Modern symmetric-key Encryption

Citation

- I would like to thank Claude Crepeau for allowing me to use his slide from his crypto course to mount my course. Some of these slides are taken directly from his course.

- Comp 547 at Mcgill university

Overview of sec

- The Concrete Approach
- The Asymptotic Approach
- Defining Computationally-Secure Encryption
  - The Basic Definition of Security
- Constructing Secure Encryption Schemes
    - Pseudorandom Generators
    - Proofs by Reduction
    - Fixed-Length Encryption Scheme
- Stronger Security Notions
  - Security for Multiple Encryptions
  - Chosen-Plaintext Attacks and CPA-Security

Computational Security

- What does it mean to be pseudo-random
  - Things can look random when they are not

- This can be used to achieve secure encryption while using short keys

Computational Security

- Encrypt many messages using short keys
- Limitations of perfect secrecy can be bypassed
- We can achieve a strong but necessarily weaker notion than perfect secrecy

Computational approach to secure encryption

- A computation encryption scheme can be broken given enough time
  - Try all the keys until you find the right one
  - Guess keys until you find the right one

- Under certain assumptions, it should take millions of years to break an encryption scheme even given all the (current and future) computation power available on earth

Weakening of security

- The computational approach incorporates two relaxations of the notion of perfect security
  - Security is only preserved against efficient adversaries that run in a feasible amount of time
  - Adversaries can potentially succeed with some very small probability.

Concrete security

- The concrete approach quantifies the security of a cryptographic scheme by bounding the maximum success probability of any adversary running for at most some fixed amount of time.

- That is, let t,ε be positive constants with $\varepsilon \leq 1$.

  A scheme is (t,ε)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ε.

Concrete security

- Modern private-key encryption schemes are generally assumed to give almost optimal security in the following sense:
  - When the key has length s, an adversary running in time t can succeed in breaking the scheme with probability at most (c is small)

$$c * \frac{t}{2^s}$$

Asymptotic security

- An algorithm
  - Takes a parameter n
  - Use random coins
- The success probability of an algorithm is the probability that it produces the correct output
- The running time and success probability of an algorithm are all viewed as functions of n.

Algorithm running time and success probability.

- Running time
  - The running time of an algorithm is how many steps it takes until it stops
- An algorithm is efficient if the algorithm runs in polynomial time
  - An algorithm is polynomial time if there exists a constant c,d such that the running time of algorithm is less than c$\cdot$ n^d .
- An algorithm has small probability of success if the probability that the algorithm succeeds is negligible in n.

Negligible function

- A function μ: $\mathbb{N} \rightarrow R^{+}$ is negligible if
  - Computer science definition:

$$\forall\, c \in N \;: \mu(n) \in O(n^{-c})$$

  - Math definition:

$$\forall\, c \in N : \exists\, n' \in N \;\; \forall n \geq n'$$

$$\mu(n) \leq \frac{1}{n^{c}}$$

- An algorithm has small probability of success if the probability that the algorithm succeeds is negligible in n.

- Class of negligible functions is closed under addition and multiplication

Security

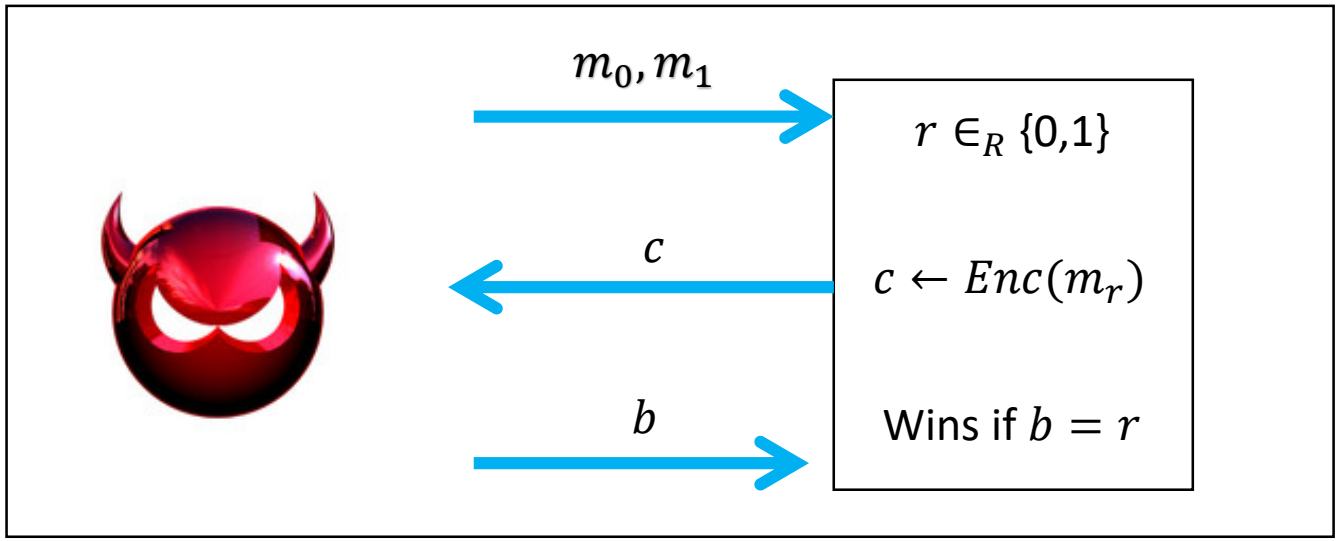
- A scheme is secure if:

  Every Probabilistic Polynomial Time Adversary (viewed as an algorithm) succeeds in breaking the scheme with only negligible probability.

Warning

- Negligible probability might be large for small values
  - Example: f(n) = $\min\left(2^{\{n-128\}}, 1\right)$

Secure encryption scheme (in terms of game)



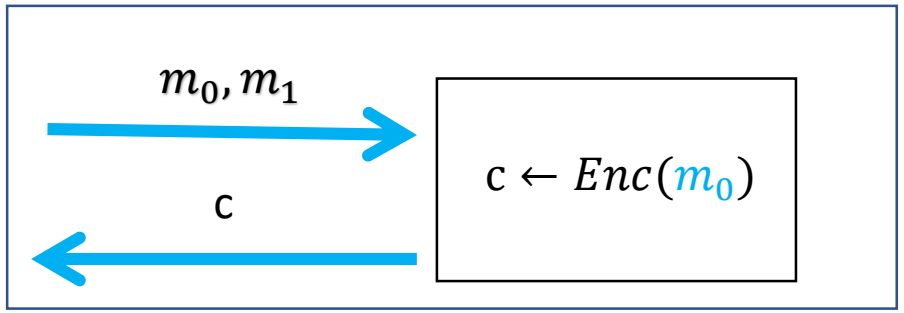
G

An encryption scheme is secure

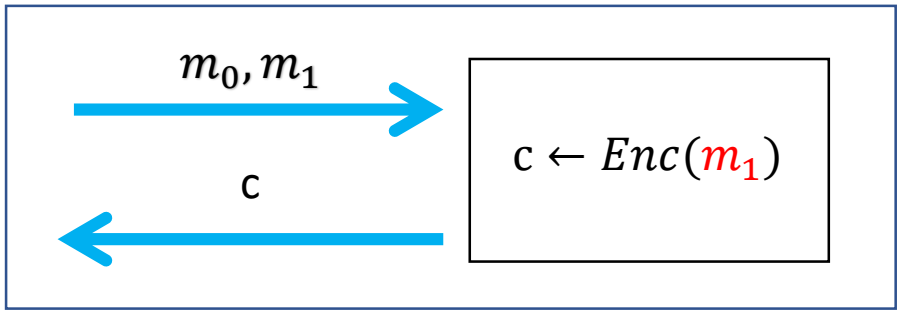
$$\forall A \in PPT \;\; \exists\, \mu \in negl : \Pr[A(G) = win] \leq \frac{1}{2} + \mu(n)$$

Every PPT adversary does only negligibly better than guessing.

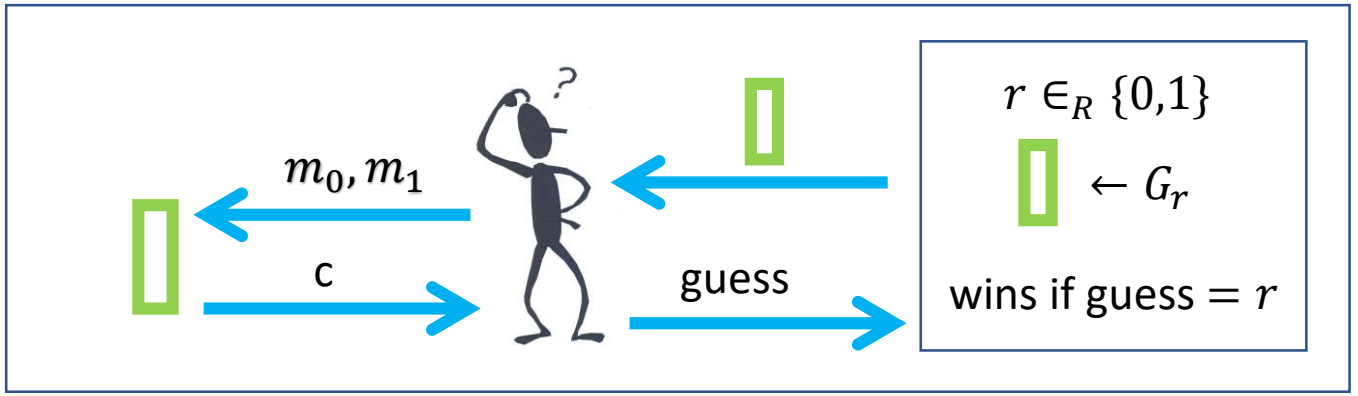
Encryption game



G_0



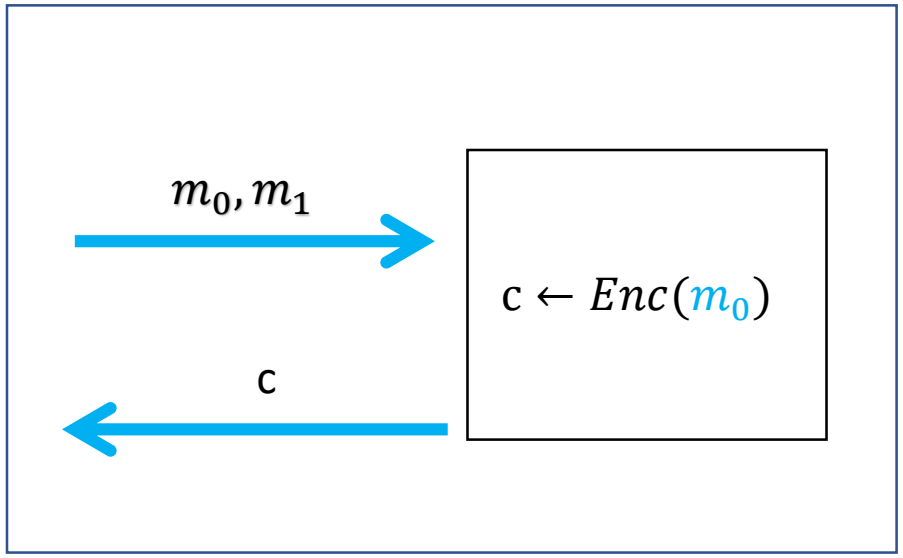
G_1



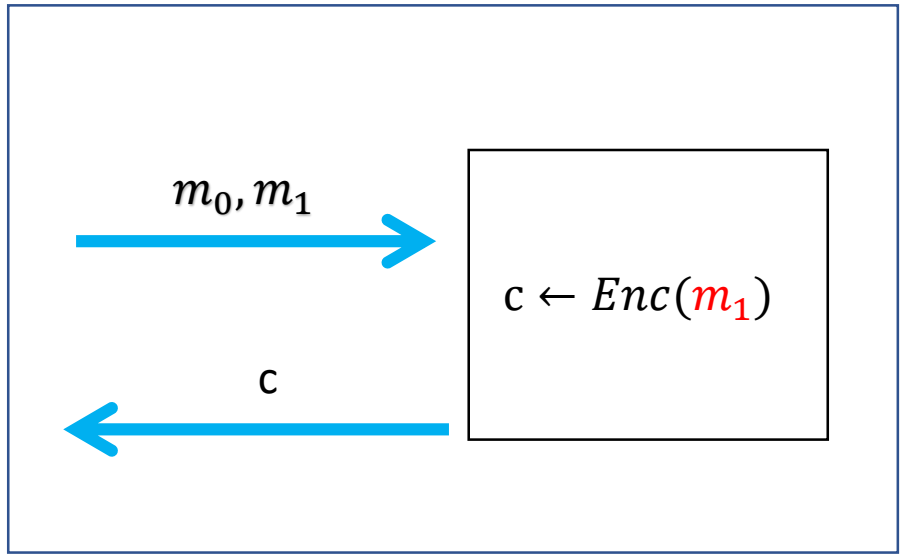
An encryption scheme is secure

$$\forall D \in PPT \ \exists \mu \in negl : \Pr[A(G) = win] \leq \frac{1}{2} + \mu(n)$$

Encryption game



G_0



G_1

An encryption scheme is secure
if a distinguisher cannot guess which of these two games
he is playing with more than one-half plus negligible probability

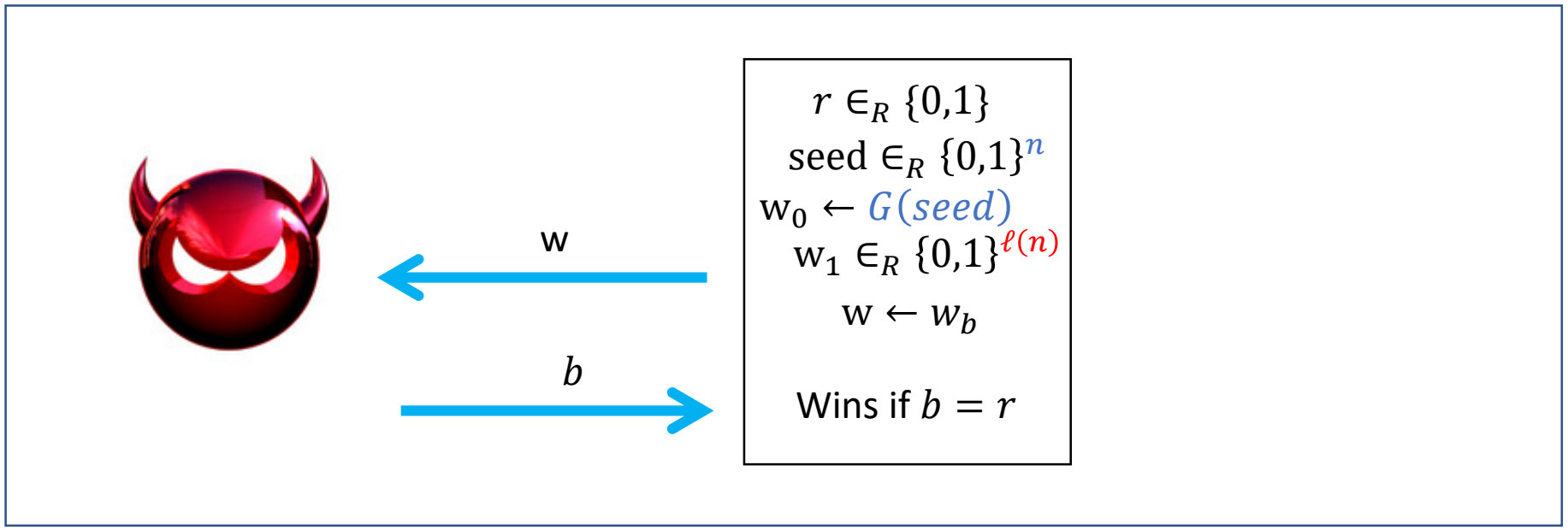
Computational indistinguishability

- Two games G_0, G_1 (parameterized by n) are computationally indistinguishable if
  - For all PPT distinguisher, there exists a negligible function $\mu(n)$ such that when the distinguisher is given G sampled from G_0, G_1 at random, the probability that he correctly guesses which game he was given is at most

$$1/2 + \mu(n)$$

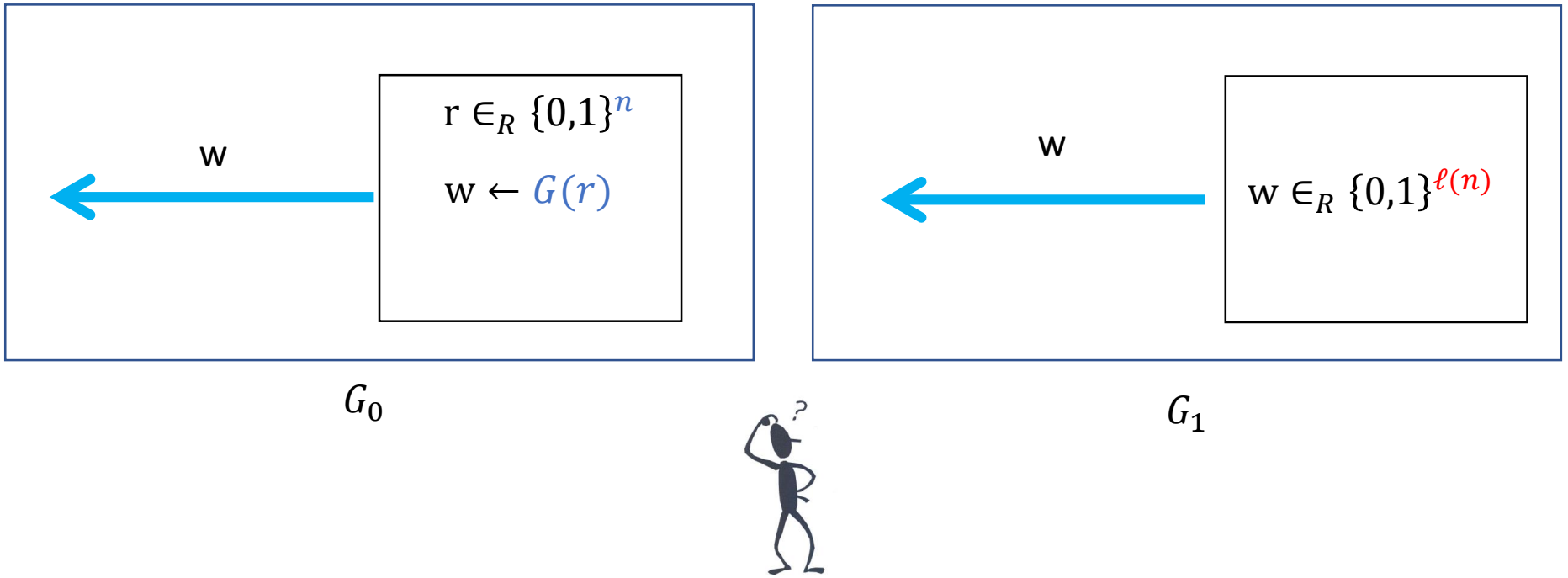
Definition of Pseudo-random generator

- A function $G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ is a PRG if
  - Expansion: $\ell(n) > n$ (trivial if $\ell(n) \leq n$)
  - Pseudo-random: $\Pr[Adv\ wins\ the\ following\ game] \leq \frac{1}{2} + \mu(n)$

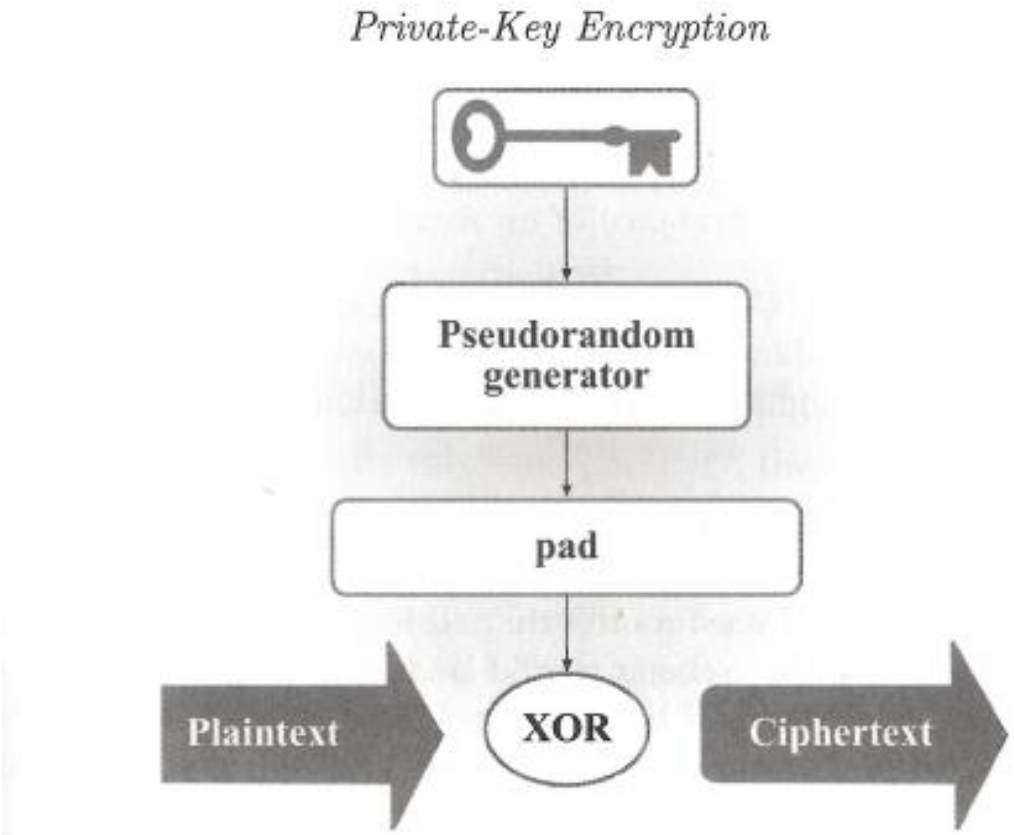


Definition of Pseudo-random generator

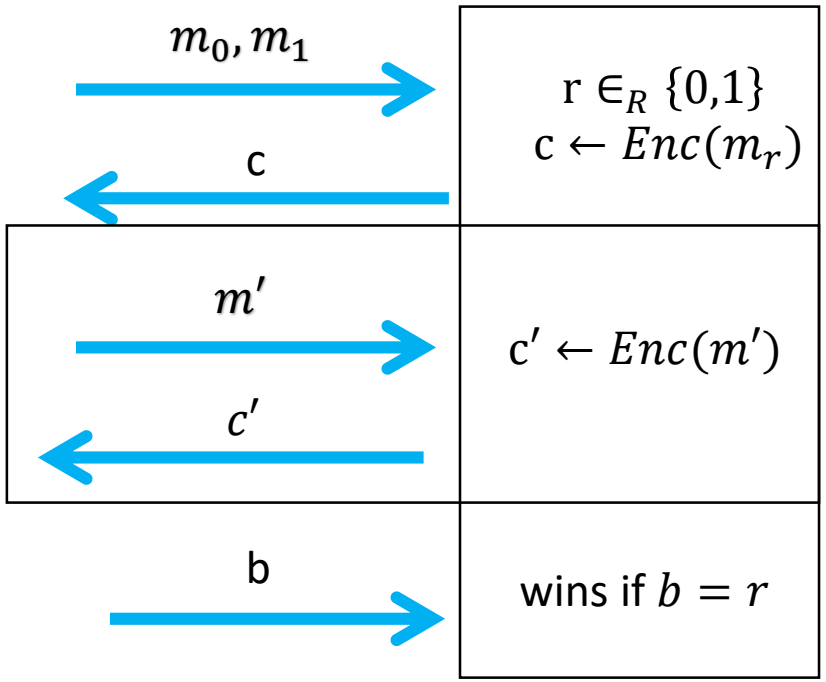
- A function $G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ is a PRG if
  - Expansion: $\ell(n) > n$ (trivial if $\ell(n) \leq n$)
  - Pseudo-random: following two games are computationally indistinguishable



Encrypting a message from a short key using a pseudo-random generator



CPA-secure



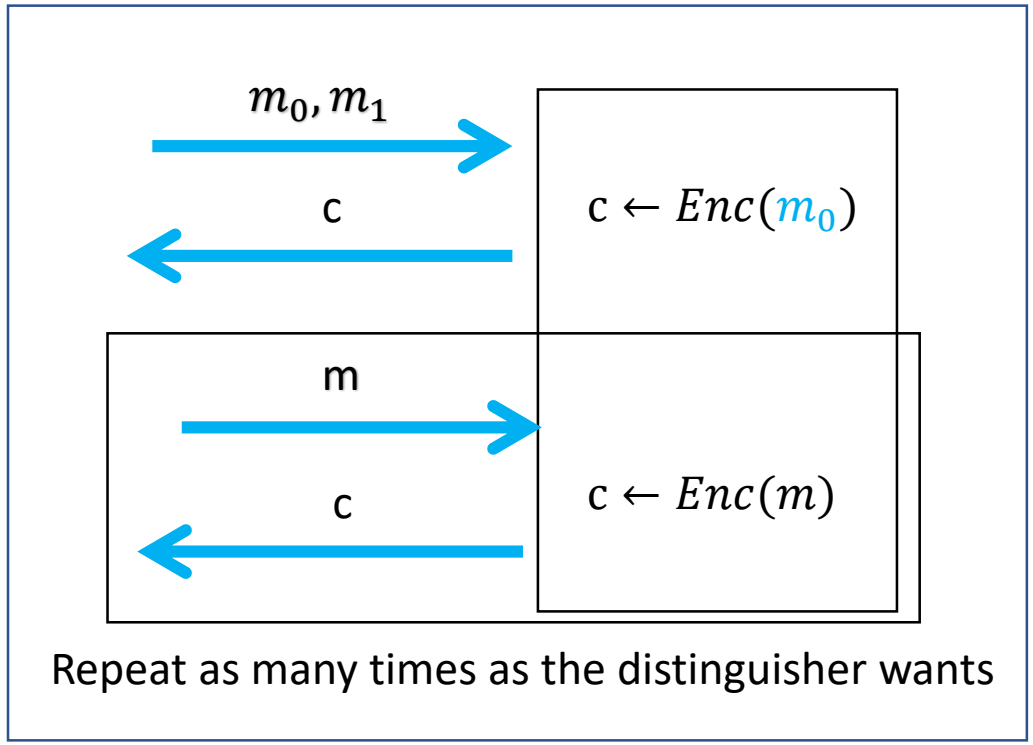
Repeat as many times as the adversary wants

An encryption scheme is secure

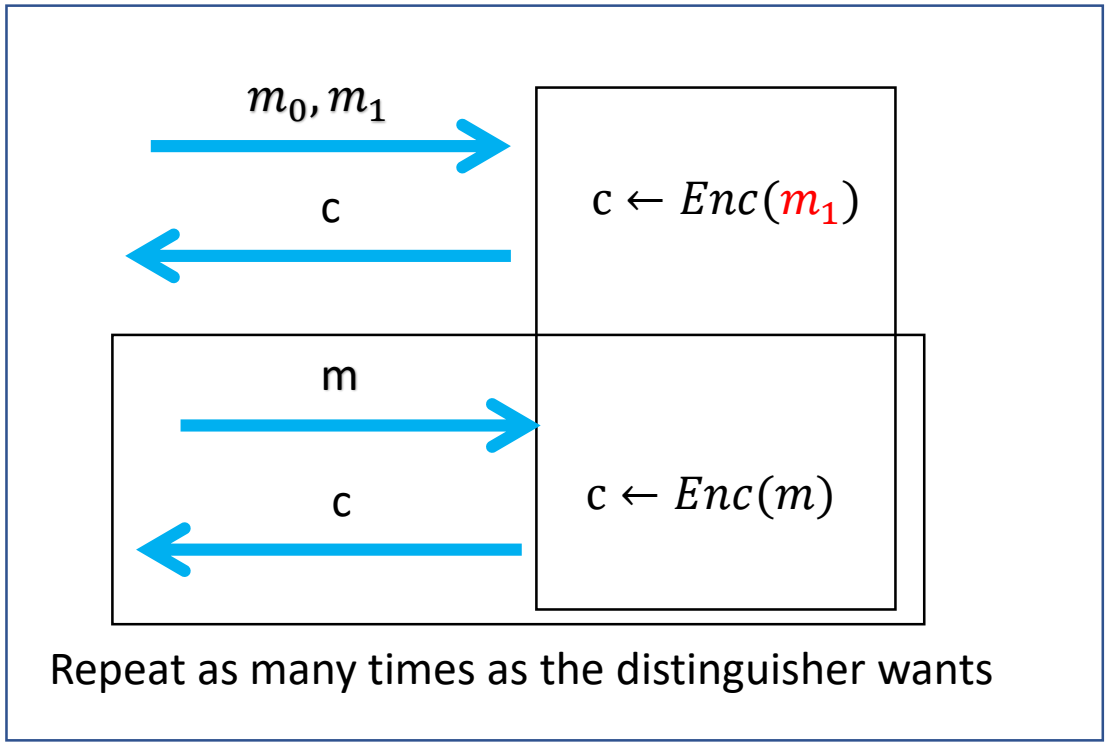
$$\forall A \in PPT \ \exists \mu \in negl : \Pr[A(G) = win] \leq \frac{1}{2} + \mu(n)$$

(every PPT adversary does only negligibly better than guessing.)

Chosen-plaintext security



G_0



G_1

Midway islands (non-CPA secure)

- American cryptanalysts thought: * = Midway Island
- Americans sent: “Midway is low on water”
- Japanese sent: “* blah blah”
- Americans confirmed that * = Midway Island
- Lesson: Adversaries can influence the message.

On the (in)security of deterministic encryption scheme

- An encryption scheme is deterministic

  Each plaintext maps to a unique ciphertext

- Can deterministic encryption scheme be CPA-secure?

  No!

  Encrypting the same plaintext twice results in the same ciphertext.

- Lesson: **Secure encryption requires randomness**

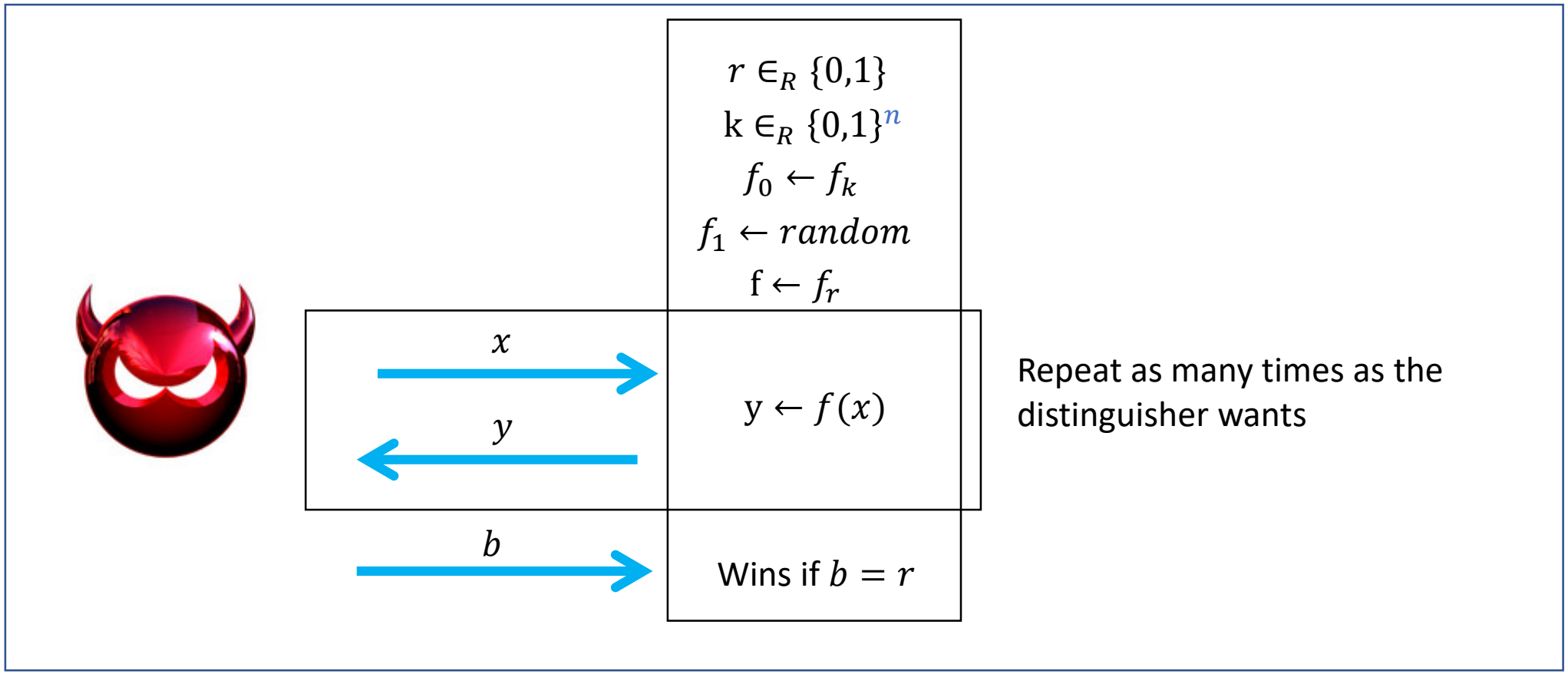
Definition of random function

- Consistency: if you query a random function with the same input, it will give you the same output

- Random: If you provide a new input to a random function, it will give you a random output

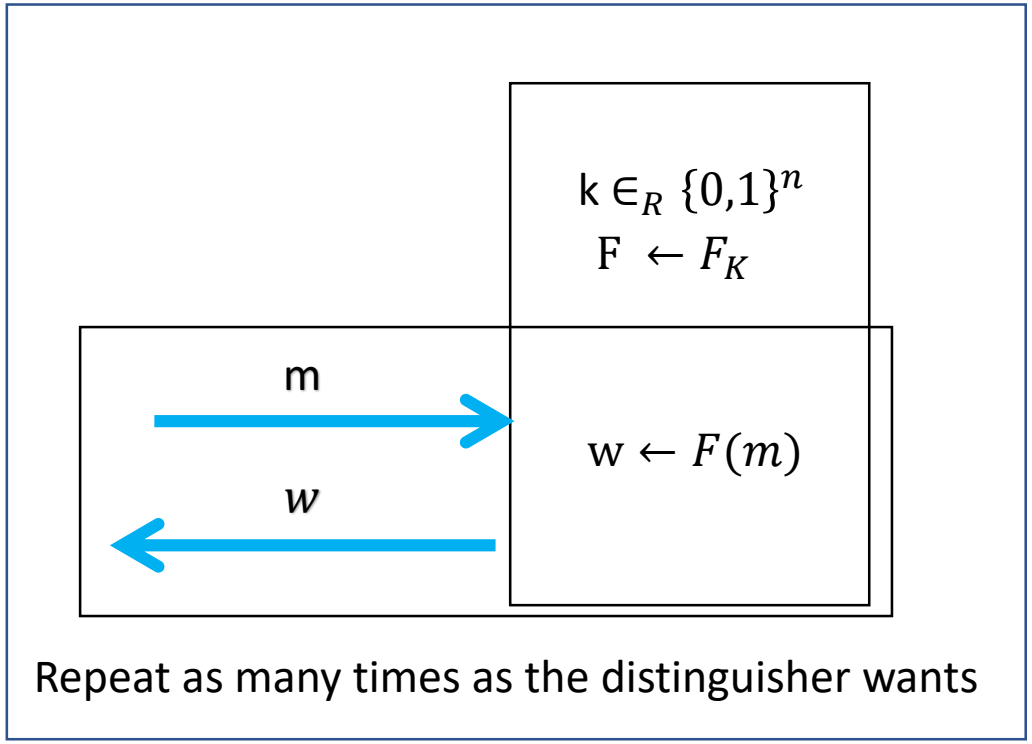
Pseudo-random function

A class of functions $(F_1, \dots, F_{2^n})$ if every PPT adversary wins the following game with probability $\frac{1}{2} + \mu(n)$ where $\mu \in negl$

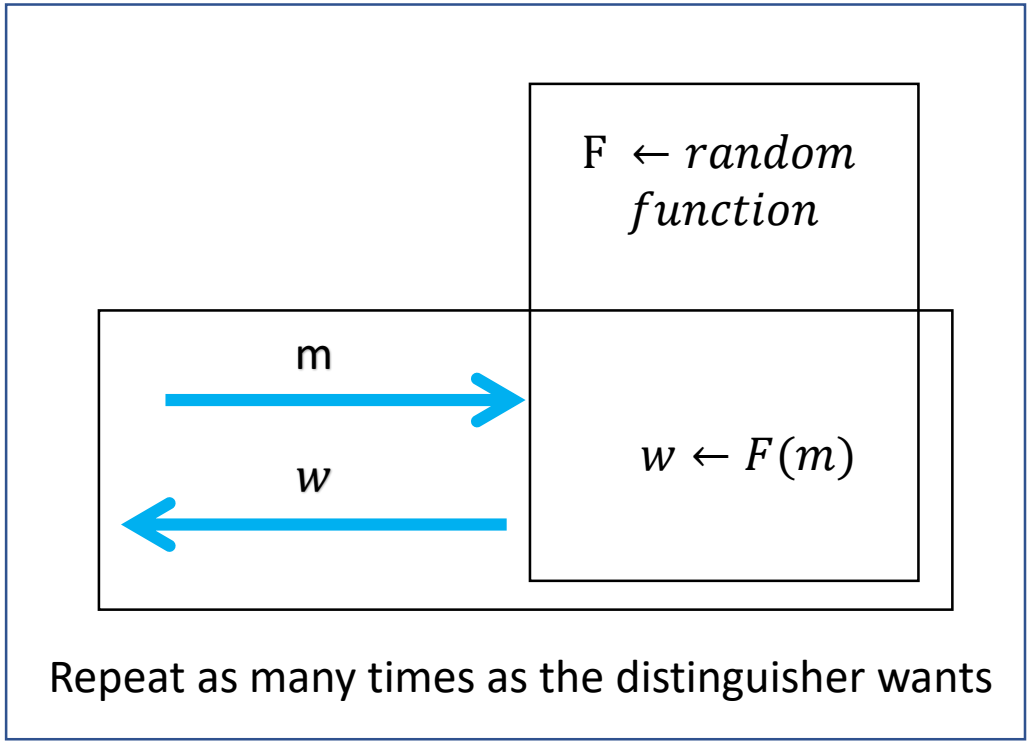


Pseudo-random function

- A class of functions $(F_1, \dots, F_{2^n})$ is pseudo-random if the following two games are indistinguishable



G_0

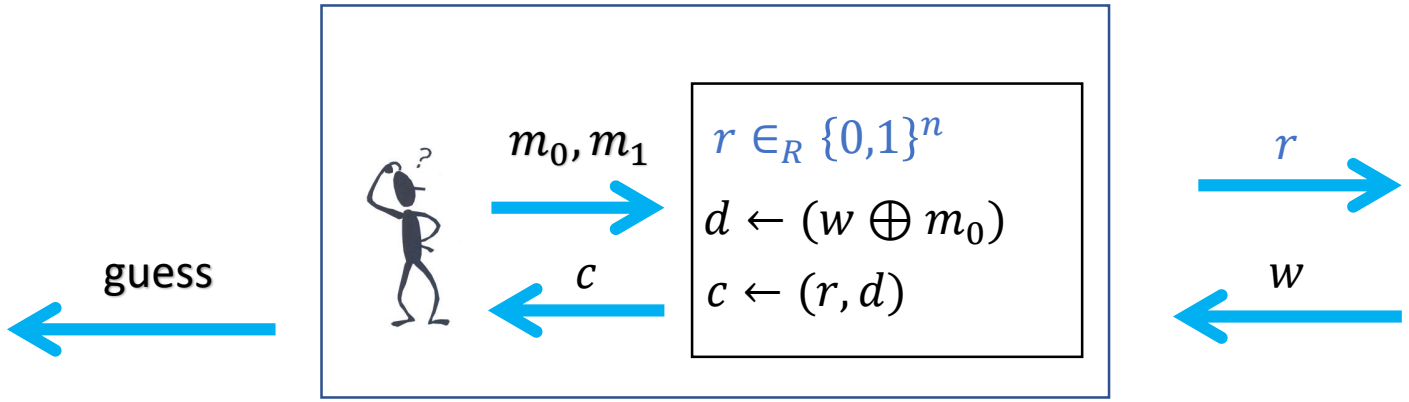


G_1

CPA-secure encryption scheme from PRF

- $Keygen(\{1\}^s)$
  - $k \in_R \{0,1\}^s$
- $Enc_k(m)$
  - $r \in_R \{0,1\}^n$
  - $c \leftarrow (r, F_k(r) \oplus m)$
- $Dec_k(c)$
  - $(r, d) \leftarrow c$
  - $m \leftarrow F_k(r) \oplus d$

Building a distinguisher for the PRF using a distinguisher for the encryption scheme



Building a distinguisher for the PRF using a distinguisher for the encryption scheme