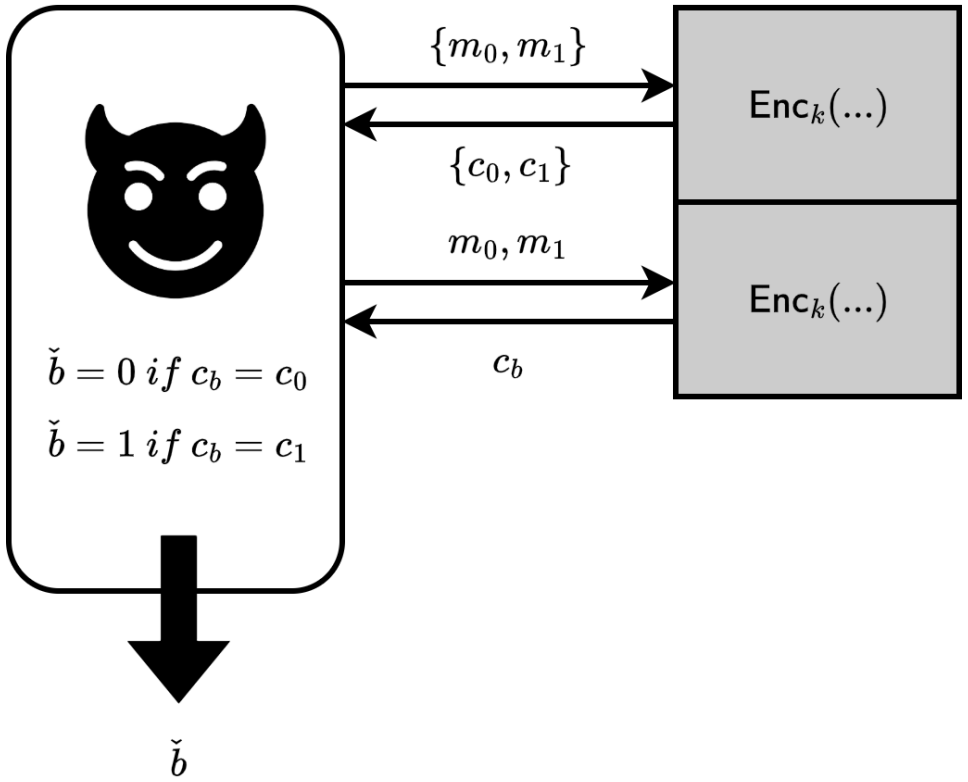


What if $\text{Enc}_k(m)$ always outputs the same ciphertext c?

- An encryption scheme $\text{SE} = (\text{Gen}, \text{Enc}, \text{Dec})$ is secure against CPA attacks if ∀ polytime Adv, $\Pr\left(\text{Exp}^{\text{cpa}}_{\text{SE}}(n) = 1\right) \leq \frac{1}{2} + \text{negl}(n)$

Security against Chosen Plaintext Attacks (CPA)

- CPA security {…}. How many queries can be made? $q = \text{poly}(n)$



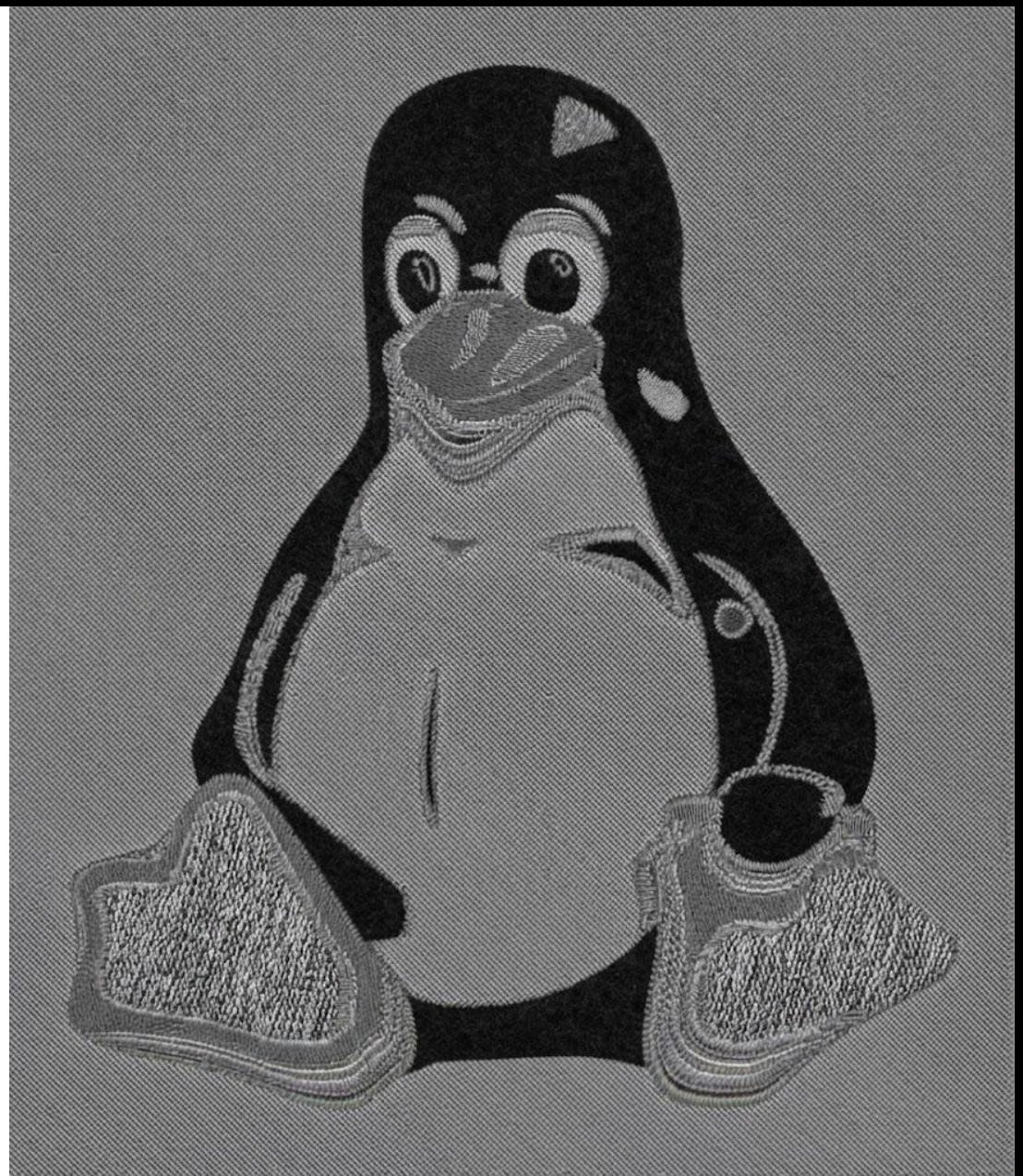
What if $\text{Enc}_k(m)$ always outputs the same ciphertext c?

⇒ CPA security is only possible if $\text{Enc}_k(m)$ is "randomized"

- An encryption scheme $\text{SE} = (\text{Gen}, \text{Enc}, \text{Dec})$ is secure against CPA attacks if ∀ polytime Adv, $\Pr\left(\text{Exp}^{\text{cpa}}_{\text{SE}}(n) = 1\right) \leq \frac{1}{2} + \text{negl}(n)$

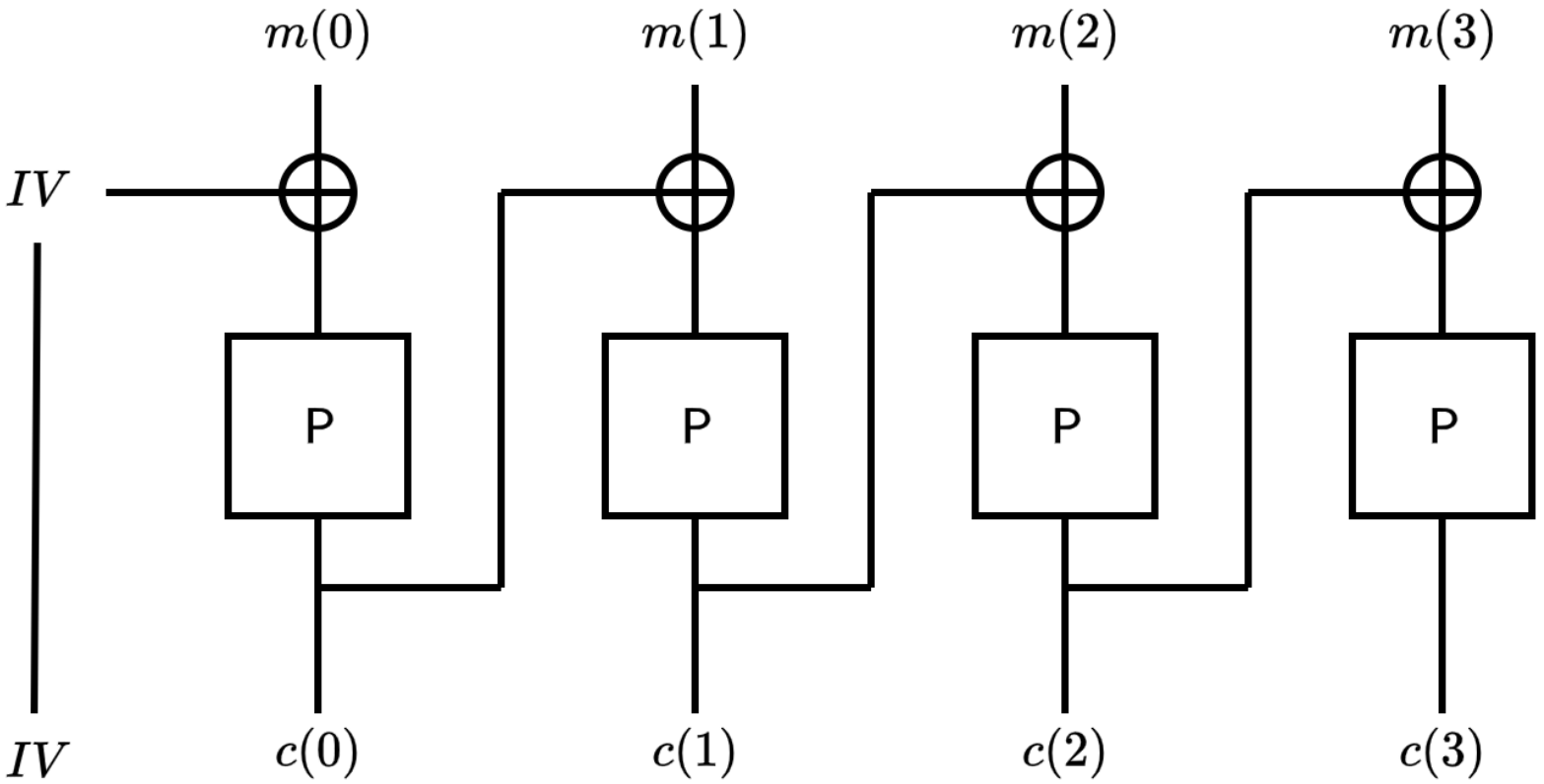
Does it matter?





Cipher Block Chaining (CBC) encryption

- If P is a $(t, \epsilon_{\mathsf{P}})$ PRP and IV is random, CBC is $(t - q \cdot l \cdot \tau_{\mathsf{P}}, \epsilon_{\mathrm{cpa}})$-secure with $\epsilon_{\mathrm{cpa}} \leq 2 \cdot \epsilon_{\mathsf{P}} + \frac{(q \cdot l)^2}{2^n}$ and τ_{P} the permutation's execution time

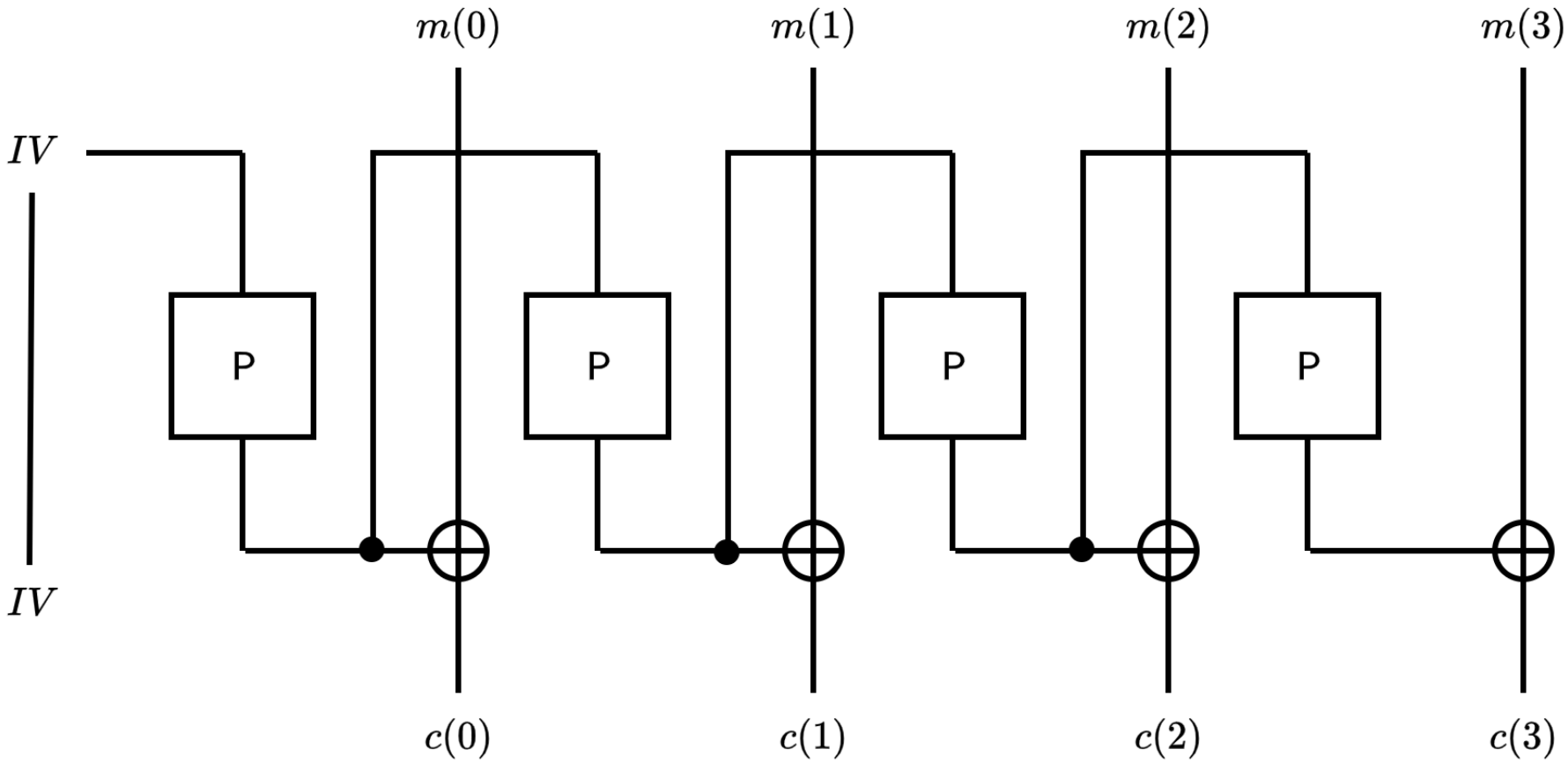


Is the bound tight {...}? (e.g., say $q = 1$)

What happens if IV is fixed? {...}

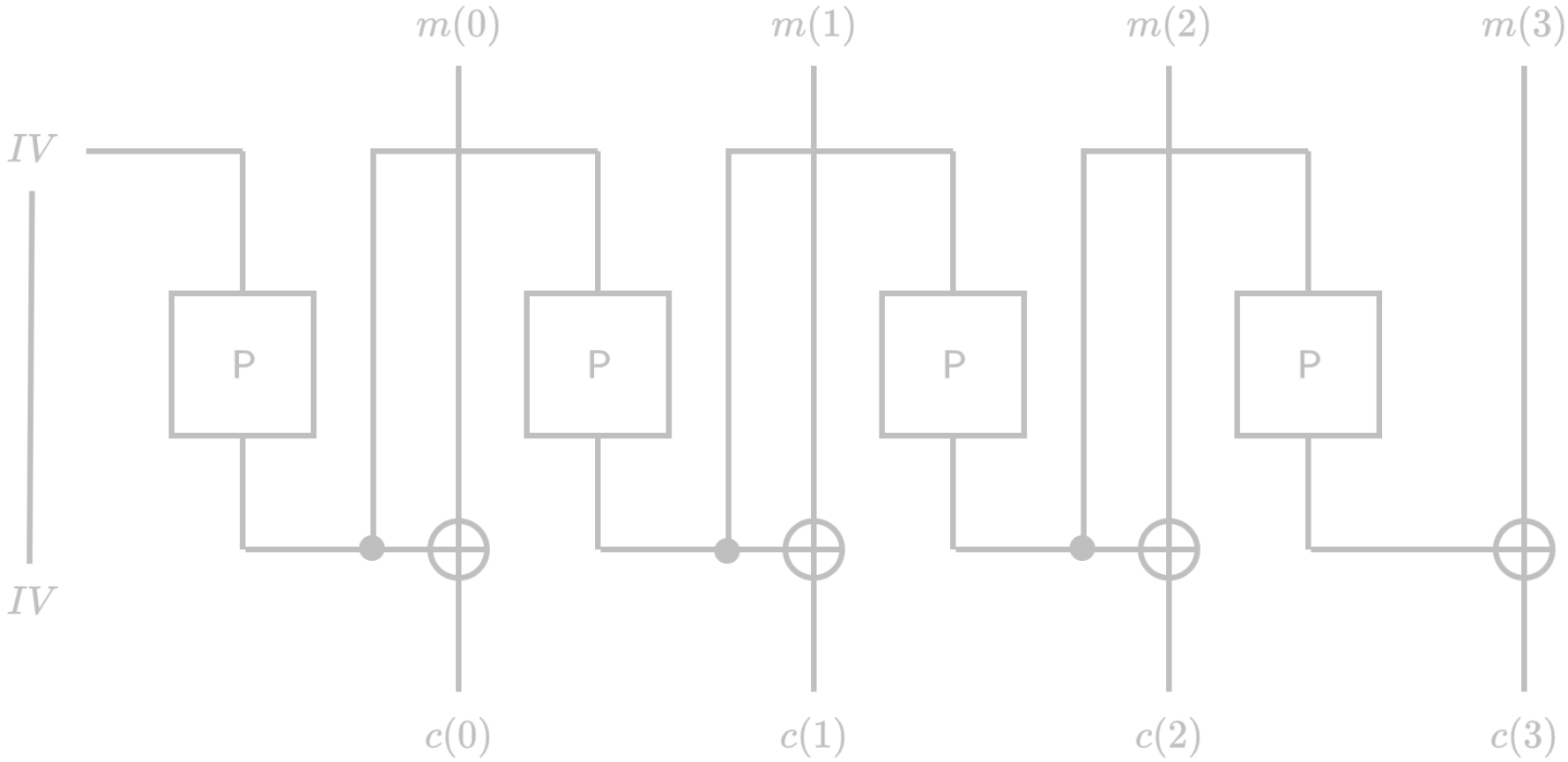
Fixed IV caveats

- Question: is it better or worse with the OFB mode below? {…}



Fixed IV caveats

- Question: is it better or worse with the OFB mode below? {…}



⇒ Can we weaken this requirement (see authenticated encryption)

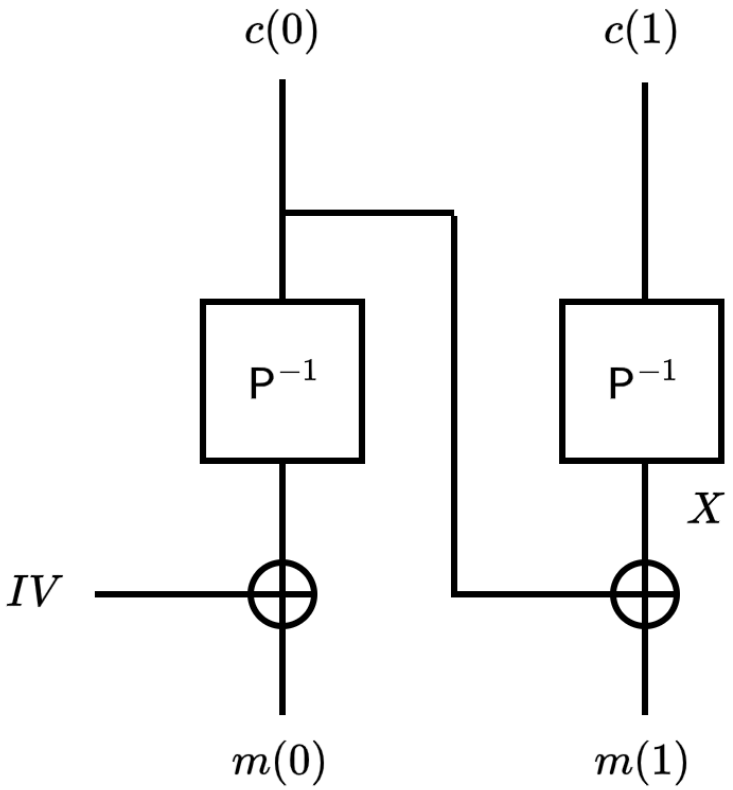
Message padding (PKCS #7 standard)

- Say the permutation P operates on 16 byte, if the last block
  - Is full: add one block $16_{\text{dec}}||16_{\text{dec}}||16_{\text{dec}}||\dots||16_{\text{dec}}$
  - Has one byte b_0: pad as $b_0||\ 15_{\text{dec}}||\ 15_{\text{dec}}||\dots||15_{\text{dec}}$
  - Has two bytes b_0, b_1: pad as $b_0||b_1||14_{\text{dec}}||\ 14_{\text{dec}}||\dots||14_{\text{dec}}$
  - Same rule up to fifteen more bytes

- Allows encryption messages of any (byte) length
- Decryption then checks that the last block is well padded

Padding oracle attacks

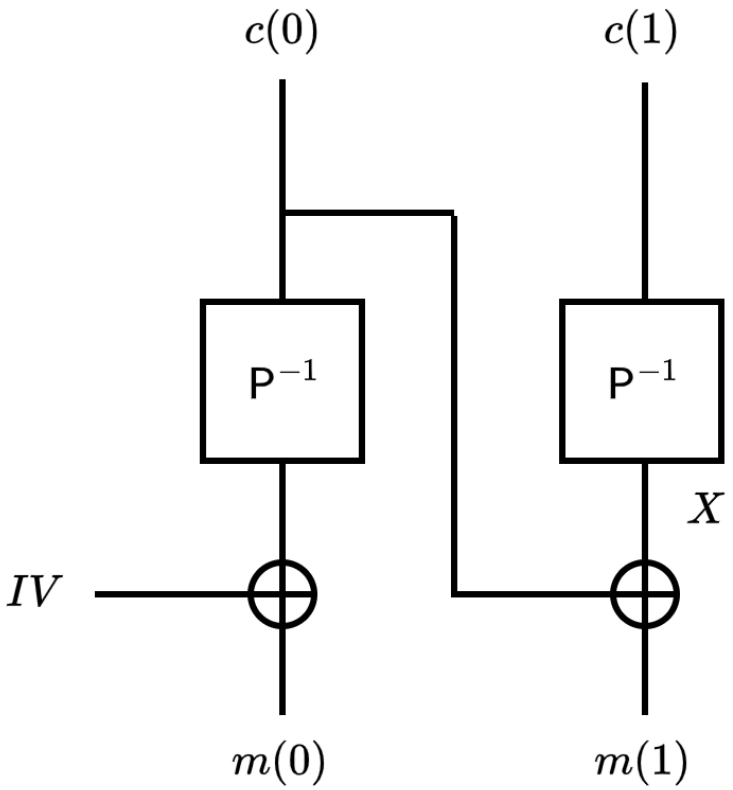
- Say Adv wants to decrypt c_1 and can learn whether p_1 is valid



How to proceed?

Padding oracle attacks

- Say Adv wants to decrypt c_1 and can learn whether p_1 is valid

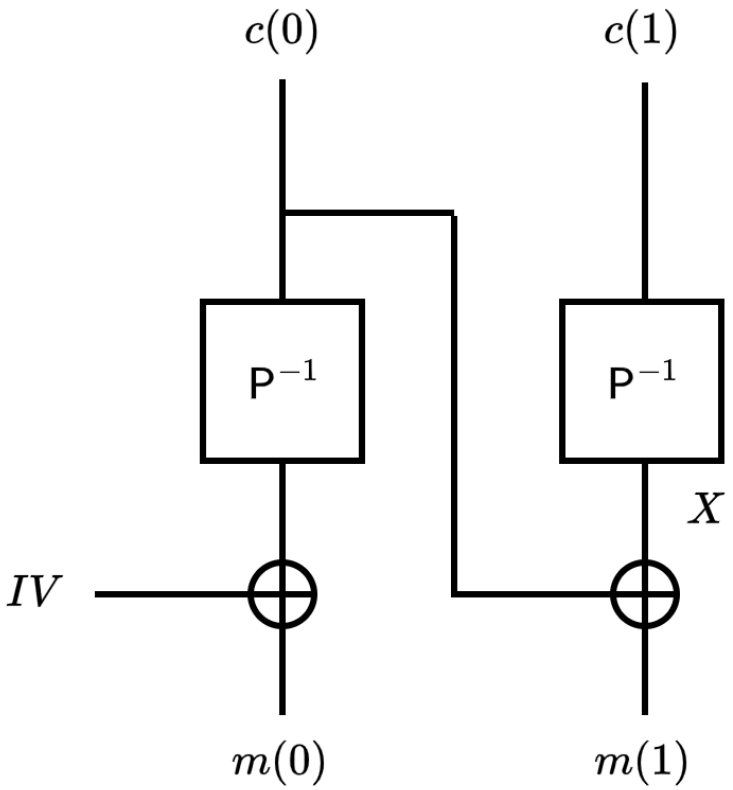


How to proceed?

- Pick a random c_0, vary last byte until it is accepted $\Rightarrow c_0(15) \oplus X(15) = 1_{\text{dec}}$
  - Finds $X(15)$ with trials on average

Padding oracle attacks

- Say Adv wants to decrypt c_1 and can learn whether p_1 is valid

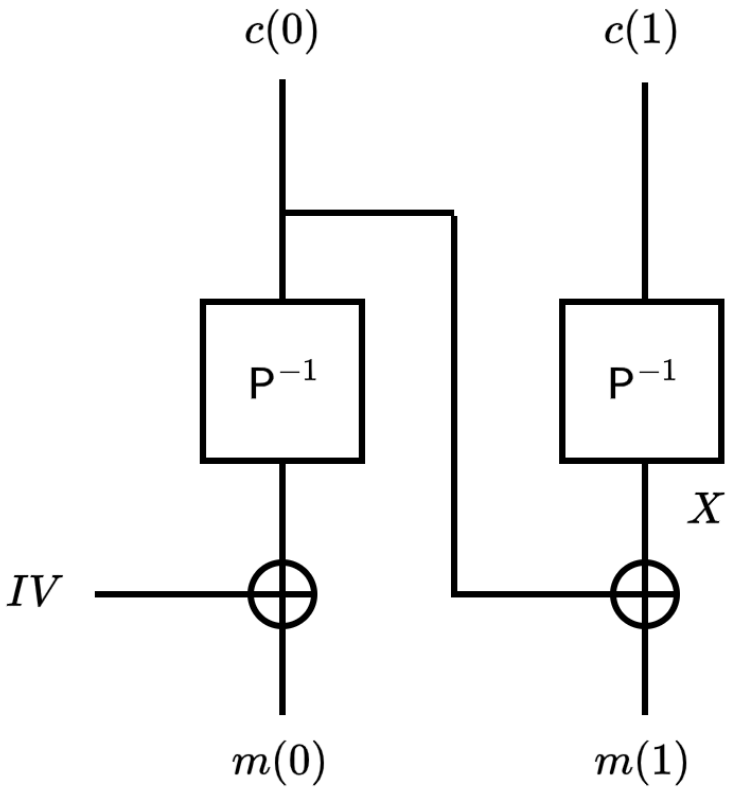


How to proceed?

- Pick a random c_0, vary last byte until it is accepted $\Rightarrow c_0(15) \oplus X(15) = 1_{\text{dec}}$
  - Finds $X(15)$ with 128 trials on average

Padding oracle attacks

- Say Adv wants to decrypt c_1 and can learn whether p_1 is valid

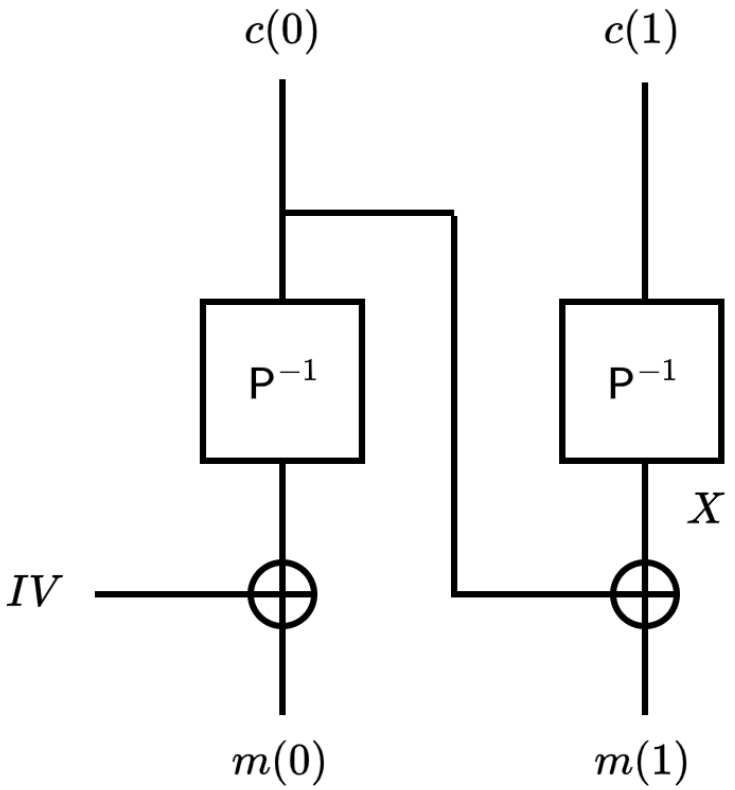


How to proceed?

- Pick a random c_0, vary last byte until it is accepted $\Rightarrow c_0(15) \oplus X(15) = 1_{\text{dec}}$
  - Finds $X(15)$ with 128 trials on average
- Set $c_0(15)$ to $X(15) \oplus 2_{\text{dec}}$, vary $c_0(14)$ until it is accepted $\Rightarrow c_0(14) \oplus X(14) = 2_{\text{dec}}$

Padding oracle attacks

- Say Adv wants to decrypt c_1 and can learn whether p_1 is valid



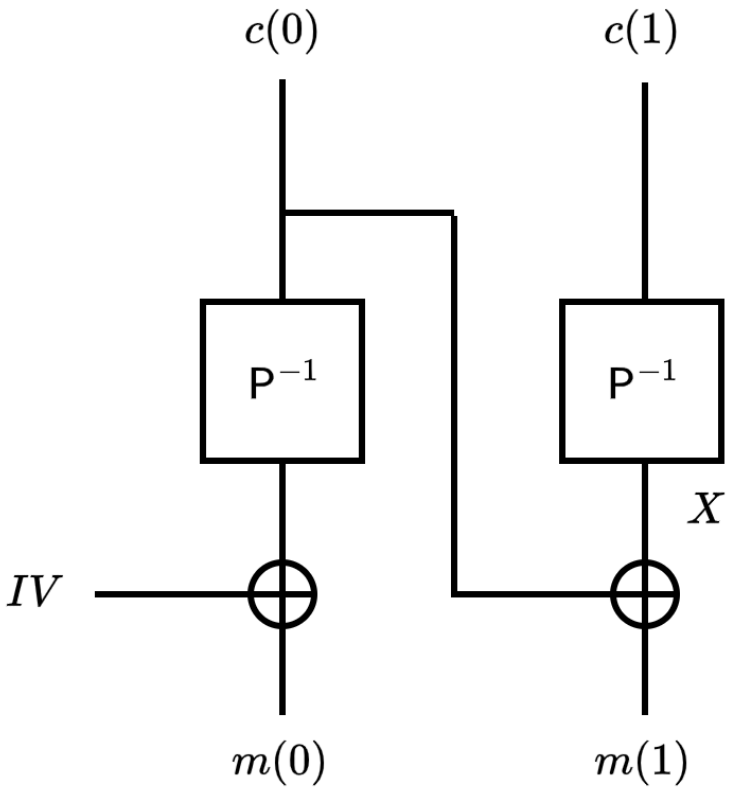
How to proceed?

- Pick a random c_0, vary last byte until it is accepted $\Rightarrow c_0(15) \oplus X(15) = 1_{\text{dec}}$
  - Finds $X(15)$ with 128 trials on average
- Set $c_0(15)$ to $X(15) \oplus 2_{\text{dec}}$, vary $c_0(14)$ until it is accepted $\Rightarrow c_0(14) \oplus X(14) = 2_{\text{dec}}$

$\Rightarrow$ Finds full X with 16 x 128 trials on average

Padding oracle attacks

- Say Adv wants to decrypt c_1 and can learn whether p_1 is valid



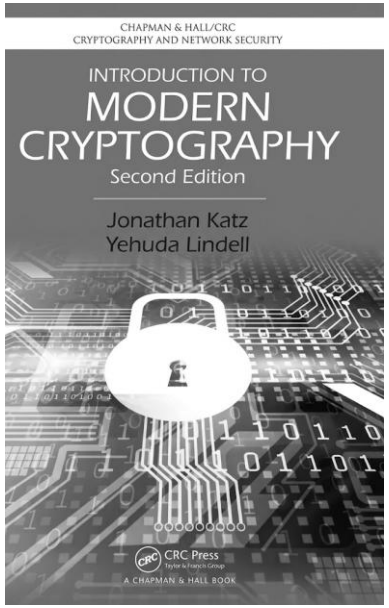
How to proceed?

- Pick a random c_0, vary last byte until it is accepted $\Rightarrow c_0(15) \oplus X(15) = 1_{\text{dec}}$
  - Finds $X(15)$ with 128 trials on average
- Set $c_0(15)$ to $X(15) \oplus 2_{\text{dec}}$, vary $c_0(14)$ until it is accepted $\Rightarrow c_0(14) \oplus X(14) = 2_{\text{dec}}$

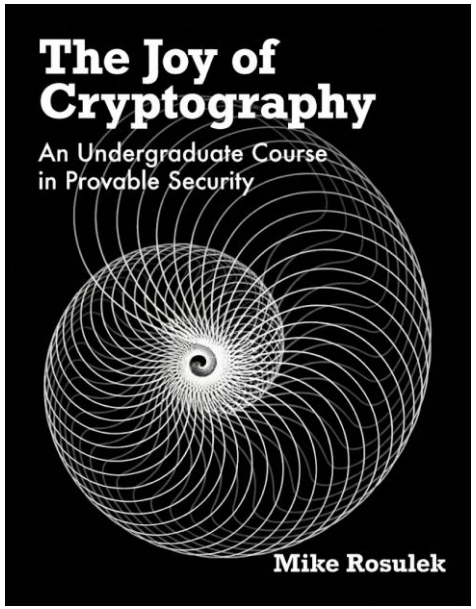
$\Rightarrow$ Finds full X with 16 x 128 trials on average

- Chosen Ciphertext Attacks (CCA) can prevented with a MAC

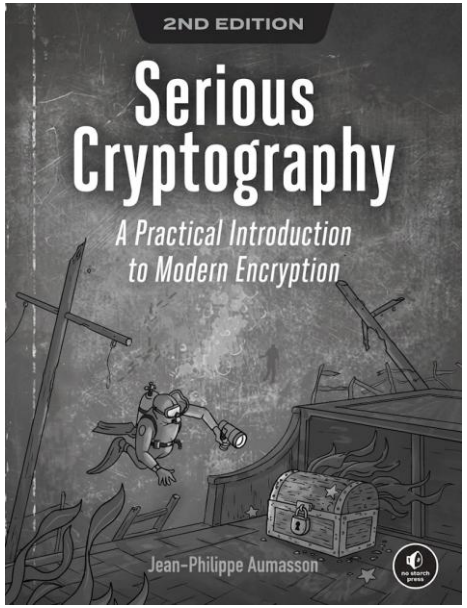
References



(a)



(b)



(c)

This lecture mostly follows Chapter 3 of Reference (a). The security definitions, cryptographic primitives and modes of operation we discuss are detailed there (with a number of additional concepts and examples that we skipped for conciseness purposes). Reference (b) provides an alternative view to these topics. Reference (c) provides discusses these topics in a less technical manner and provides a simple explanation and discussion of padding oracle attacks.

Questions?