Quiz 2 answers

Question I

- Construct a PRG G such that $G'(r) \coloneqq G(r) \oplus (r \,||\, r)$ is not a PRG

- $G(r' \,||\, b) \coloneqq \tilde{G}(r') \,||\, b$
  - $\tilde{G} : \{0,1\}^{n-1} \to \{0,1\}^{2n-1}$ is a PRG
  - $r' \in \{0,1\}^{n-1}$

- Last bit of $G'(r)$ is always zero

Question 1

- $G(r' \,||\, b) \coloneqq \tilde{G}(r') \,||\, b$
  - $\tilde{G} : \{0,1\}^{n-1} \to \{0,1\}^{2n-1}$ is a PRG
  - $r' \in \{0,1\}^{n-1}$
- $G(r' \,||\, b) \oplus (c \,||\, r' \,||\, b) \coloneqq (G'(r') \oplus (c \,||\, r')) \,||\, (b \oplus b)$
  - $c = r' \,||\, b$
- $(G'(r') \oplus (c \,||\, r')) \,||\, (b \oplus b) = (G'(r') \oplus (c \,||\, r')) \,||\, 0$
- Since the last bit is always zero clearly not random

Question 2

- Show that $G(r) \coloneqq F_r(r)$ is not a PRG
  - $F_k(x)$ is a PRF from $\{0,1\}^n \rightarrow \{0,1\}^{2n}$

- $F_k(x) \coloneqq if\ (x = k)\ then\ (k\ ||\ k)\ else\ F'_k(x)$
  - $F'_k(x)$ is a PRF

- $G(r) = F_r(r) = (r||r)$ which is clearly not random

Question 3

- Solve for $y = x^2 + 2x + 6 \; mod \; 2^n$
- If there is a solution output 0.
- Otherwise output 1.

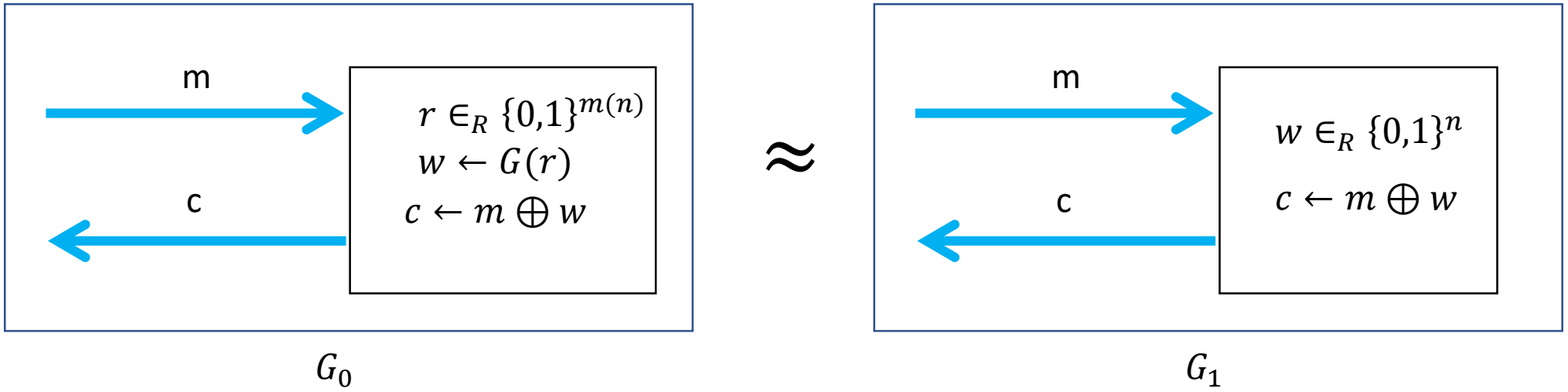
Question 4

- Distinguisher
  - If the game outputs 3 then output 1
  - Otherwise output 0

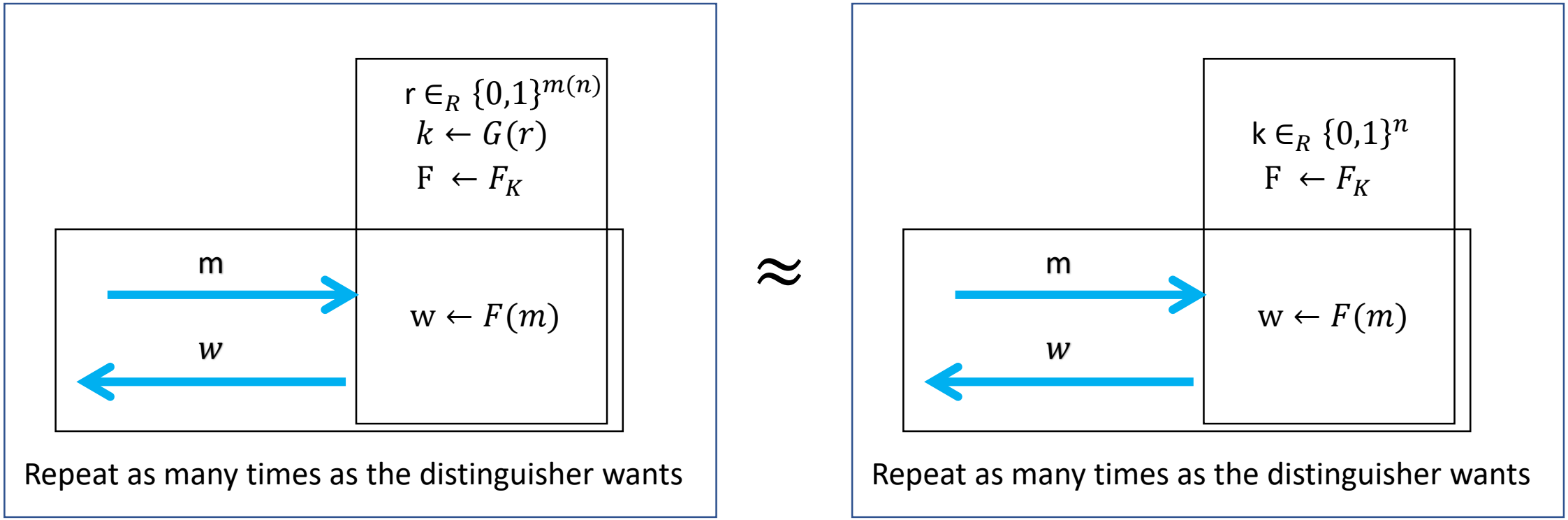
- $|\Pr[D(G_1) = 1] - \Pr[D(G_0) = 1]| = 1 - (1 - \frac{1}{2^{2n}})$

- Can’t do better

Show that encryption from PRG is secure

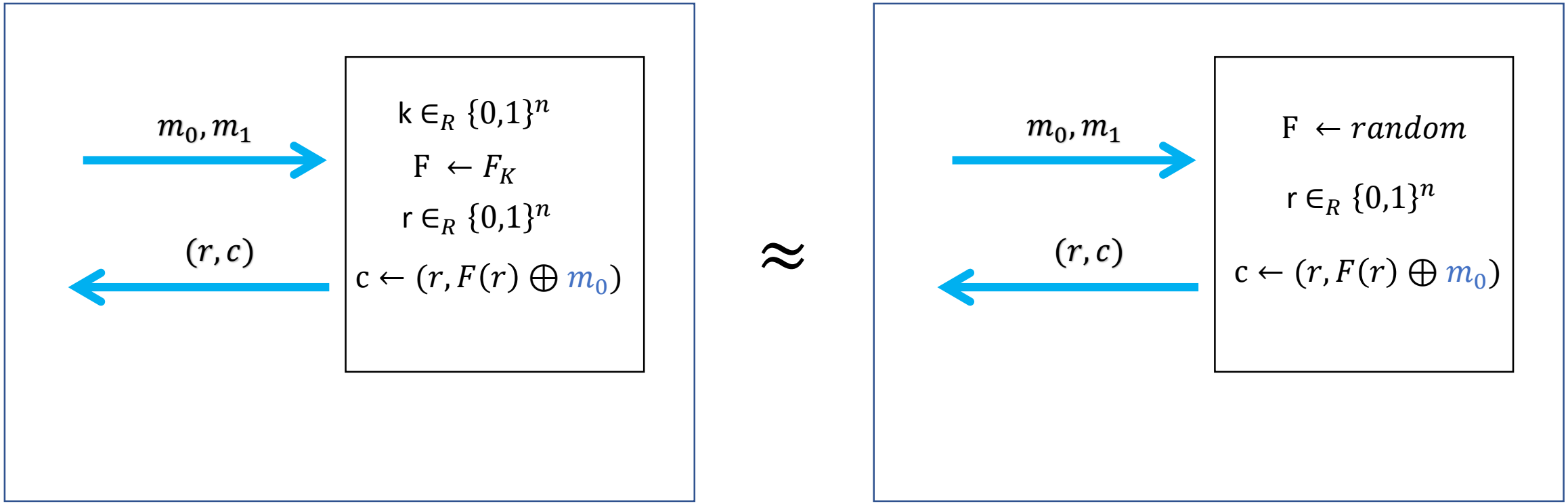


Show that $F_{G(k)}(x)$ is a PRF if G is a PRG



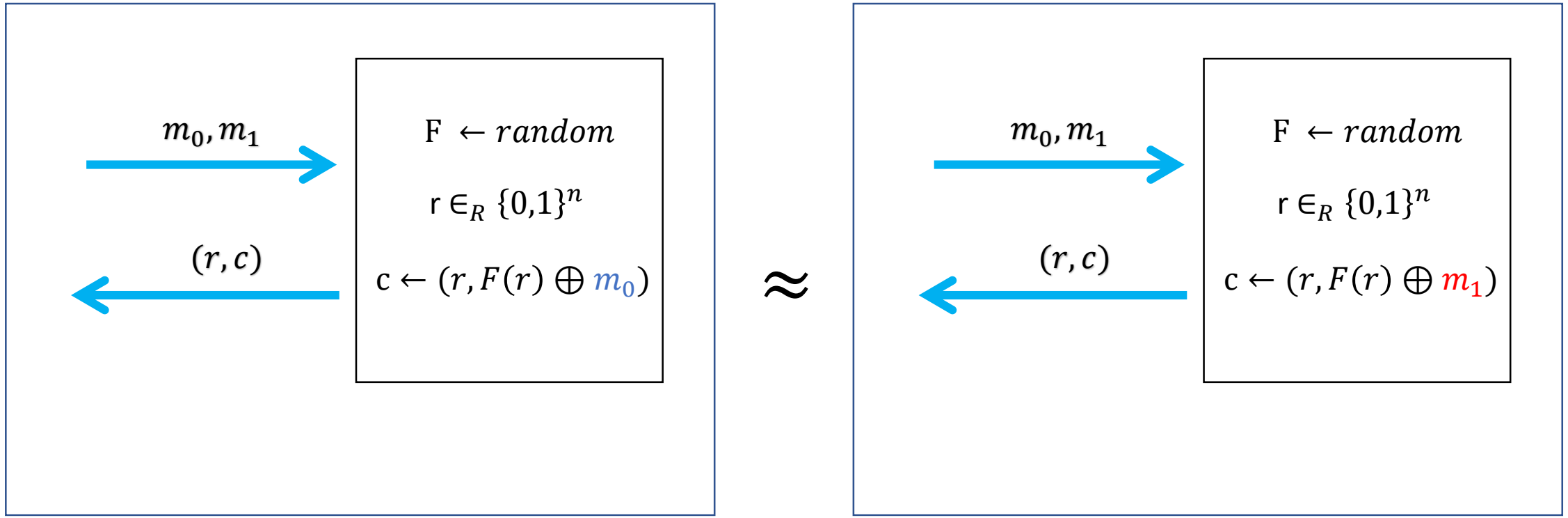
$Game_1$ $Game_2$

Show that PRF construction of encryption is secure



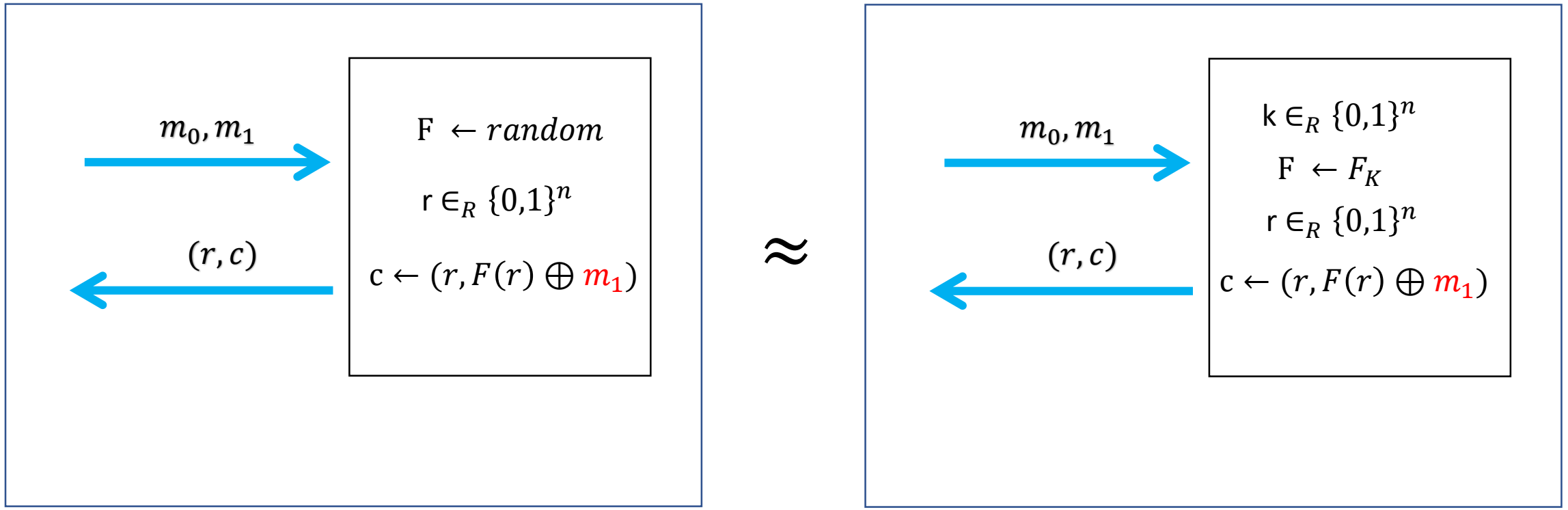
G_0 G_1

Show that PRF construction of encryption is secure



G_1 G_2

Show that PRF construction of encryption is secure

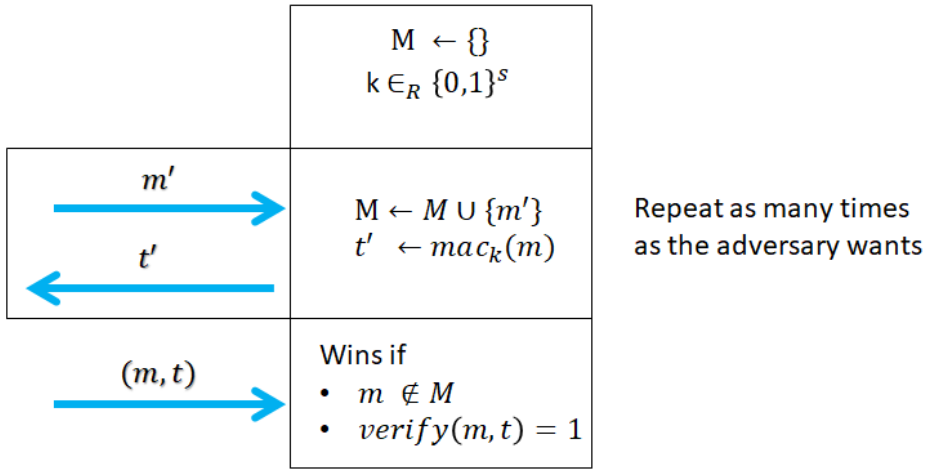


G_2 G_0

MAC forgery game

- Allow the adversary to learn tags for as many message as he wants

- A mac scheme is secure if
  - $\Pr[adv\ wins\ the\ forgery\ game] \leq negl(s)$



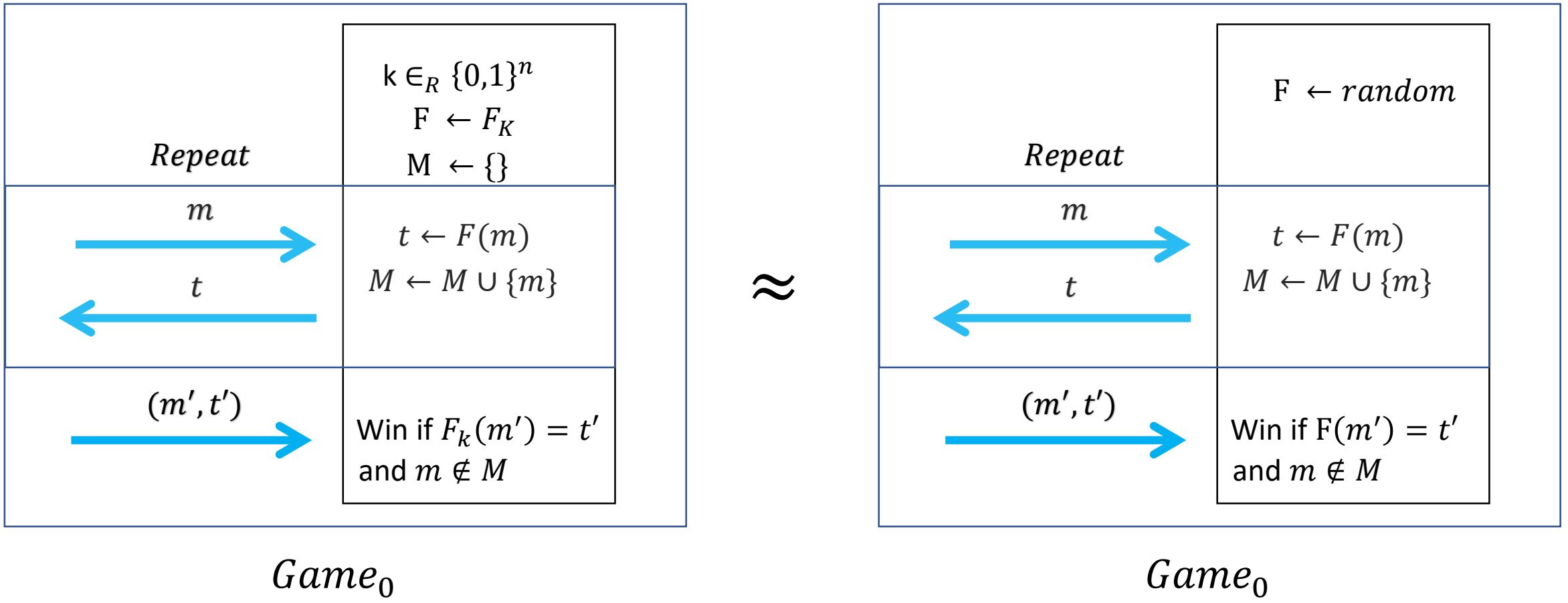
Mac example exercise

- Given two schemes mac', mac'' construct a mac scheme mac scheme mac such that mac is secure as long as either mac' or mac'' is secure

$$keygen(1^s) \coloneqq k_1 \leftarrow keygen'(1^s), k_2 \leftarrow keygen''(1^s)$$

$$mac_{(k_1,k_2)}(m) \coloneqq mac'_{k_1}(m) \,||\, mac''_{k_2}(m)$$

Security of macs from PRF via hybrid games



$Game_0$ $Game_0$

Mac forgery example

- Let mac’ and mac’’ be mac schemes, is the following a mac?

$$\mathrm{m}ac_k(m) \coloneqq \; mac'_k(m) \oplus mac''_k(m)$$

Collision resistance hash function

- A function h is collision-resistant if it is hard to find x, y such that $\mathrm{h}(x) = h(y)$

Collision-resistant hash function exercises

If $H: \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ is a CRHF

1. $H'(b \,||\, x) := b \,||\, H(x)$ a CHR? Yes
   1. Input longer than output
   2. $H(b \,||\, x) = H(b' \,||\, y) \Rightarrow b = b' \; and \; H(x) = H(y')$

Collision-resistant hash function exercises

If $H: \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ is a CRHF

1. $H'(x \,||\, y) := H(x) \,||\, H(y)$ a $CRHF?$ Yes
    1. Input longer than output
    2. H(x) = H(x') or H(y) = H(y')

Collision-resistant hash function exercises

If $H: \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ is a CRHF

1. $H'(x, y) := H(x) \oplus H(y)$ a CRHF? No
   - If x = y then $H'(x, y) = 0$

Collision-resistant hash function exercises

If $H: \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ is a CRHF

1. $H'(x, y) := H(x) \oplus H(y)$ a CRHF? No
   - If x = y then $H'(x, y) = 0$

Collision-resistant hash function exercises

If $H: \{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ is a CRHF

1. $H'(x||y) := H(x) \oplus y$ a CRHF? No

- Set $y = H(x)$

  $H'(x||y) = H(x) \oplus y$

  $= H(x) \oplus H(x) = 0$

Computational assumptions

Types of computational assumptions

- Decisional: distinguish between two distributions (or games)

- Search: trying to find an element which fulfills certain properties

Discrete logarithm assumption

- Reminder
  - G be group of size n
  - g is a generator of g if $\{g^1, g^2, \dots, g^n\} = g$

- Given an output y sampled randomly, it is hard to find x such that $g^x = y$

Examples of hardness reduction

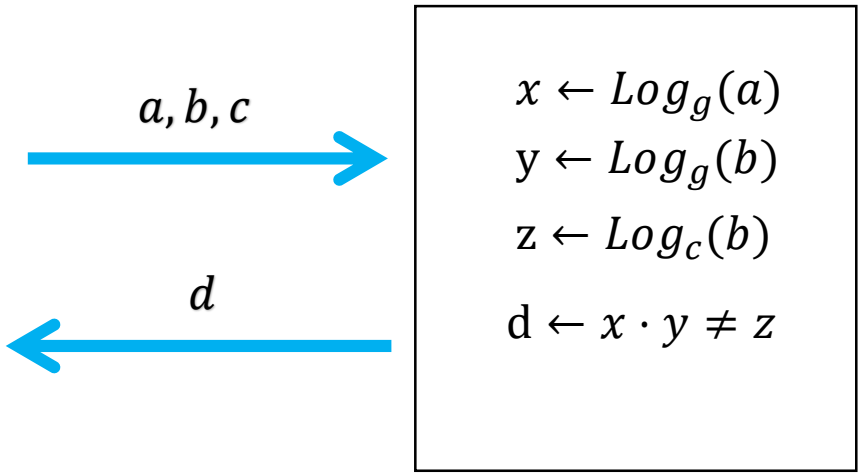
- Given g, h random generators, finding x, y such that $g^x = h^y$ is hard
- $(g^x)^{y/x} = h^y$
- Finding such x, y is equivalent to finding the $\log_g(h)$

Decisional Diffie Hellman problem

- Reminder
  - G be group of size n
  - g is a generator of g if $\{g^1, g^2, \dots, g^n\} = G$
- The computational Diffie-Hellman assumption states that the two following game are indistinguishable

Reduction

- DDH is hard ⇒ discrete logarithm is hard
  - Equivalent to saying that an efficient solver for discrete logarithm can be used to construct an efficient distinguisher for DDH.



Quadratic Residuosity

- Reminder
  - G be group of size n

- We say that $y \in G$ is a quadratic residue $(y \in QR)$
  if there exists an $x \in G$ such that $y = x^2$

- We say that $y \in G$ is a quadratic non-residue $(y \in QNR)$
  if there exists no $x \in G$ such that $y = x^2$

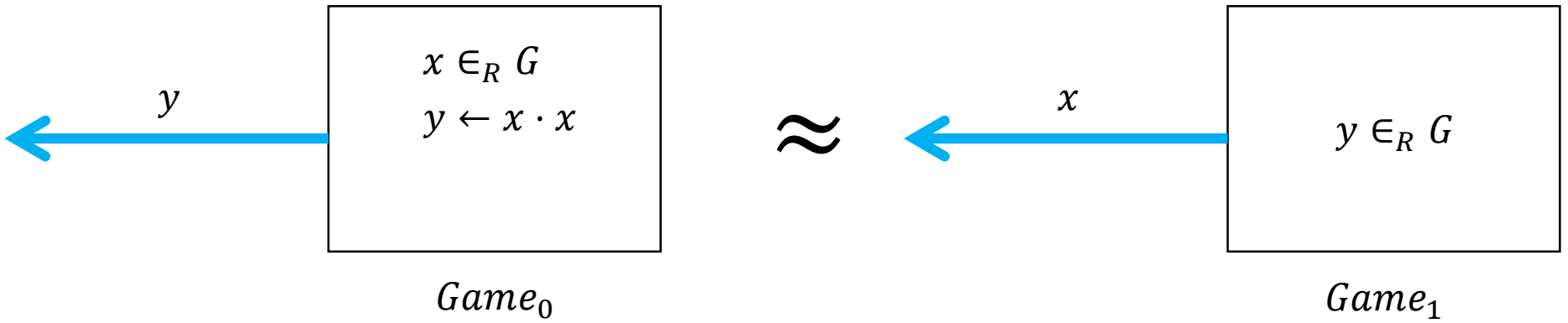
Quadratic Residuosity

- Reminder
  - G be group of size n

- Properties:
  - $x, y \in QR \Rightarrow x \cdot y \in QR$
  - $x \in QR, y \in QNR \Rightarrow x \cdot y \in QNR$
  - $x \in QNR, y \in QNR \Rightarrow x \cdot y \in QR$

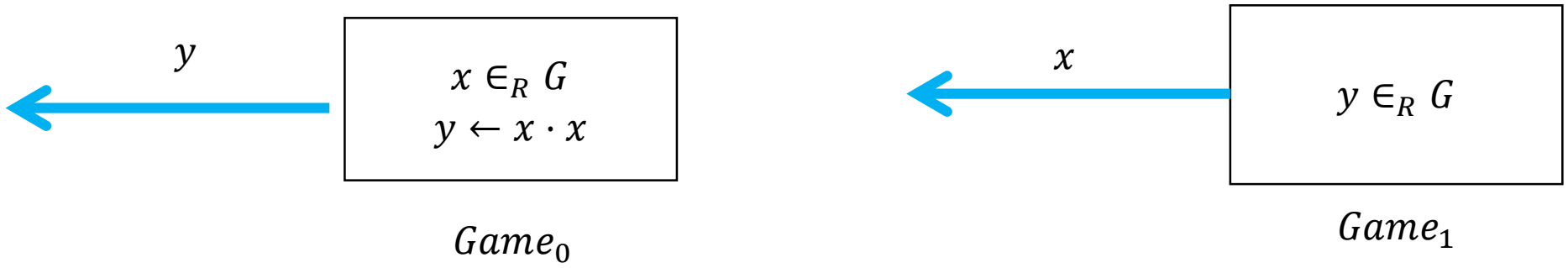
Quadratic Residuosity

- Reminder
  - G be group of size n
  - g is a generator of g if $\{g^1, g^2, \dots, g^n\} = g$
- The quadratic Residuosity assumption states that the two following game are indistinguishable



Equivalence for quadratic residuosity problem

- $Game_0 \approx Game_1 \Leftrightarrow Game_0 \approx Game_2$

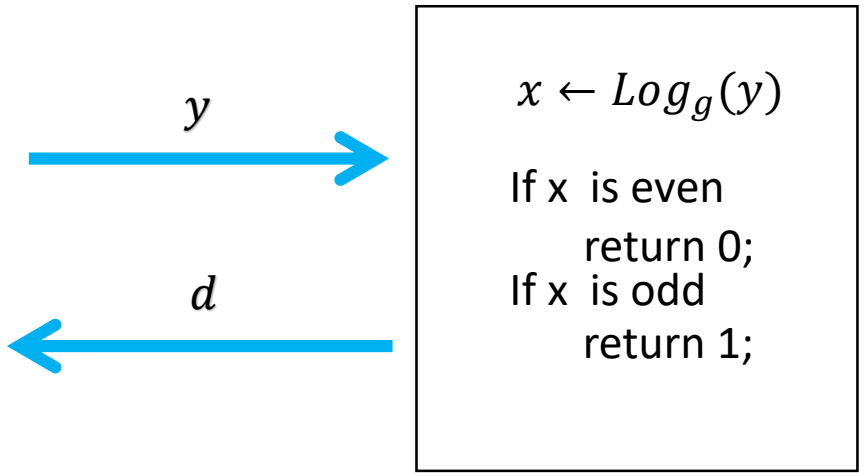


QR is hard ⇒ discrete logarithm is hard

- Theorem I: All generators g of G belong to QNR
  - Proof
    - $g \in QR \Rightarrow g^i \in QR$
    - g does not generate $x \in QNR$
    - g is not a generator

- Theorem II: Let g be a generator from G, $y \in QR \Leftrightarrow x = g^{2m}$
  - $y \in QR \Rightarrow x = g^{2m}$ follows from the fact that $y = x^2 = (g^m)^2$
  - $y \in QR \Leftarrow x = g^{2m}$ follows from the fact g^{2i+1} is in QNR
    - $u \in QR, v \in QNR \Rightarrow u \cdot b \in QNR$

QR is hard ⇒ discrete logarithm is hard



Factoring assumption

- Given n such that $n = p \cdot q$ where
  - p, q are prime
  - p, q were randomly sampled
  - Most significant bits of p, q are $1,1$
- It is hard to compute p, q

Polynomial recovery

- Parameters
  - n : total number of points $\big(1, f(1)\big), \dots, (n, f(n))$
  - d : degree of polynomial
  - t : points that agree with polynomial
- Problems
  - Find polynomial p of degree d such that $\big(i, f(i)\big) = (i, p(i))$ for at least t points

Learning with errors

- Parameters
  - Random linear function f: $Z_p^n \rightarrow Z_p$
  - Error vectors e chosen from some Distribution $\mathcal{X}$
  - Random vectors x_i

- Problem compute $f(x)$ from $f(x_i) + e_i$

Example of LWE problem

- $X = \{-1,0,1\}$
- f$(x, y, z) \coloneqq 5x + 7y + 3z \ \ (mod\ 23)$

- (x,y,z) = (2,3,1), e = 1 f(2,3,1) + 1 = 12
- (x,y,z) = (1,1,3), e = -1 f(1,1,3) - 1 = 20