Pseudorandomness IV and Encryption I

601.442/642 Modern Cryptography

19th February 2026

Logistics

Logistics

- Midterm 1 done! Congratulations!

Logistics

- Midterm 1 done! Congratulations!
    - There will be a curve

Logistics

- Midterm 1 done! Congratulations!
    - There will be a curve
- Homework 4 assigned today, due in one week (on 02/26)

Logistics

- Midterm 1 done! Congratulations!
  - There will be a curve
- Homework 4 assigned today, due in one week (on 02/26)
  - See updated schedule on Canvas

Constructing PRFs

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali ’84)

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali ’84)
    - Assume a PRG $G : \{0,1\}^\lambda \rightarrow \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation’s output.

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali ’84)
  - Assume a PRG $G : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation’s output.
- In practice, we use AES.

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali '84)
  - Assume a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation's output.
- In practice, we use AES.
  - $\{\mathsf{AES_k}\}_{k \in \{0,1\}^{128}}$, also supports key sizes of 192 and 256

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali '84)
  - Assume a PRG $G : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation's output.
- In practice, we use AES.
  - $\{\mathsf{AES}_\mathsf{k}\}_{k \in \{0,1\}^{128}}$, also supports key sizes of 192 and 256
  - $\mathsf{AES}_k : \{0,1\}^{128} \to \{0,1\}^{128}$

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali '84)
  - Assume a PRG $G : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation's output.
- In practice, we use AES.
  - $\{\mathsf{AES}_\mathsf{k}\}_{k \in \{0,1\}^{128}}$, also supports key sizes of 192 and 256
  - $\mathsf{AES}_k : \{0,1\}^{128} \to \{0,1\}^{128}$
- How things like AES are designed is important, but outside the scope of this course

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali '84)
  - Assume a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation's output.
- In practice, we use AES.
  - $\{\mathsf{AES}_\mathsf{k}\}_{k\in\{0,1\}^{128}}$, also supports key sizes of 192 and 256
  - $\mathsf{AES}_k : \{0,1\}^{128} \to \{0,1\}^{128}$
- How things like AES are designed is important, but outside the scope of this course
- You may also see AES referred to as a *block cipher*. This means it has a "decrypt" procedure, essentially a built-in way to reverse the output.

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali '84)
  - Assume a PRG $G : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation's output.
- In practice, we use AES.
  - $\{\mathsf{AES_k}\}_{k \in \{0,1\}^{128}}$, also supports key sizes of 192 and 256
  - $\mathsf{AES}_k : \{0,1\}^{128} \to \{0,1\}^{128}$
- How things like AES are designed is important, but outside the scope of this course
- You may also see AES referred to as a *block cipher*. This means it has a "decrypt" procedure, essentially a built-in way to reverse the output.
  - Note: This means AES is actually a pseudrandom *permutation*!

Constructing PRFs

- Theory: PRFs can be constructed from PRGs (Goldreich-Goldwasser-Micali '84)
  - Assume a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$. Use the PRF input x to descend a tree built out of invoking G on either the left or right half of the previous invocation's output.
- In practice, we use AES.
  - $\{\mathsf{AES}_\mathsf{k}\}_{k\in\{0,1\}^{128}}$, also supports key sizes of 192 and 256
  - $\mathsf{AES}_k : \{0,1\}^{128} \to \{0,1\}^{128}$
- How things like AES are designed is important, but outside the scope of this course
- You may also see AES referred to as a *block cipher*. This means it has a "decrypt" procedure, essentially a built-in way to reverse the output.
  - Note: This means AES is actually a pseudrandom *permutation*!
  - In practice, PRPs and PRFs can be used interchangeably. We will prove this if we have time.

PRF Game

PRF Game

$$b \xleftarrow{\$} \{0,1\}$$
$$k \xleftarrow{\$} \{0,1\}^\lambda$$
$$T := \{\}$$

PRF Game

$$b \xleftarrow{\$} \{0,1\}$$
$$k \xleftarrow{\$} \{0,1\}^{\lambda}$$
$$T := \{\}$$

$$\xleftarrow{\quad x \quad}$$

PRF Game

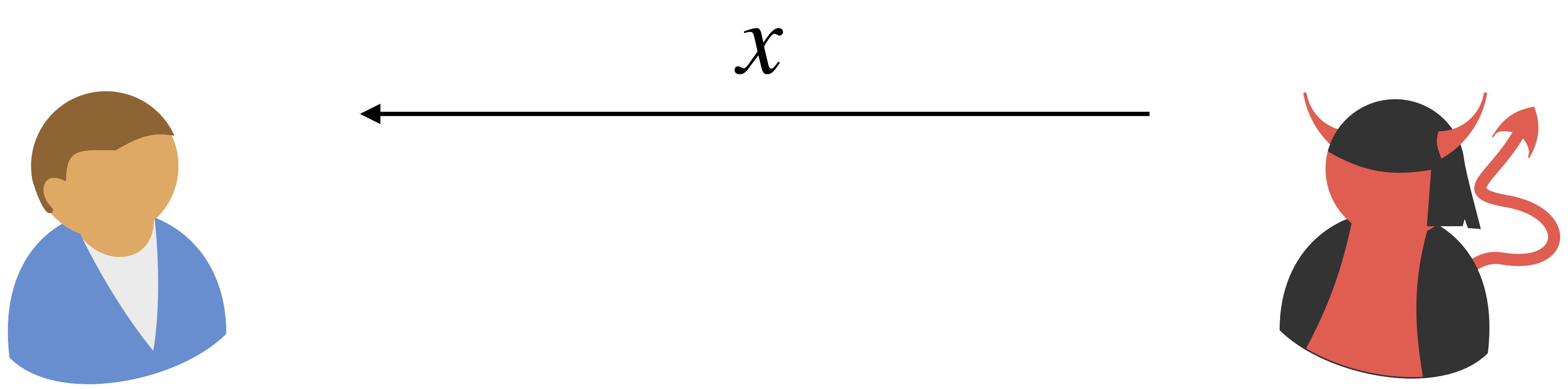
$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$

$y = F_k(x)$



PRF Game

$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$

$\quad y = F_k(x)$

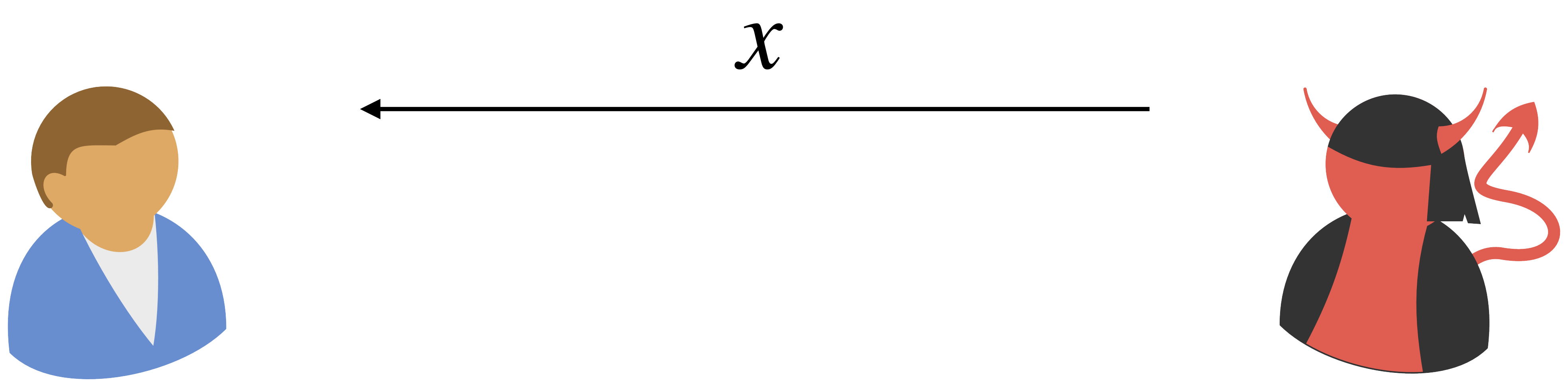
else

$\quad$ **if** $x \notin T$

$\quad\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad\quad T[x] = r$

$\quad y = T[x]$



PRF Game

$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$

$\quad y = F_k(x)$

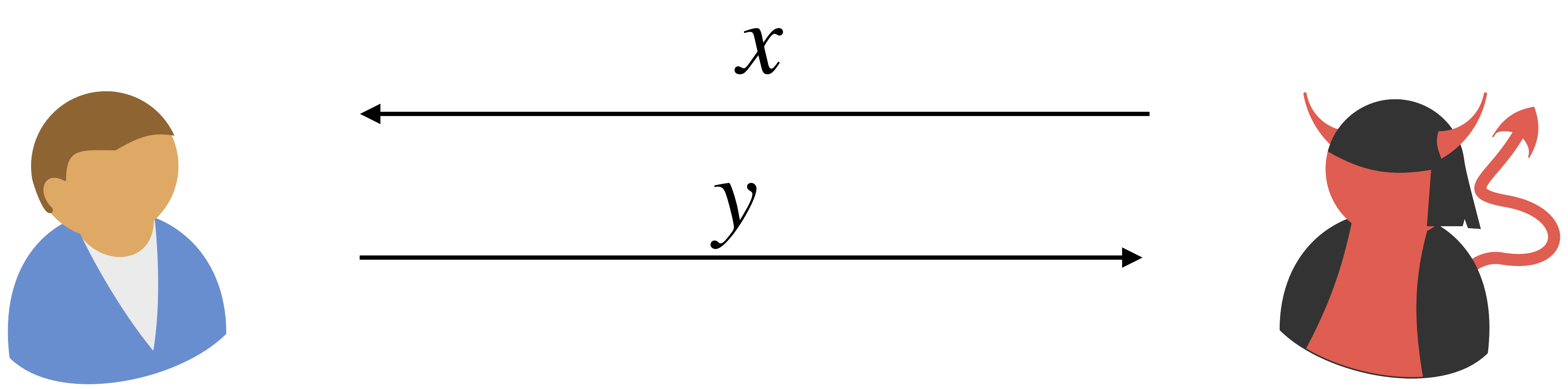
else

$\quad$ **if** $x \notin T$

$\quad\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad\quad T[x] = r$

$\quad y = T[x]$



PRF Game

$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$

$\quad y = F_k(x)$

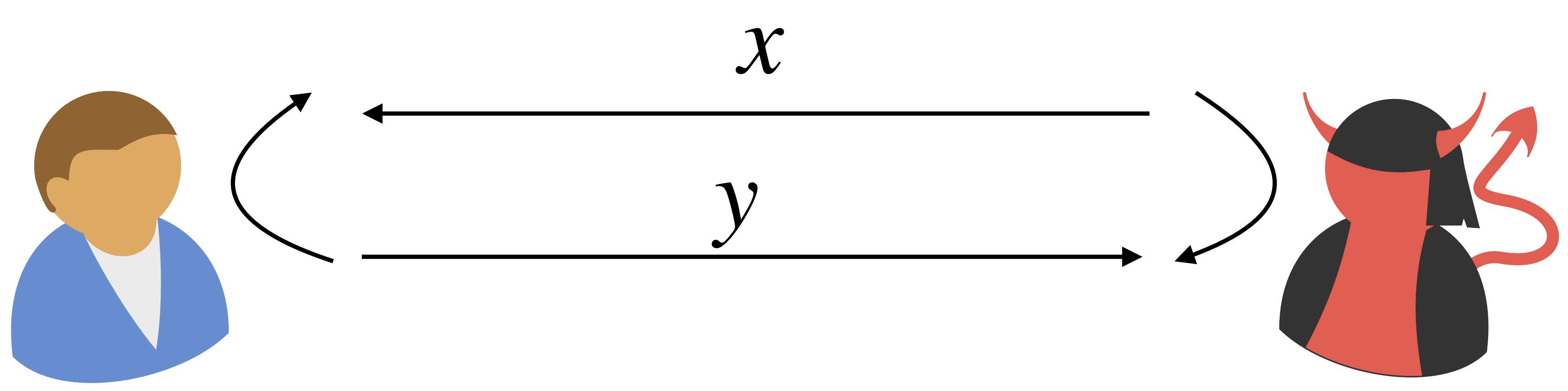
else

$\quad$ **if** $x \notin T$

$\qquad r \xleftarrow{\$} \{0,1\}^\lambda$

$\qquad T[x] = r$

$\quad y = T[x]$



PRF Game

$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$

$\quad y = F_k(x)$

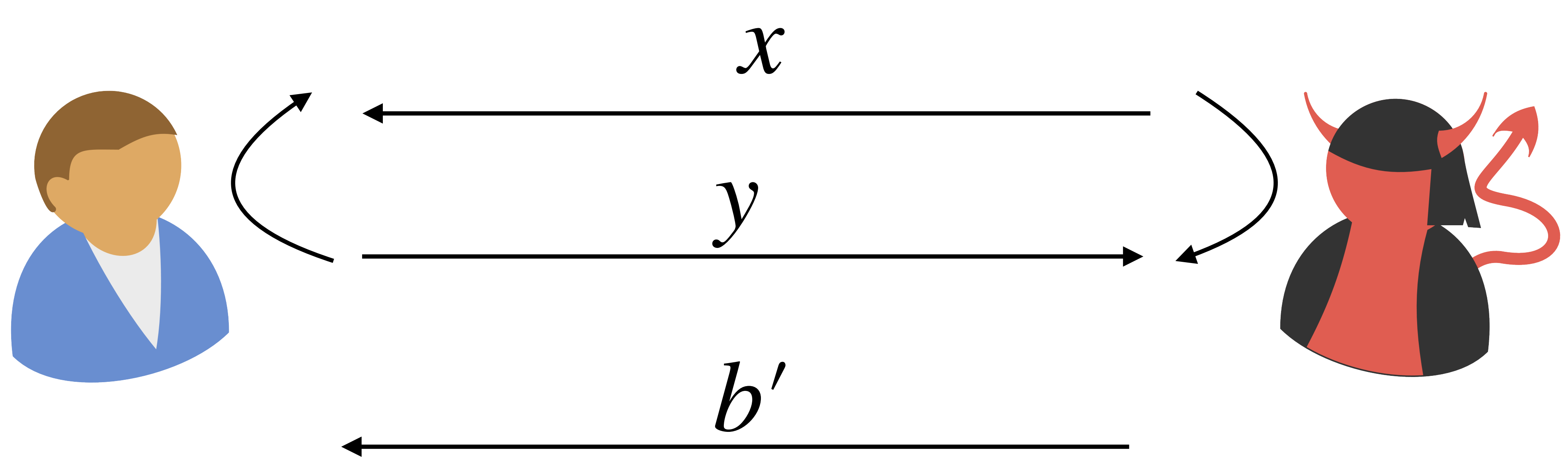
else

$\quad$ **if** $x \notin T$

$\quad\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad\quad T[x] = r$

$\quad y = T[x]$



PRF Game

$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$

$\quad y = F_k(x)$

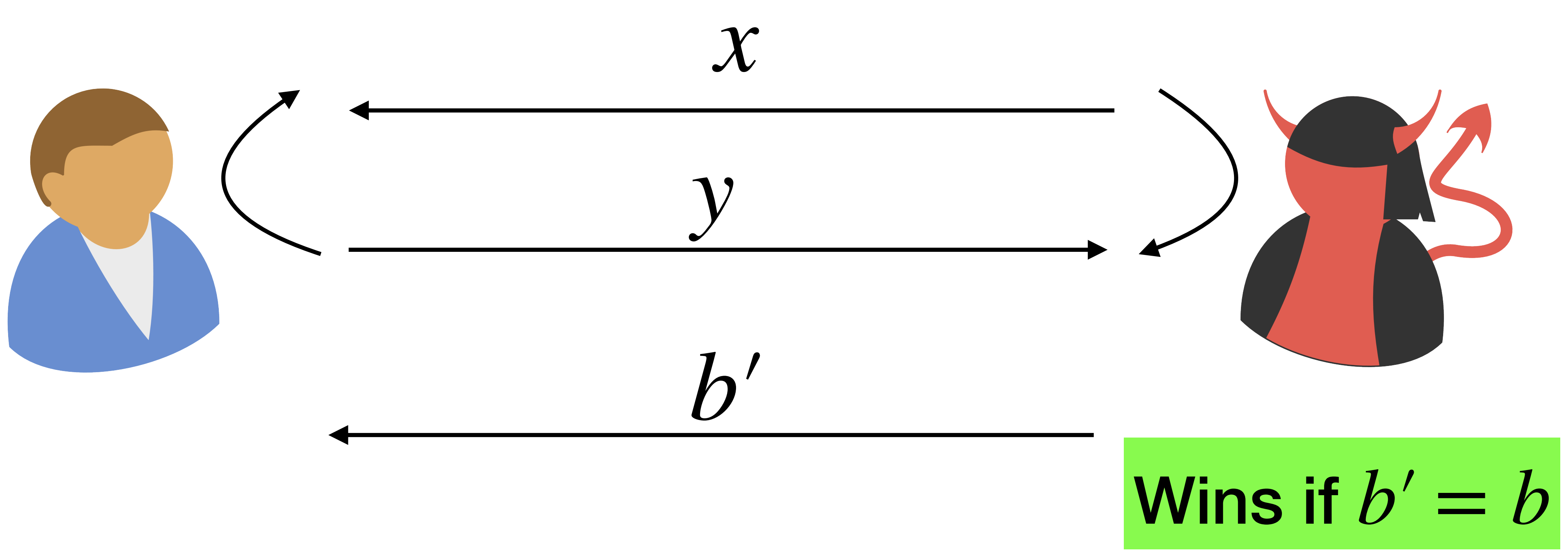
else

$\quad$ **if** $x \notin T$

$\quad\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad\quad T[x] = r$

$\quad y = T[x]$



PRF Game

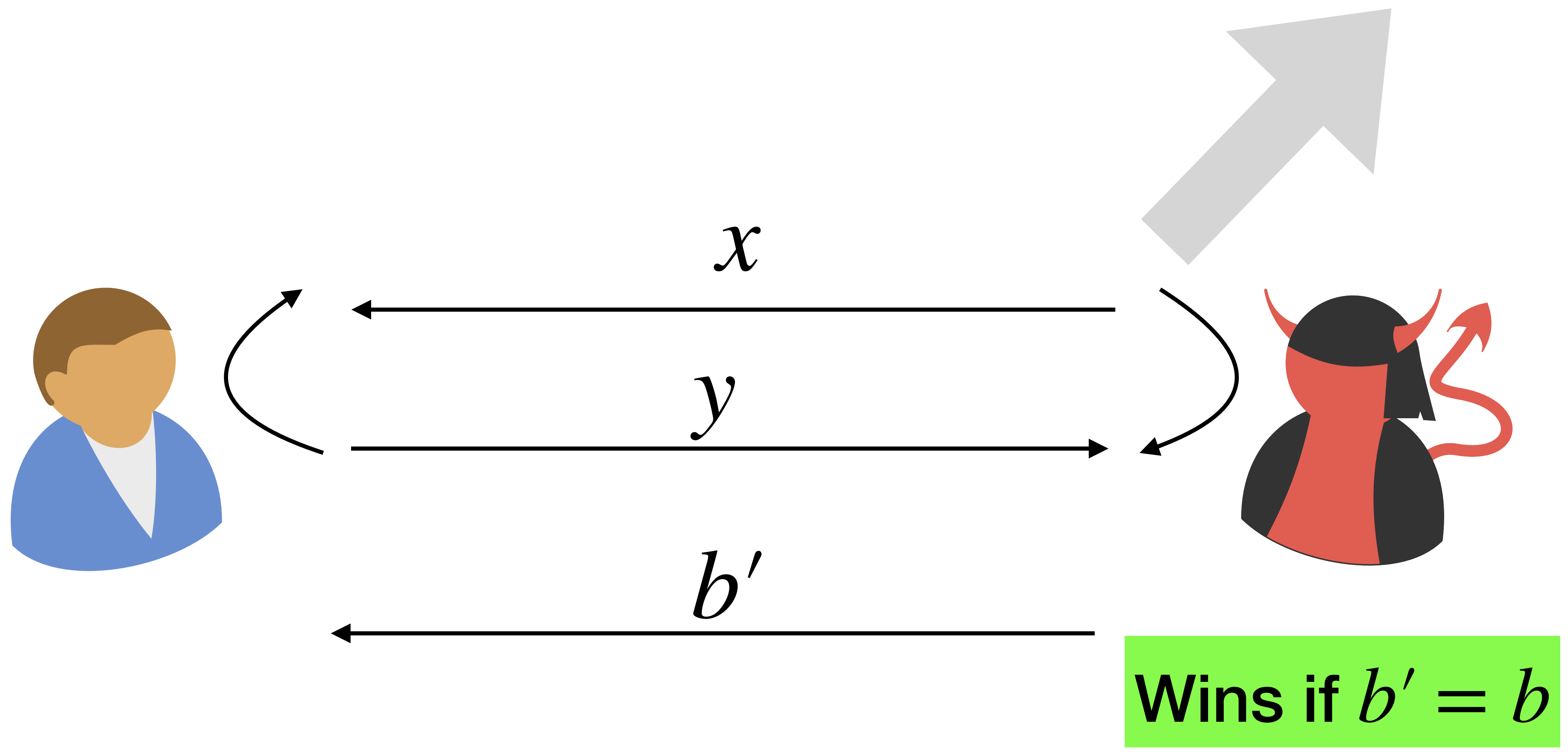
$\mathcal{A}$ can be *adaptive*

$b \xleftarrow{\$} \{0,1\}$

$k \xleftarrow{\$} \{0,1\}^\lambda$

$T := \{\}$

if $b = 0$
    $y = F_k(x)$
else
    if $x \notin T$
        $r \xleftarrow{\$} \{0,1\}^\lambda$
        $T[x] = r$
    $y = T[x]$



Multi-Message Security

Multi-Message Security

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \quad \overset{c}{\approx} \quad D_1 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

Multi-Message Security

Multi-Message Security

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

Multi-Message Security

Multi-Message Security

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

$$b \xleftarrow{\$} \{0,1\}$$
$$k \leftarrow \mathsf{KeyGen}(1^\lambda)$$
for $i = 1 \ldots q(\lambda)$:

Multi-Message Security

Multi-Message Security

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

$b \overset{\$}{\leftarrow} \{0,1\}$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

for $i = 1 \ldots q(\lambda)$:

$\quad c_i = \mathsf{Enc}(k, m_b^i)$

Multi-Message Security

<u>Multi-Message Security</u>

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

$$b \overset{\$}{\leftarrow} \{0,1\}$$
$$k \leftarrow \mathsf{KeyGen}(1^\lambda)$$
$$\text{for } i = 1 \ldots q(\lambda):$$
$$c_i = \mathsf{Enc}(k, m_b^i)$$

$$\xrightarrow{\{c_i\}_{i=1}^{q(\lambda)}}$$

Multi-Message Security

<u>Multi-Message Security</u>

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \text{ct}: \begin{array}{c} k \leftarrow \text{KeyGen}(1^\lambda) \\ \text{ct} \leftarrow \{\text{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \text{ct}: \begin{array}{c} k \leftarrow \text{KeyGen}(1^\lambda) \\ \text{ct} \leftarrow \{\text{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

$$b \overset{\$}{\leftarrow} \{0,1\}$$

$$k \leftarrow \text{KeyGen}(1^\lambda)$$

$$\text{for } i = 1 \ldots q(\lambda):$$

$$c_i = \text{Enc}(k, m_b^i)$$

$$\xrightarrow{\{c_i\}_{i=1}^{q(\lambda)}}$$

$$\xleftarrow{b'}$$

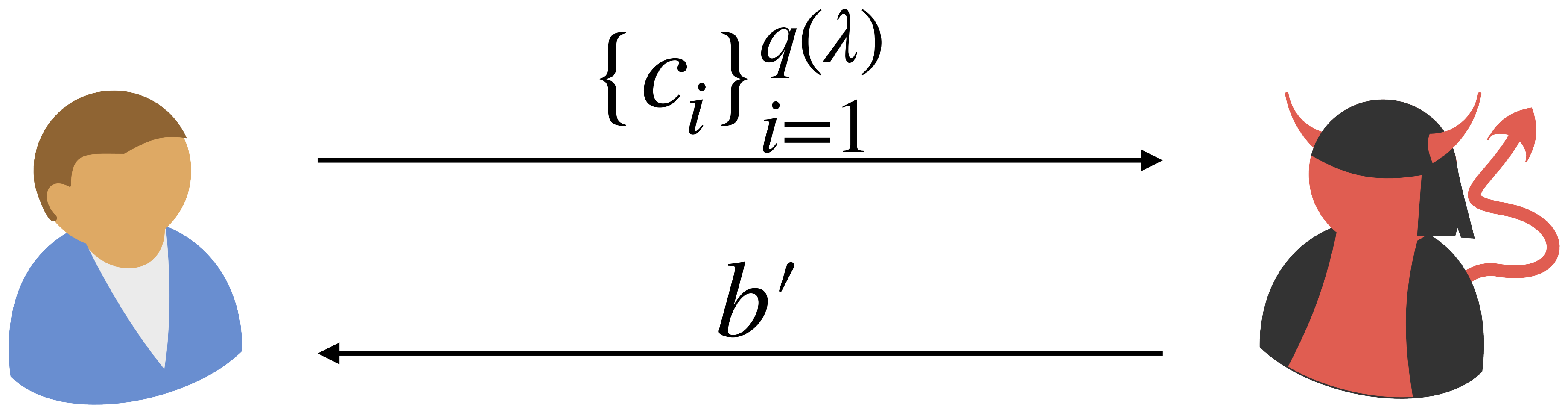
Multi-Message Security

<u>Multi-Message Security</u>

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}(1^\lambda) \\ \text{ct} \leftarrow \{\text{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}(1^\lambda) \\ \text{ct} \leftarrow \{\text{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

$$b \xleftarrow{\$} \{0,1\}$$
$$k \leftarrow \text{KeyGen}(1^\lambda)$$
$$\text{for } i = 1 \ldots q(\lambda) :$$
$$\quad c_i = \text{Enc}(k, m_b^i)$$



Wins if $b' = b$

Multi-Message Security

<u>**Multi-Message Security**</u>

An encryption scheme with message length $\ell := \ell(\lambda)$ is multi-message secure if $\forall \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_0^i)\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{ct} \leftarrow \{\mathsf{Enc}(k, m_1^i)\}_{i=1}^{q(n)} \end{array} \right\}$$

$$b \xleftarrow{\$} \{0,1\}$$
$$k \leftarrow \mathsf{KeyGen}(1^\lambda)$$
$$\text{for } i = 1 \ldots q(\lambda):$$
$$\quad c_i = \mathsf{Enc}(k, m_b^i)$$

$$\xrightarrow{\{c_i\}_{i=1}^{q(\lambda)}}$$
$$\xleftarrow{b'}$$

Wins if $b' = b$

No adaptivity!
All messages must be "chosen" upfront.

Adaptive Multi-Message Security

$\mathsf{KeyGen} \to k$

$\mathsf{Enc}(k, m) \to c$

$\mathsf{Dec}(k, c) \to m'$

Adaptive Multi-Message Security

KeyGen $\rightarrow k$
Enc$(k, m) \rightarrow c$
Dec$(k, c) \rightarrow m'$

Adaptive Multi-Message Security

$$\mathsf{KeyGen} \rightarrow k$$
$$\mathsf{Enc}(k, m) \rightarrow c$$
$$\mathsf{Dec}(k, c) \rightarrow m'$$

$$b \xleftarrow{\$} \{0,1\}$$
$$k \leftarrow \mathsf{KeyGen}(1^\lambda)$$

Adaptive Multi-Message Security

$\mathsf{KeyGen} \to k$
$\mathsf{Enc}(k, m) \to c$
$\mathsf{Dec}(k, c) \to m'$

$b \xleftarrow{\$} \{0,1\}$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\xleftarrow{\quad (m_0, m_1) \quad}$

Adaptive Multi-Message Security

KeyGen $\to k$
Enc$(k, m) \to c$
Dec$(k, c) \to m'$

$b \xleftarrow{\$} \{0,1\}$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(k, m_b)$

(m_0, m_1)

Adaptive Multi-Message Security

$\mathsf{KeyGen} \to k$

$\mathsf{Enc}(k, m) \to c$

$\mathsf{Dec}(k, c) \to m'$

$b \xleftarrow{\$} \{0,1\}$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(k, m_b)$

$\xleftarrow{(m_0, m_1)}$

$\xrightarrow{\mathsf{ct}_b}$

Adaptive Multi-Message Security

$\mathsf{KeyGen} \to k$

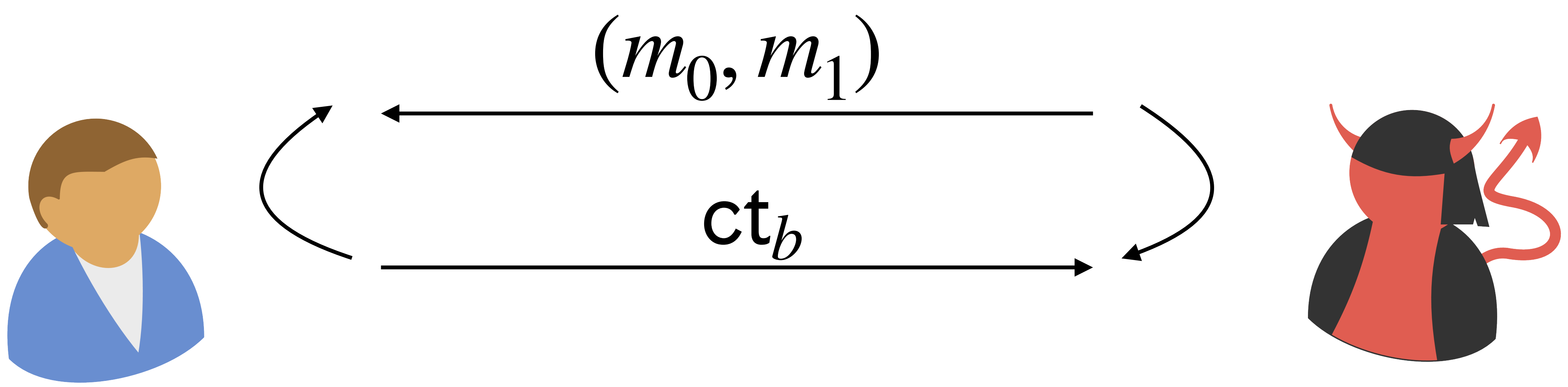
$\mathsf{Enc}(k, m) \to c$

$\mathsf{Dec}(k, c) \to m'$

$$b \xleftarrow{\$} \{0,1\}$$

$$k \leftarrow \mathsf{KeyGen}(1^\lambda)$$

$$\mathsf{ct}_b \leftarrow \mathsf{Enc}(k, m_b)$$



Adaptive Multi-Message Security

$\mathsf{KeyGen} \to k$

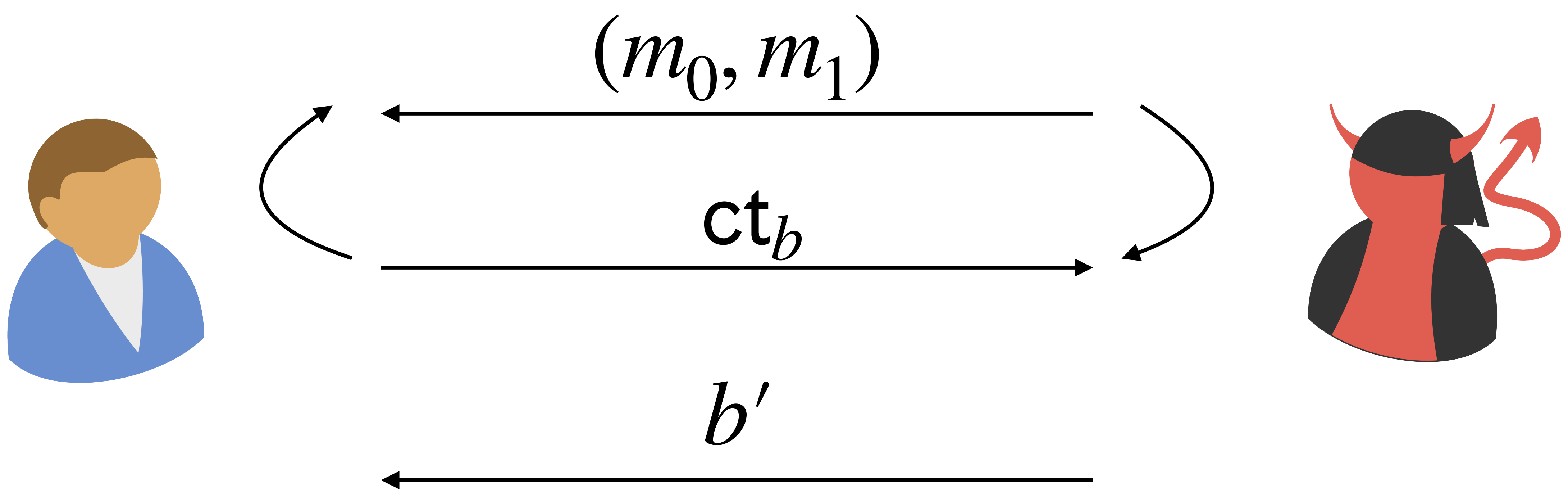
$\mathsf{Enc}(k, m) \to c$

$\mathsf{Dec}(k, c) \to m'$

$b \xleftarrow{\$} \{0,1\}$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(k, m_b)$



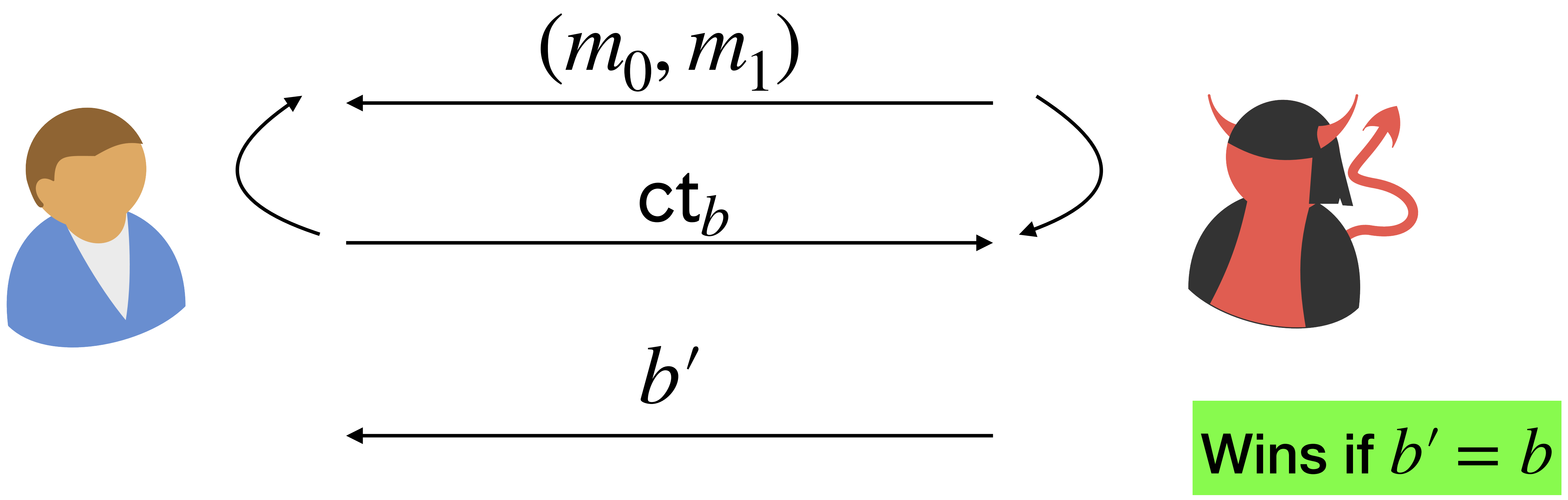
Adaptive Multi-Message Security

KeyGen $\to k$
Enc$(k, m) \to c$
Dec$(k, c) \to m'$

$b \xleftarrow{\$} \{0,1\}$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(k, m_b)$



Adaptive Multi-Message Security

KeyGen $\rightarrow k$

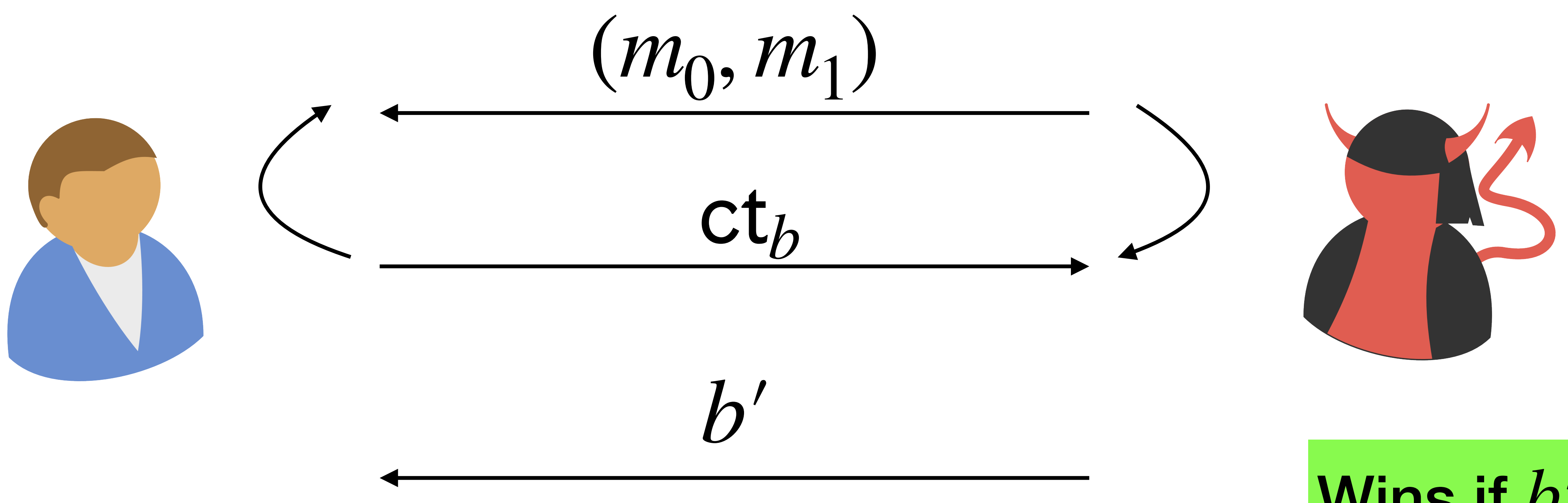
$\text{Enc}(k, m) \rightarrow c$

$\text{Dec}(k, c) \rightarrow m'$

$b \xleftarrow{\$} \{0,1\}$

$k \leftarrow \text{KeyGen}(1^\lambda)$

$\text{ct}_b \leftarrow \text{Enc}(k, m_b)$



Wins if $b' = b$

Ciphertexts should be **indistinguishable**, even when the adversary gets to **choose** the plaintext

IND-CPA Security

IND-CPA Security

IND-CPA Security

IND-CPA Security

An encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

IND-CPA Security

<u>IND-CPA Security</u>

An encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins GuessGame}] \leq \frac{1}{2} + \nu(\lambda)$$

IND-CPA Security

<u>IND-CPA Security</u>

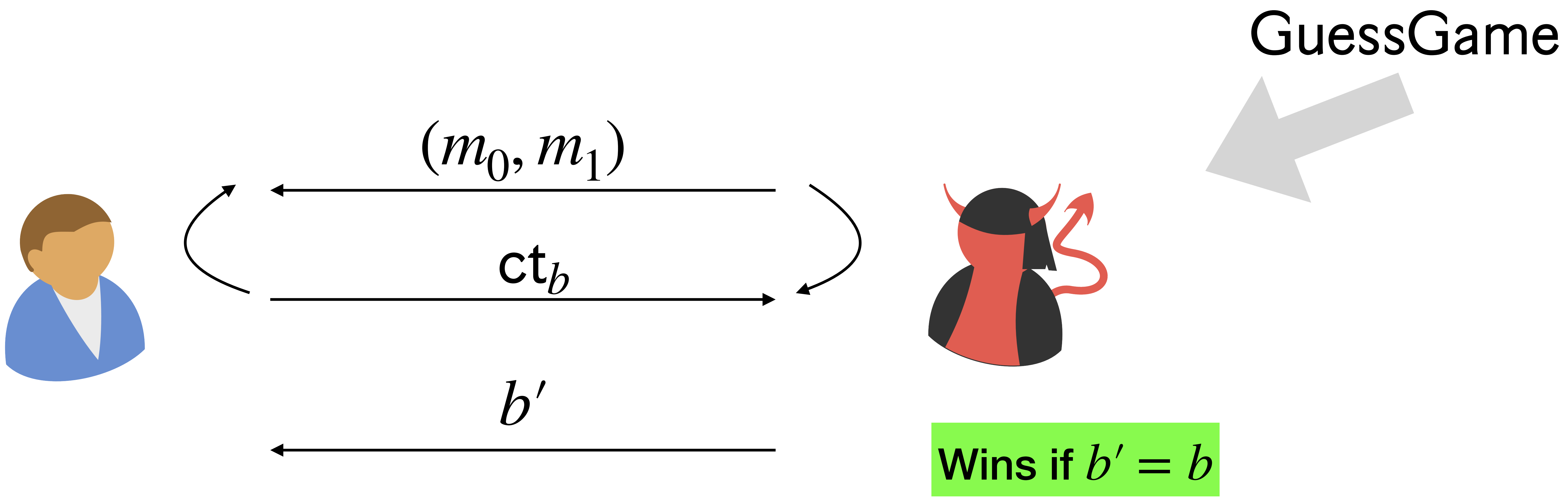
An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins GuessGame}] \leq \frac{1}{2} + \nu(\lambda)$$

$$b \xleftarrow{\$} \{0,1\}$$

$$k \leftarrow \text{KeyGen}(1^\lambda)$$

$$\text{ct}_b \leftarrow \text{Enc}(k, m_b)$$



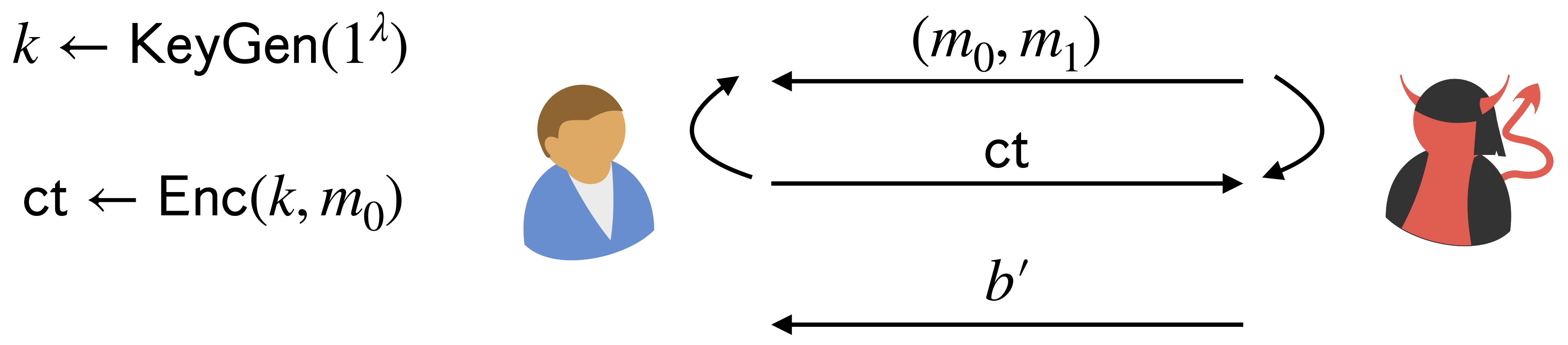
IND-CPA Security

<u>**IND-CPA Security**</u>

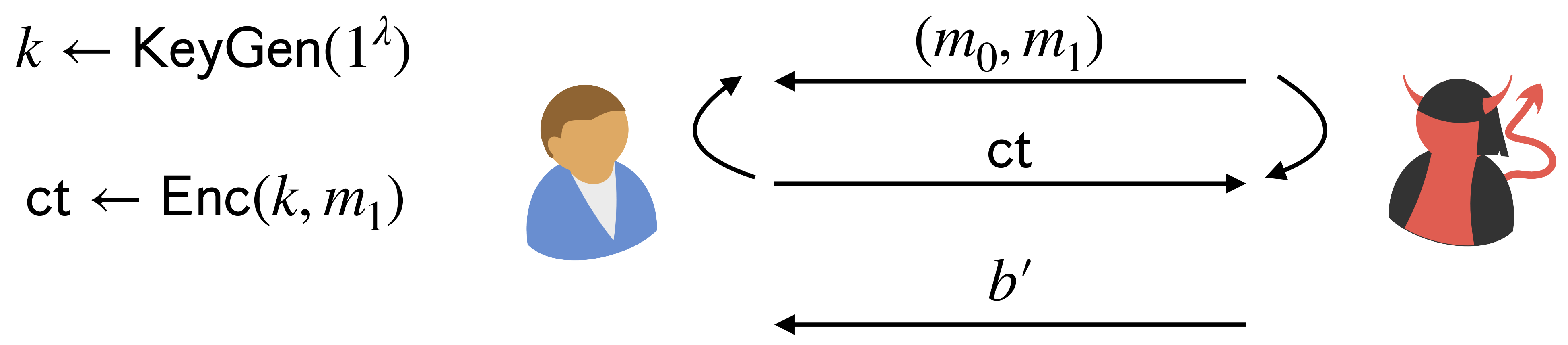
An encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\left| \Pr[\mathscr{A} \text{ outputs } 1 \text{ in } \mathsf{Game}_0] - \Pr[\mathscr{A} \text{ outputs } 1 \text{ in } \mathsf{Game}_1] \right| \leq \nu(\lambda)$$

Game_0



Game_1



IND-CPA Secure Encryption Construction

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a “multi-message” secure encryption scheme already satisfies IND-CPA security

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a “multi-message” secure encryption scheme already satisfies IND-CPA security

PRF Encryption

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a “multi-message” secure encryption scheme already satisfies IND-CPA security

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k\in\{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \rightarrow \{0,1\}^{\ell(\lambda)}$.

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a "multi-message" secure encryption scheme already satisfies IND-CPA security

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k \in \{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a “multi-message” secure encryption scheme already satisfies IND-CPA security

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k\in\{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m):

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a "multi-message" secure encryption scheme already satisfies IND-CPA security

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k \in \{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m):
  - $x \xleftarrow{\$} \{0,1\}^\lambda$

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a “multi-message” secure encryption scheme already satisfies IND-CPA security

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k\in\{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m):
  - $x \xleftarrow{\$} \{0,1\}^\lambda$
  - ct $:= (m \oplus F_k(x), x)$

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a "multi-message" secure encryption scheme already satisfies IND-CPA security

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k \in \{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m):
  - $x \xleftarrow{\$} \{0,1\}^\lambda$
  - ct $:= (m \oplus F_k(x), x)$
- Dec$(k, (c, x))$: $m := F_k(x) \oplus c$.

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a "multi-message" secure encryption scheme already satisfies IND-CPA security

Theorem: Any IND-CPA secure encryption scheme cannot be deterministic and stateless

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k \in \{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m):
  - $x \xleftarrow{\$} \{0,1\}^\lambda$
  - ct $:= (m \oplus F_k(x), x)$
- Dec$(k, (c, x))$: $m := F_k(x) \oplus c$.

IND-CPA Secure Encryption Construction

- As it turns out, our construction of a "multi-message" secure encryption scheme already satisfies IND-CPA security

Theorem: Any IND-CPA secure encryption scheme cannot be deterministic and stateless

PRF Encryption

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial and, Let $\{F_k\}_{k \in \{0,1\}^\lambda}$ be a secure family of PRFs, where $F_k : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$.

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m):
  - $x \xleftarrow{\$} \{0,1\}^\lambda$
  - ct $:= (m \oplus F_k(x), x)$
- Dec$(k, (c, x))$: $m := F_k(x) \oplus c$.

Only applies to messages of fixed length $\ell(\lambda)$

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{c} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$

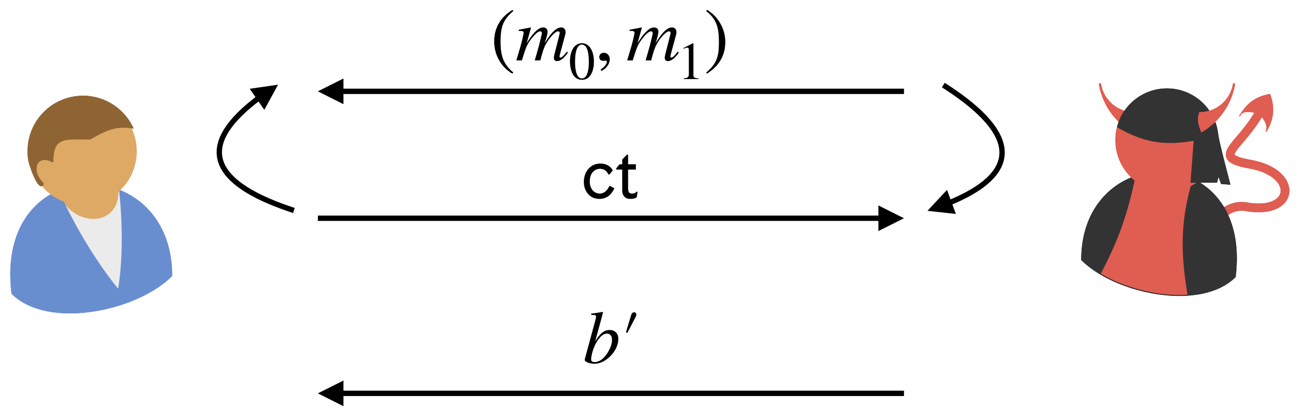
$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

Proof of Security

H_0

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

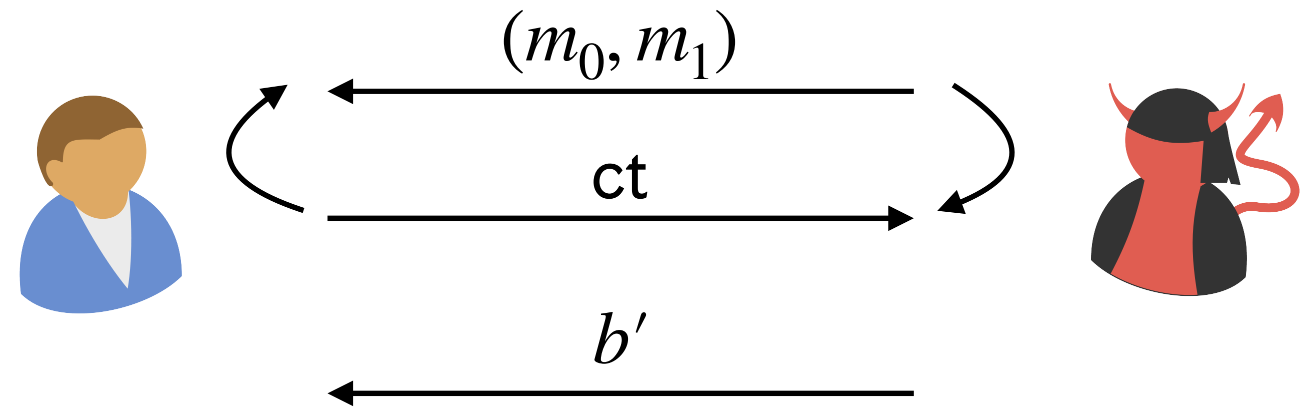
$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

H_0

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

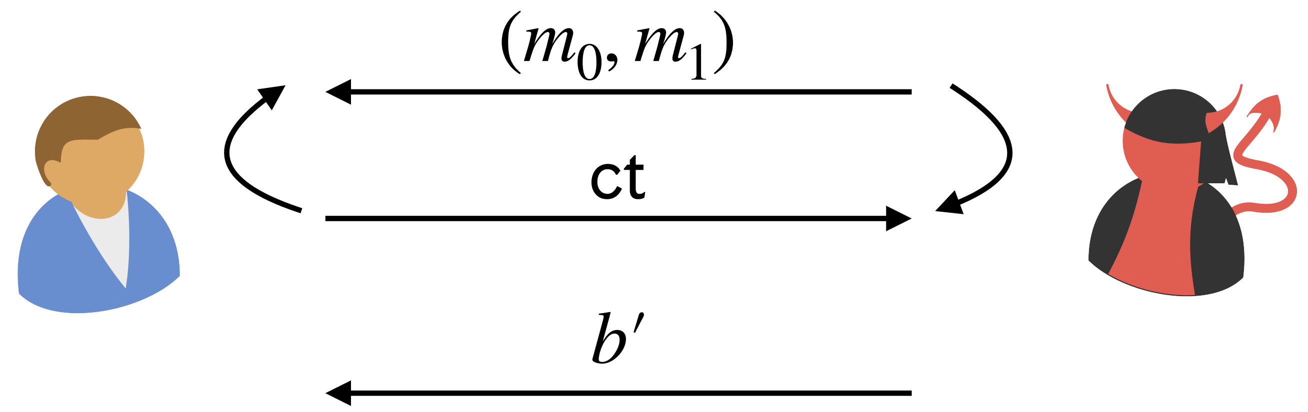
$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



$H_?$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

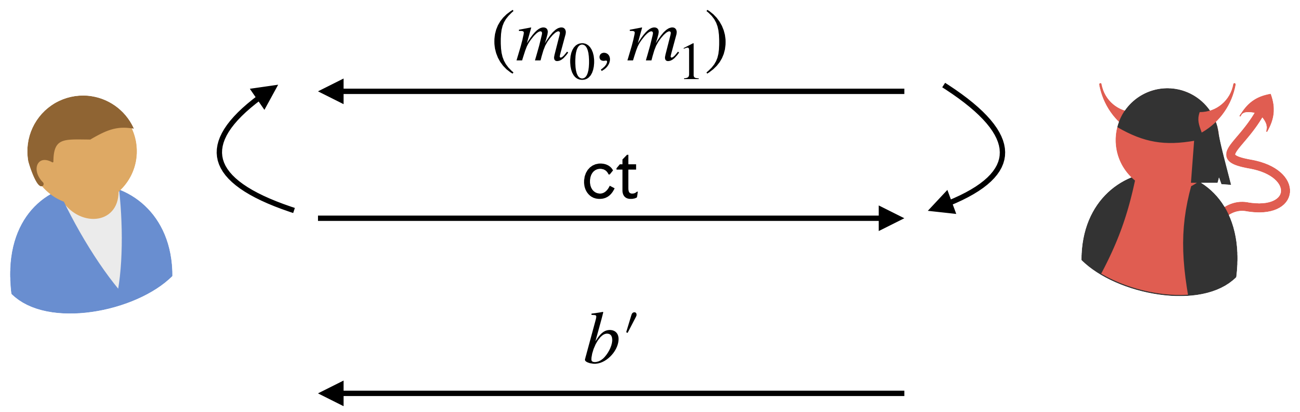
Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k,(c,x))$: $m' := F_k(x) \oplus c$

H_0

$k \leftarrow$ KeyGen(1^λ)

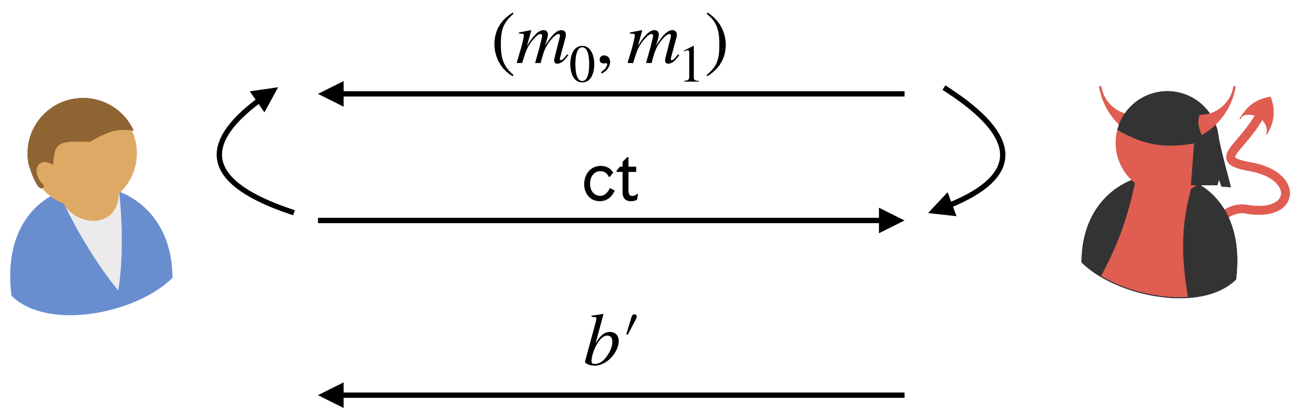
ct $\leftarrow$ Enc(k, m_0)



$H_?$

$k \leftarrow$ KeyGen(1^λ)

ct $\leftarrow$ Enc(k, m_1)



Proof of Security

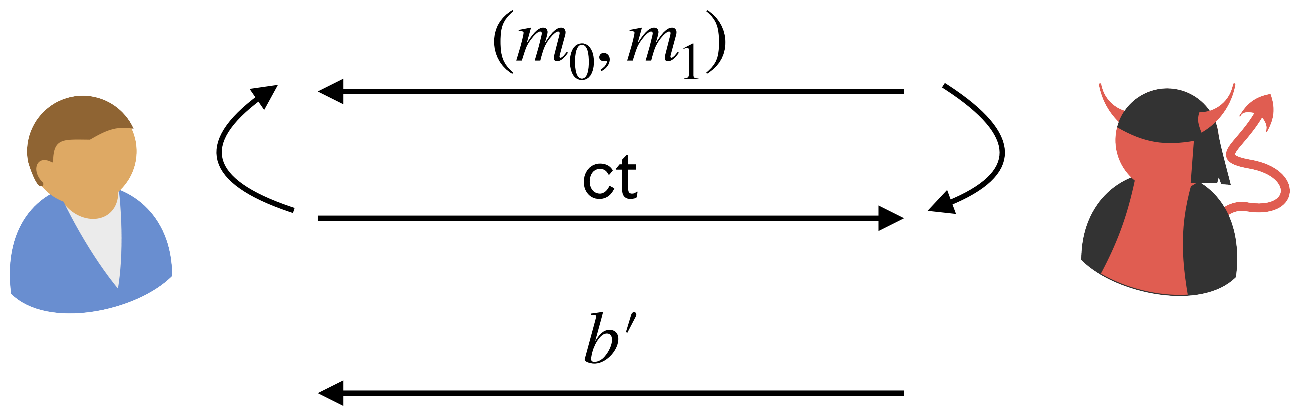
Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

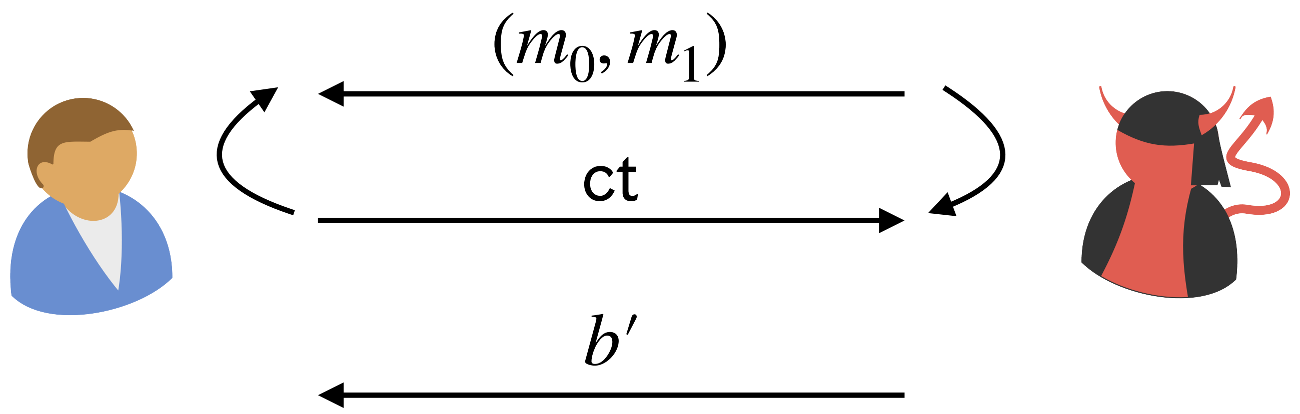
H_0 $k \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



Goal: prove that $H_0 \overset{c}{\approx} H_?$, i.e. that

$$\left| \Pr\left[W_0\right] - \Pr\left[W_?\right] \right| \leq \mathsf{negl}(\lambda)$$

$H_?$ $k \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

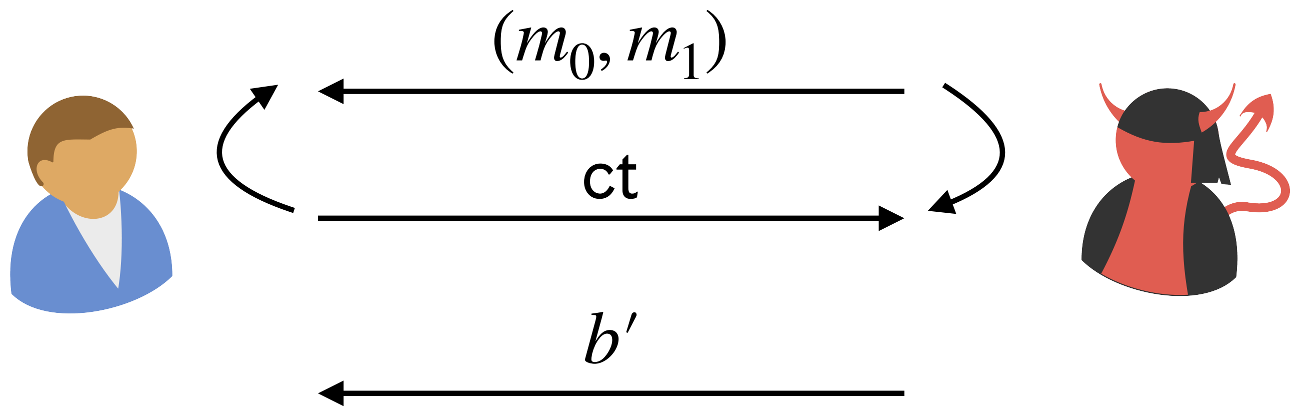
Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

H_0

$k \leftarrow$ KeyGen(1^λ)

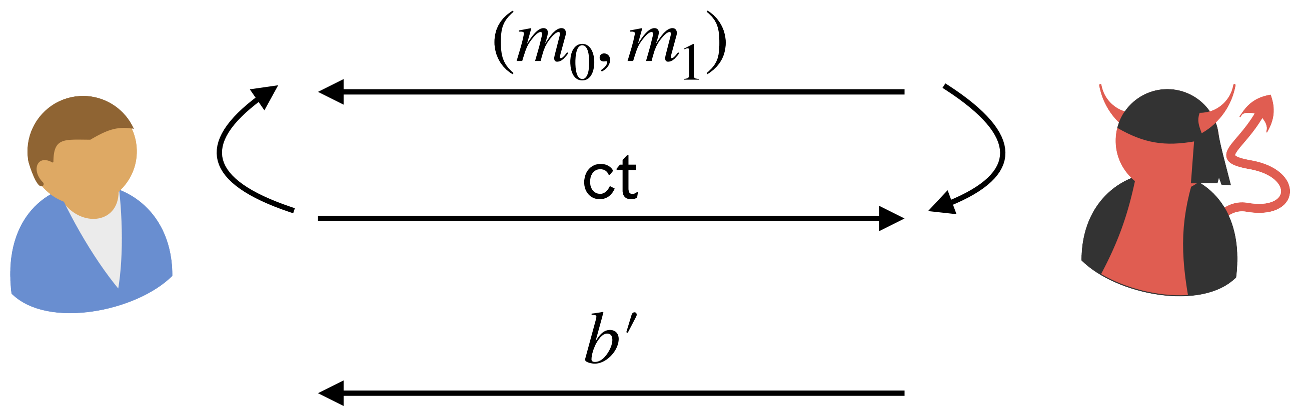
ct $\leftarrow$ Enc(k, m_0)



$H_?$

$k \leftarrow$ KeyGen(1^λ)

ct $\leftarrow$ Enc(k, m_1)



Proof of Security

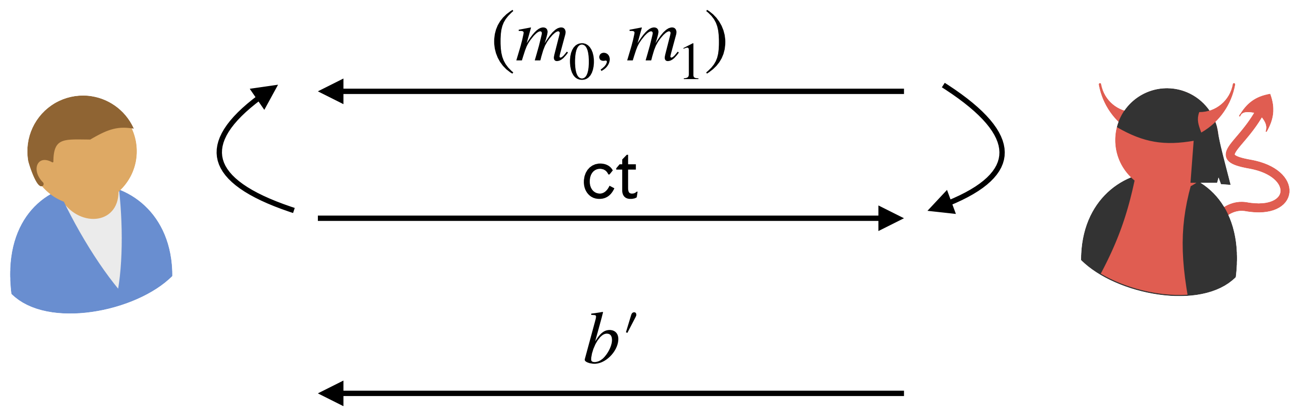
Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

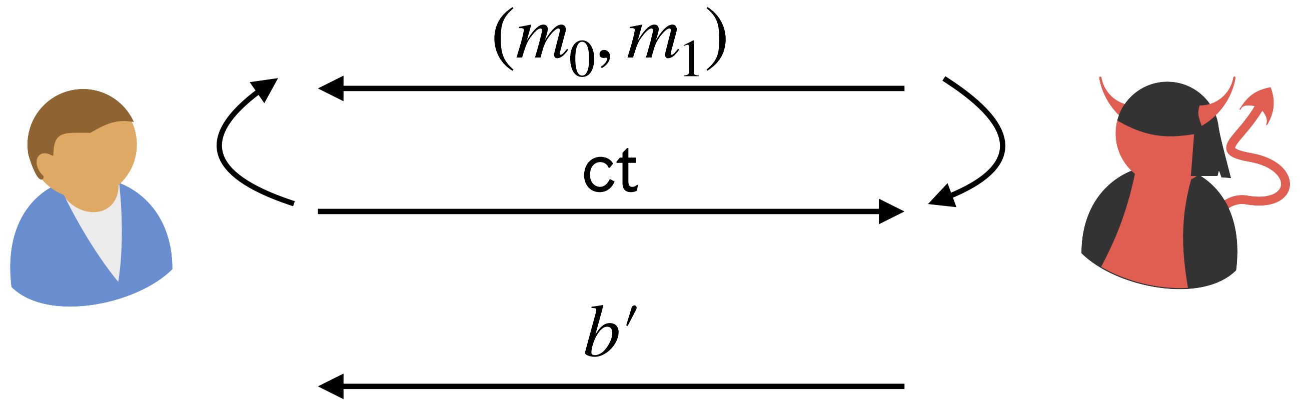
$\mathsf{Enc}(\mathsf{k},\mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k,(c,x)) : m' := F_k(x) \oplus c$

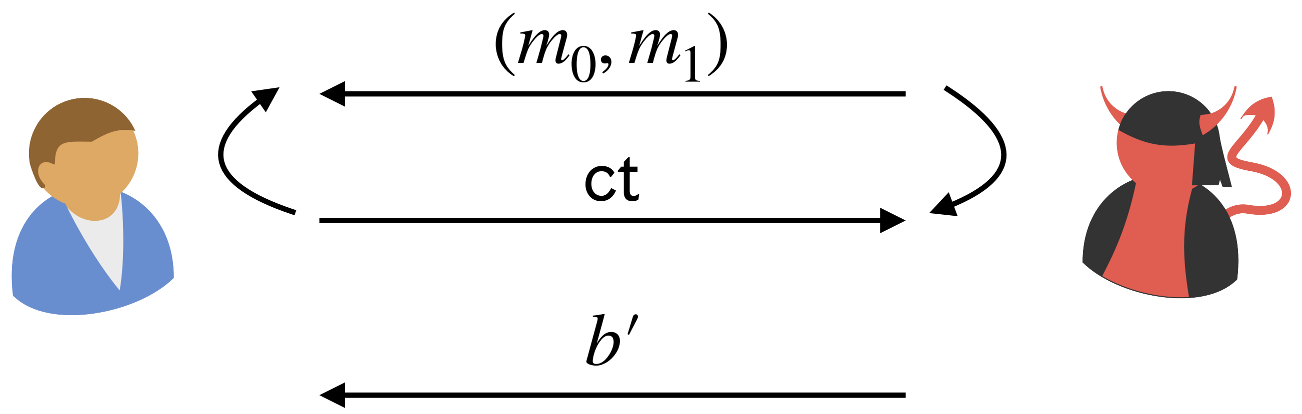
H_0 $k \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



H_1 Some small change



$H_?$ $k \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

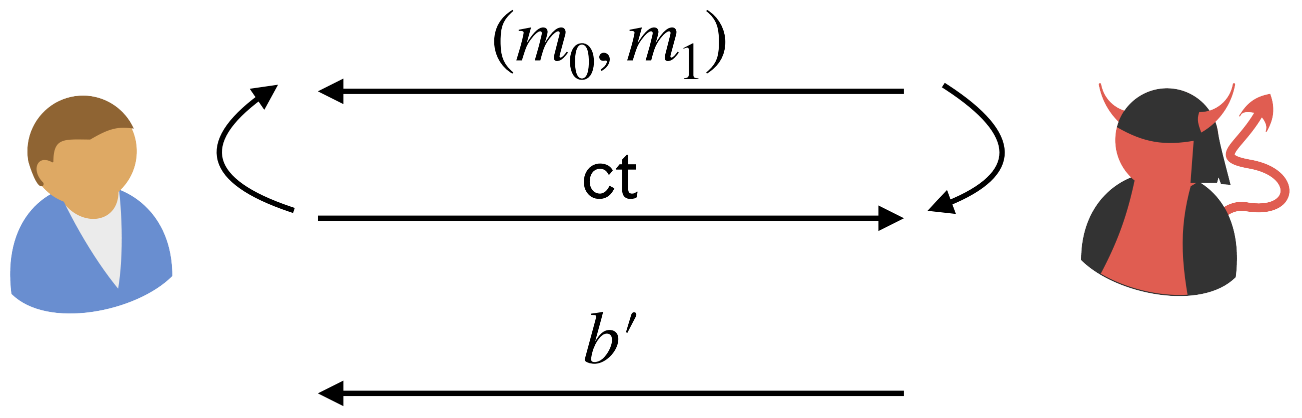
$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

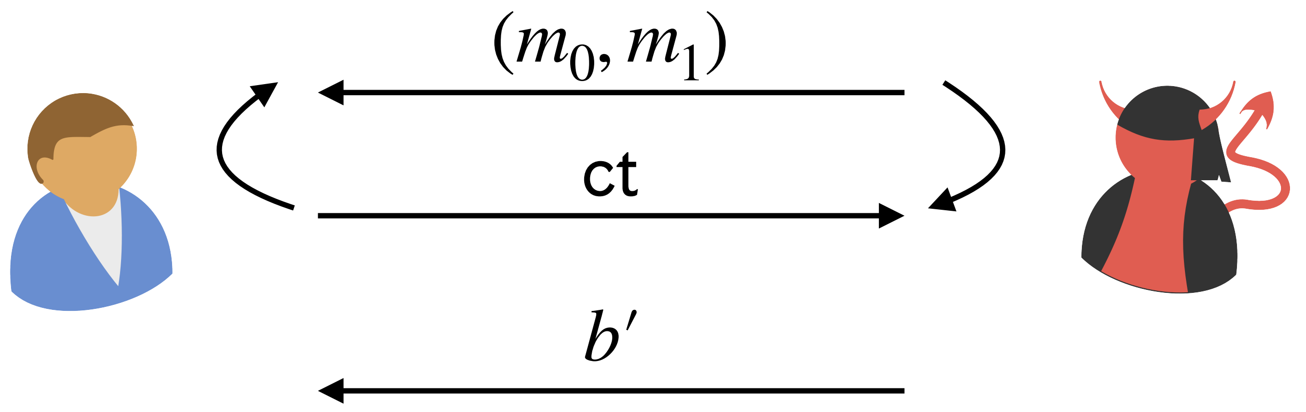
$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



H_1

Some small change

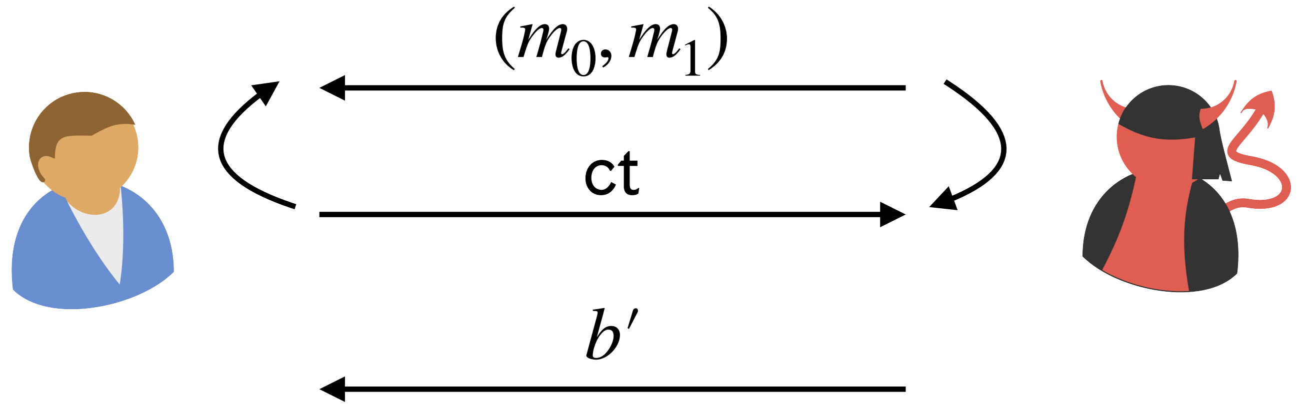


Prove that $H_0 \overset{c}{\approx} H_1$

$H_?$

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

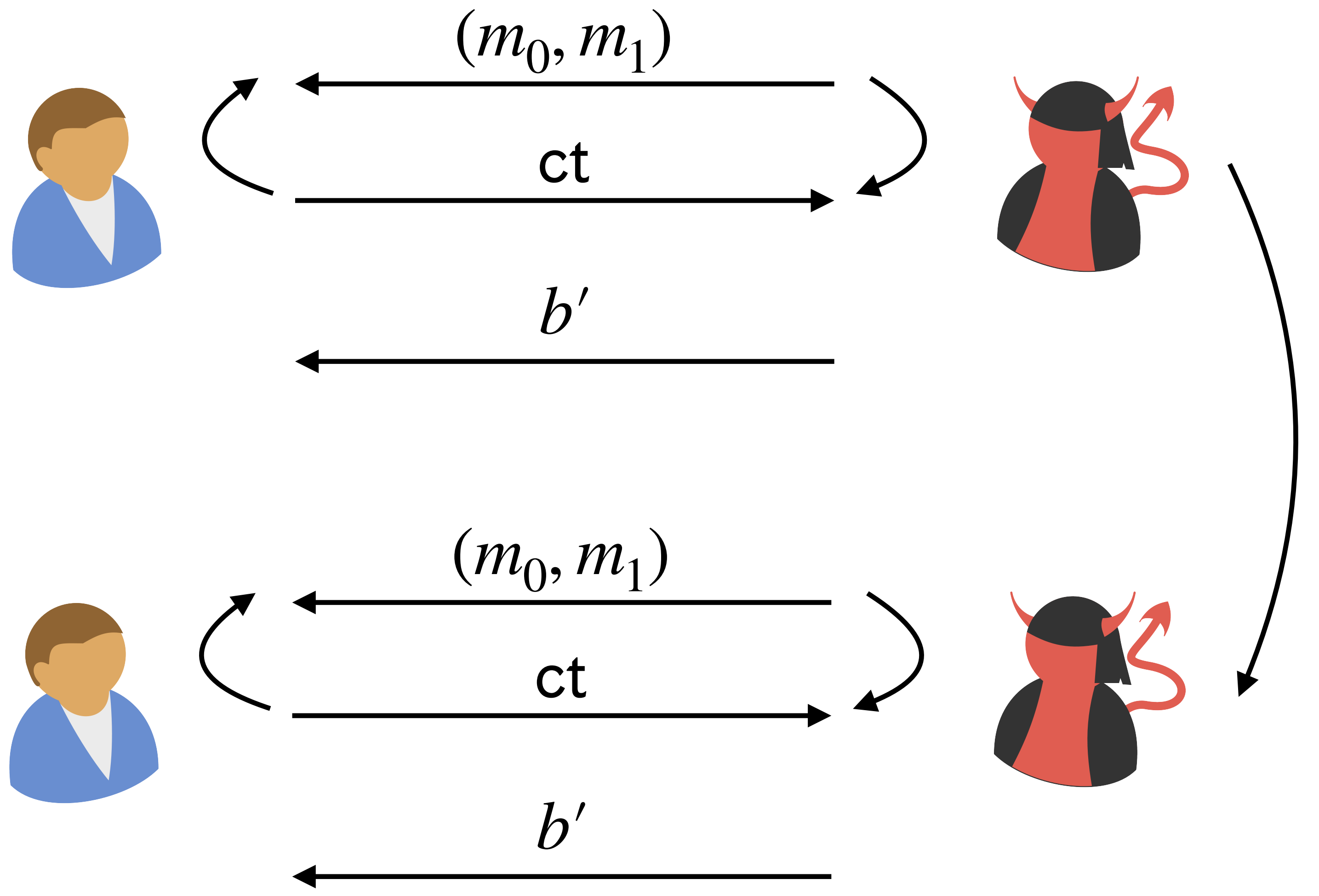
$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$; $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0: $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$

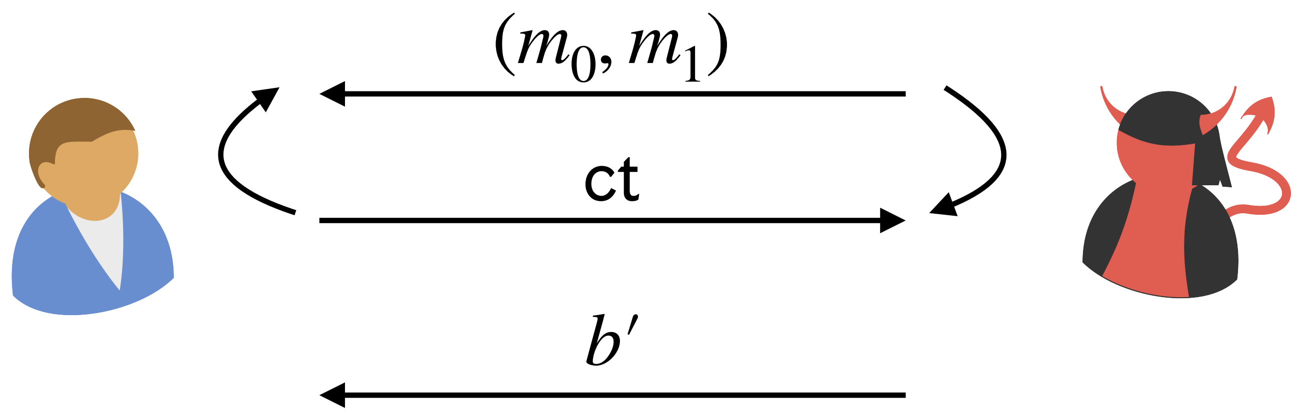
H_1: Some small change



Prove that $H_0 \overset{c}{\approx} H_1$

Prove that $\left|\Pr\left[W_0\right] - \Pr\left[W_1\right]\right| \leq \mathsf{negl}(\lambda)$

$H_?$: $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



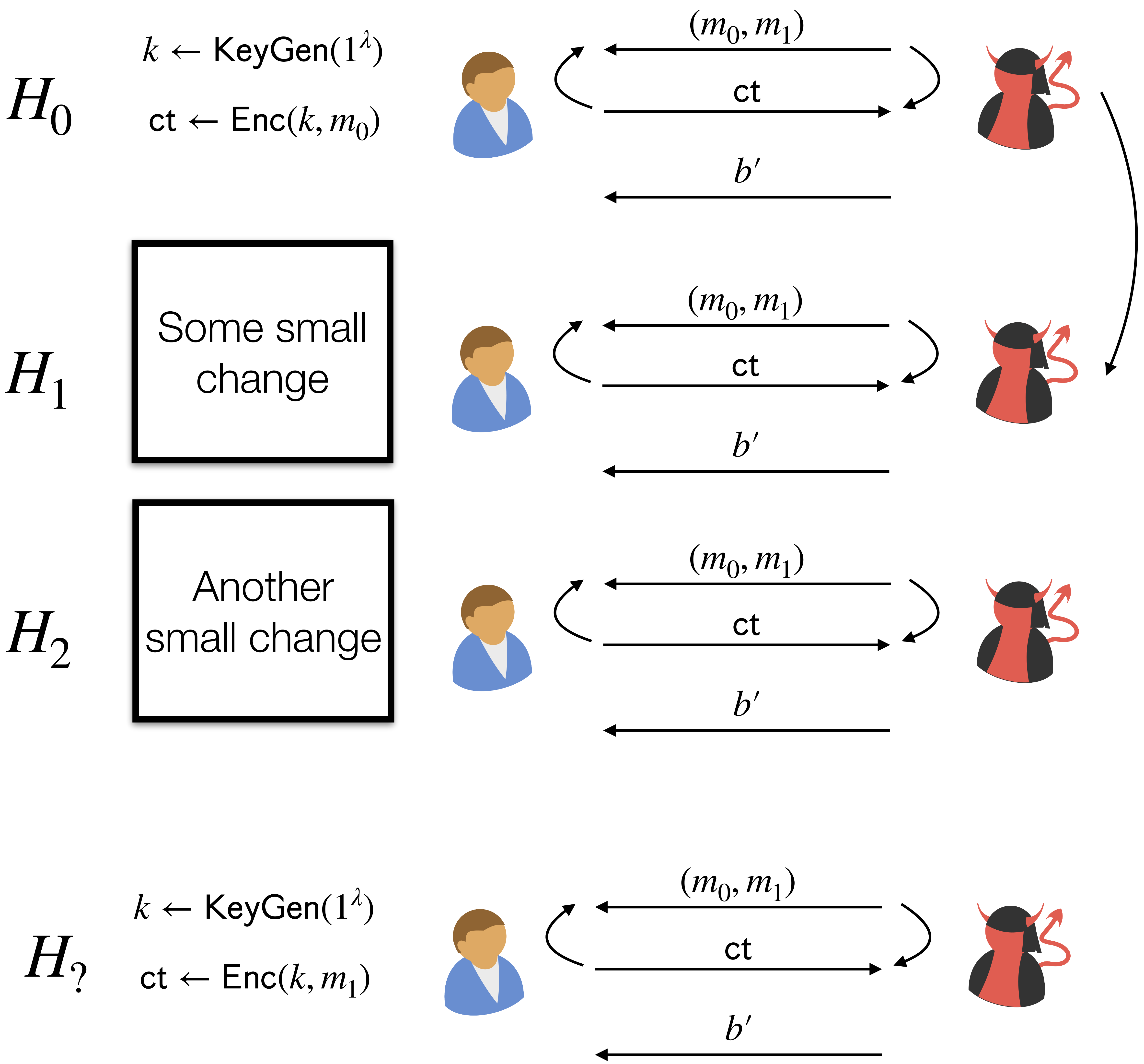
Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$



Prove that $H_0 \overset{c}{\approx} H_1$

Prove that $\left| \Pr\left[W_0\right] - \Pr\left[W_1\right] \right| \leq \text{negl}(\lambda)$

Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

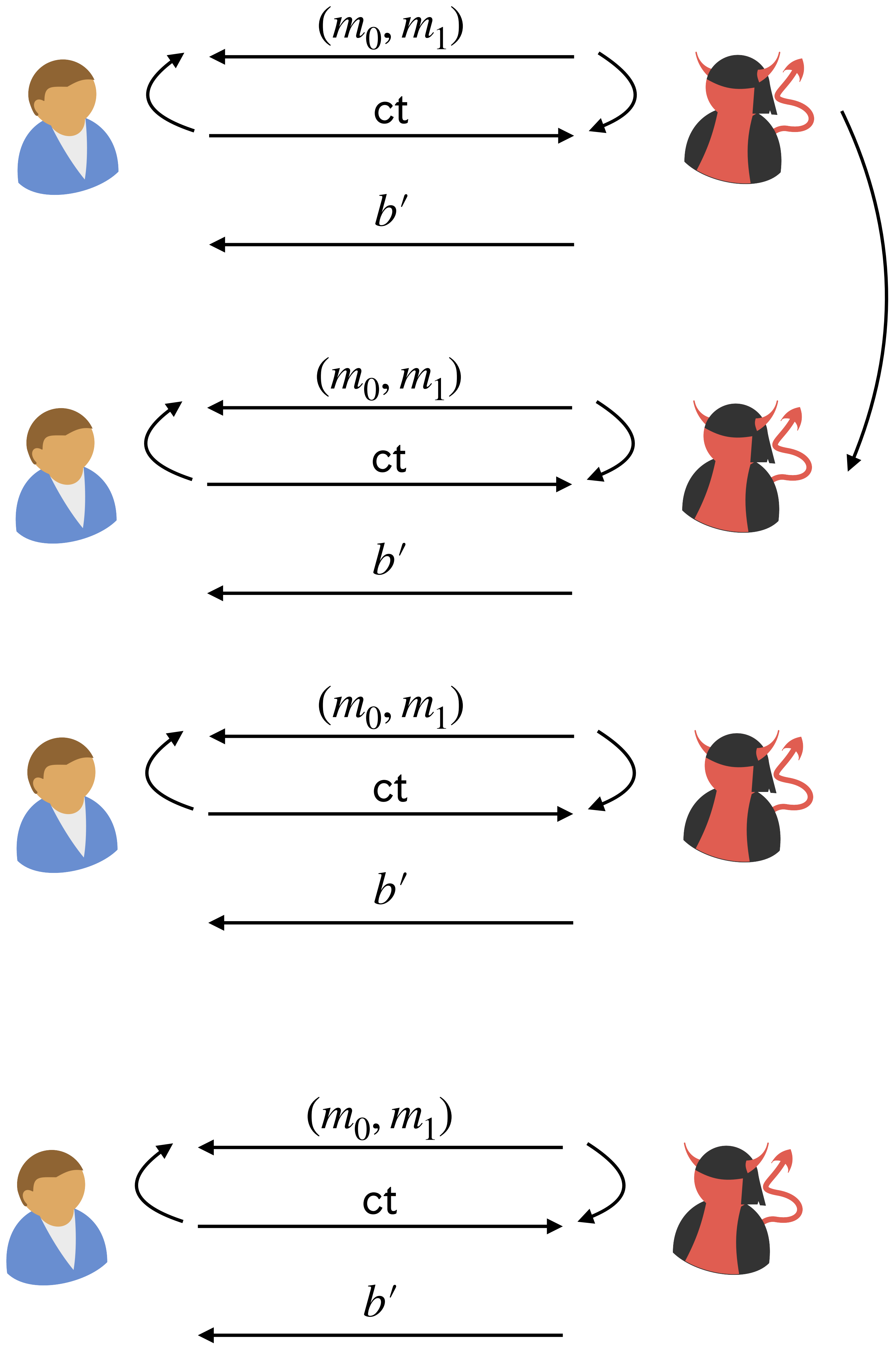
$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0 $\quad k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$

H_1 $\quad$ Some small change

H_2 $\quad$ Another small change

$H_?$ $\quad k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Prove that $H_0 \overset{c}{\approx} H_1$

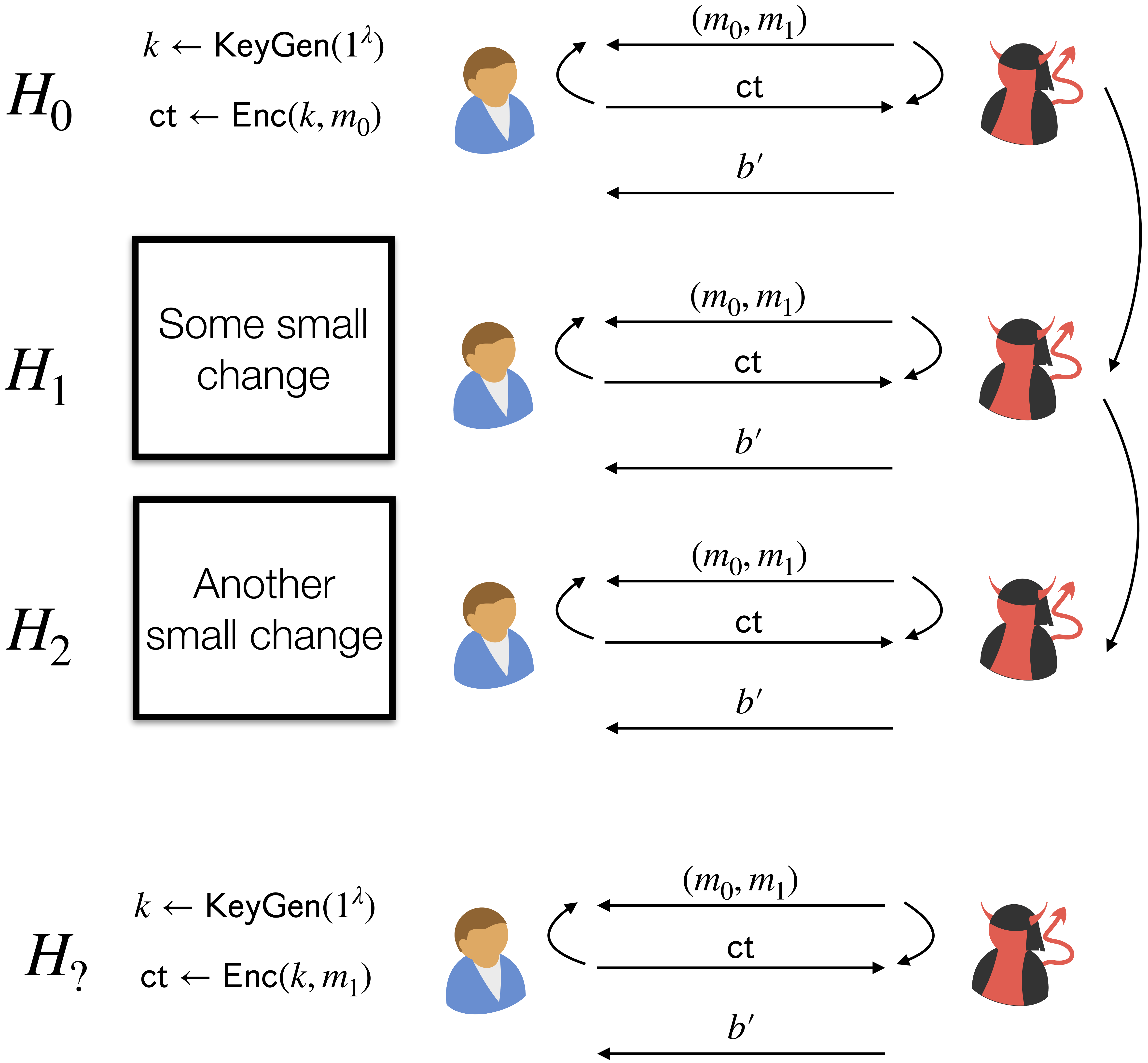
Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$



Prove that $H_0 \overset{c}{\approx} H_1$

Prove that $H_1 \overset{c}{\approx} H_2$

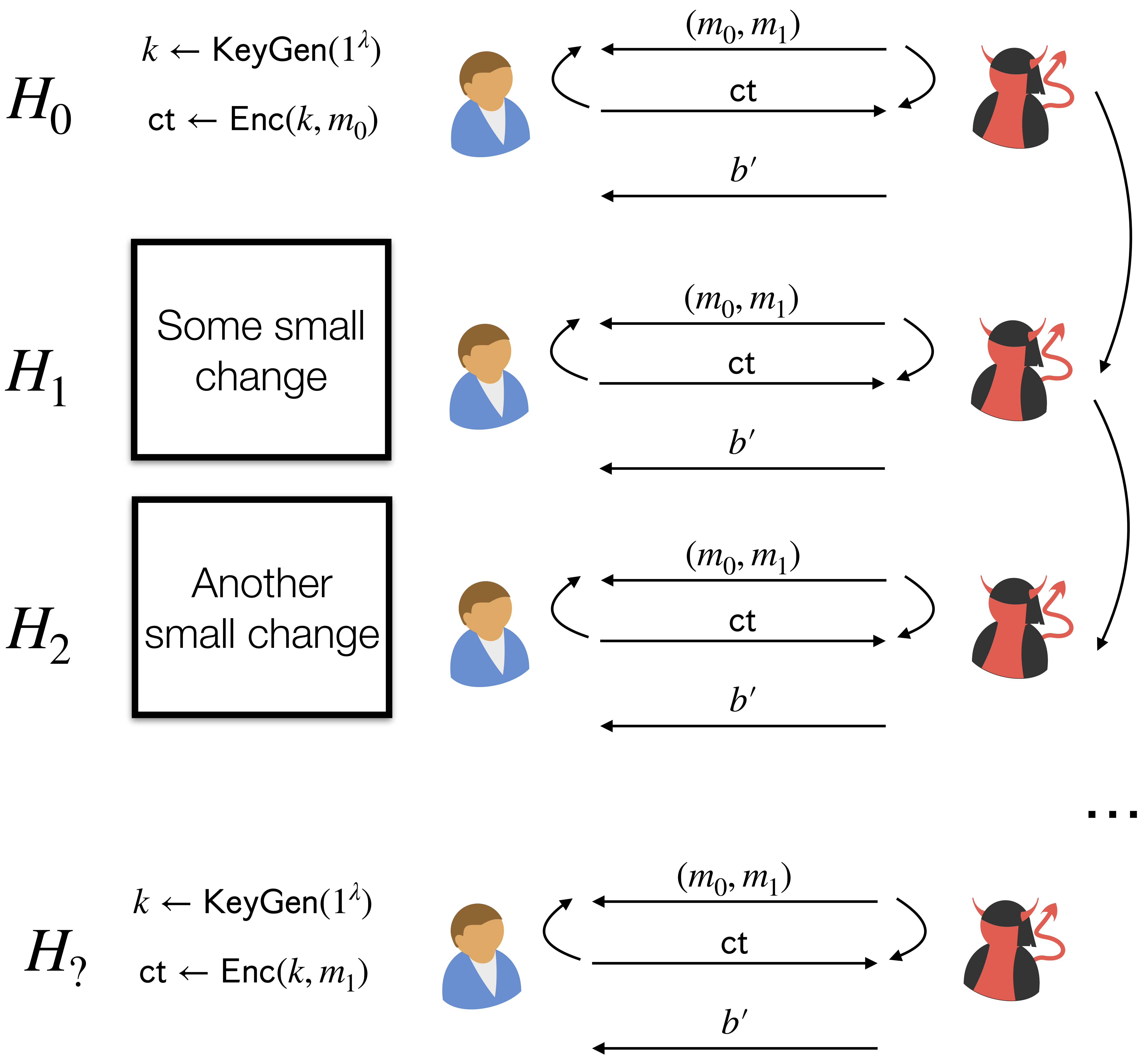
Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$



Prove that $H_0 \overset{c}{\approx} H_1$

Prove that $H_1 \overset{c}{\approx} H_2$

Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

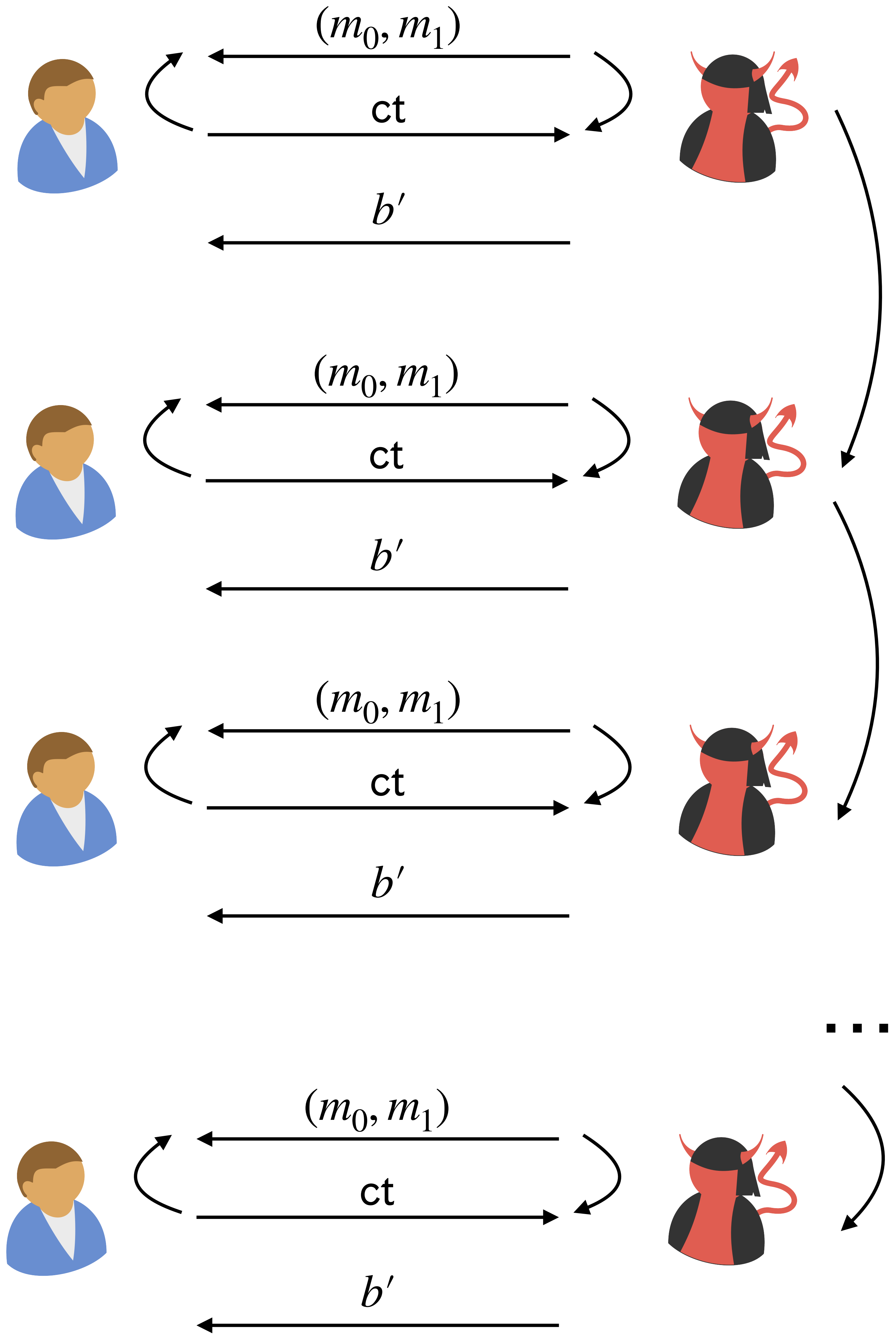
H_0 — $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$

H_1 — Some small change

H_2 — Another small change

...

$H_?$ — $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Prove that $H_0 \overset{c}{\approx} H_1$

Prove that $H_1 \overset{c}{\approx} H_2$

Proof of Security

Let W_i be the probability that $\mathscr{A}$ outputs 1 in H_i

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

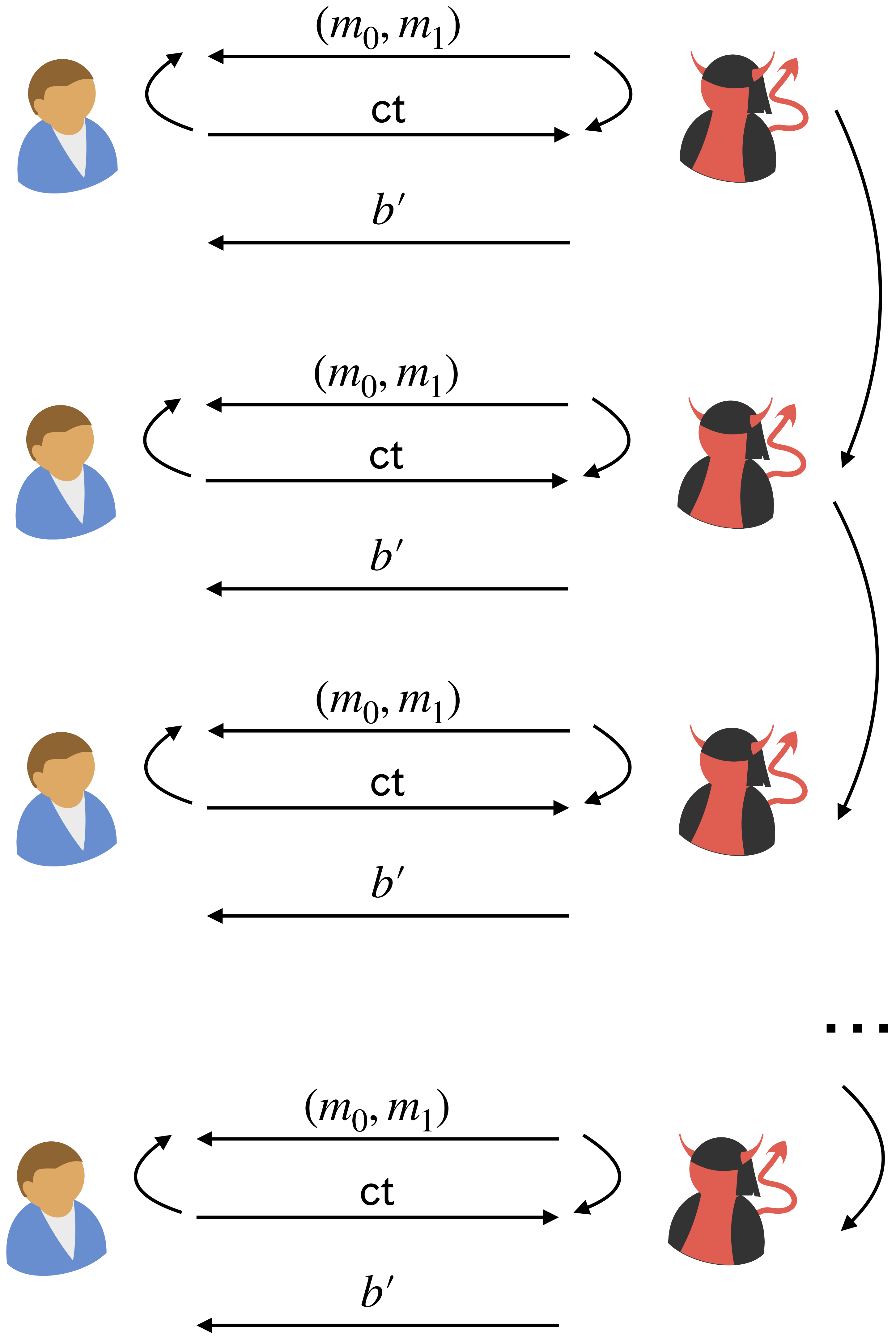
H_0: $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$

H_1: Some small change

H_2: Another small change

...

$H_?$: $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1)$



Prove that $H_0 \overset{c}{\approx} H_1$

Prove that $H_1 \overset{c}{\approx} H_2$

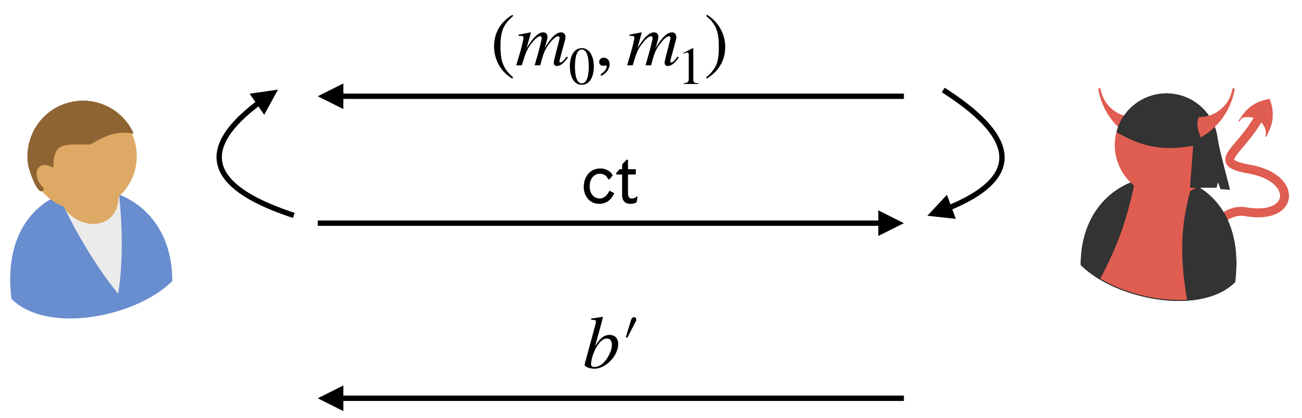
By hybrid lemma: $H_0 \overset{c}{\approx} H_?$

Proof of Security

H_0

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

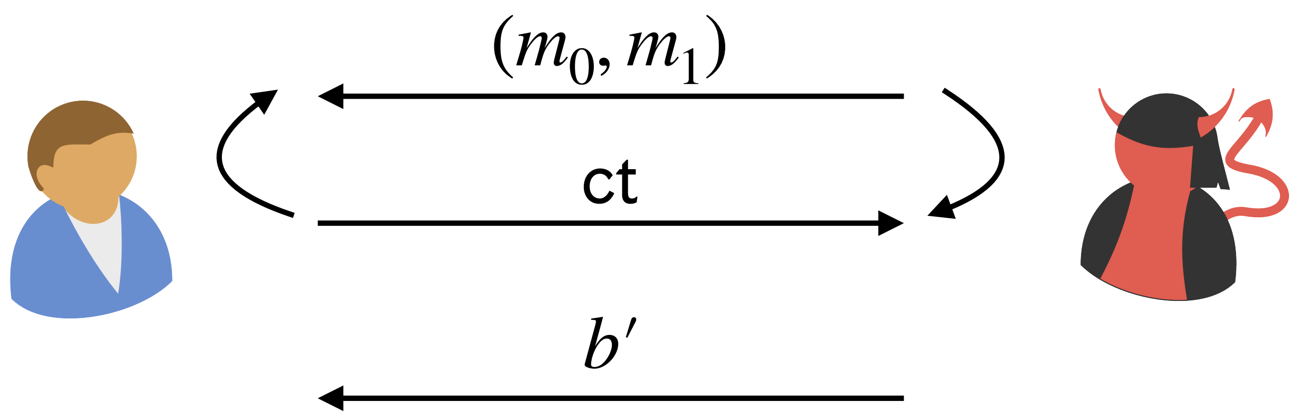
$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

Proof of Security

H_0 $\quad k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0)$



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

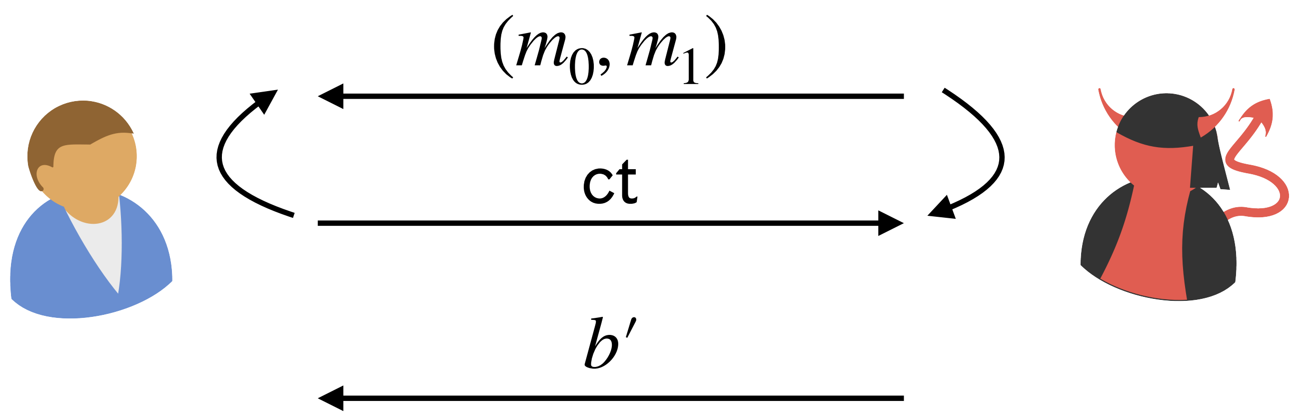
Proof of Security

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

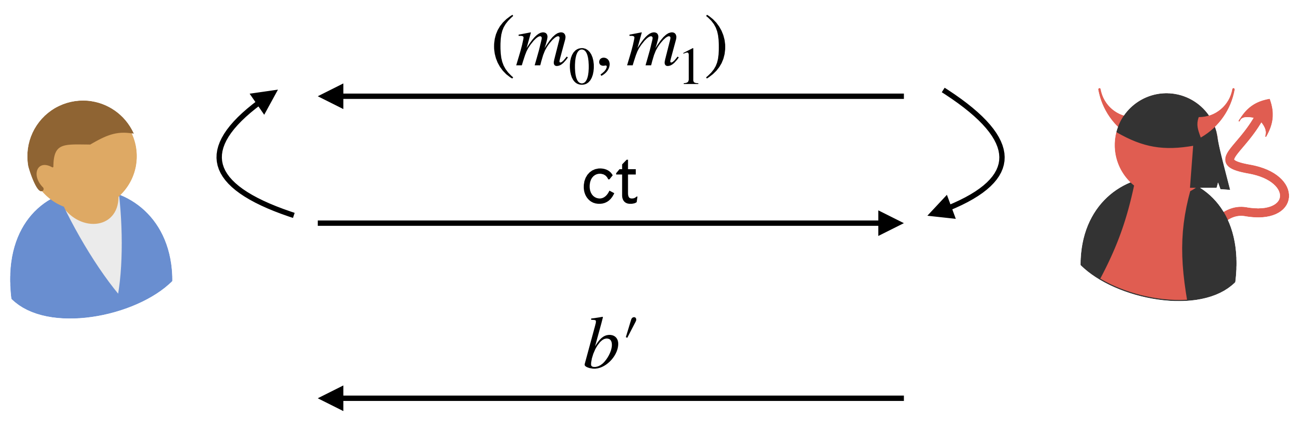
Proof of Security

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

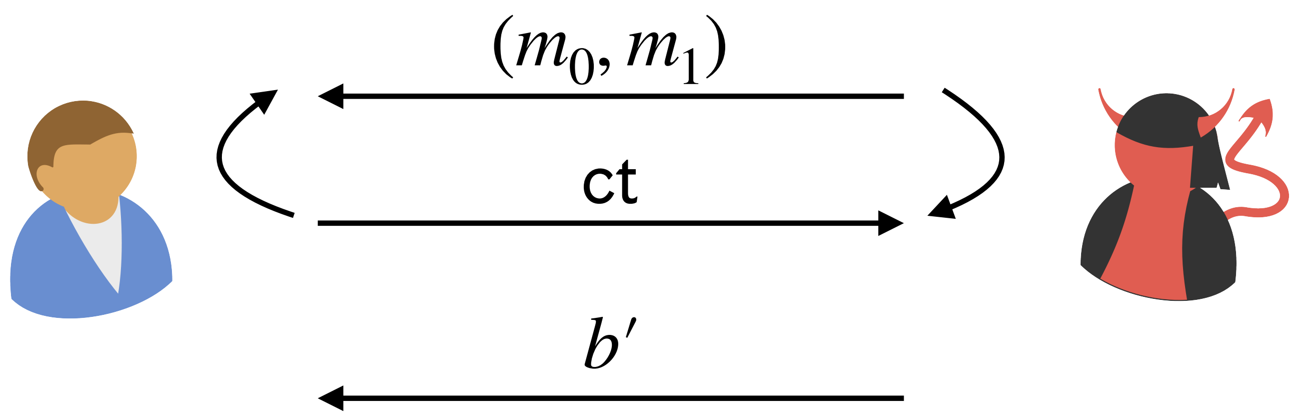
$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$



H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

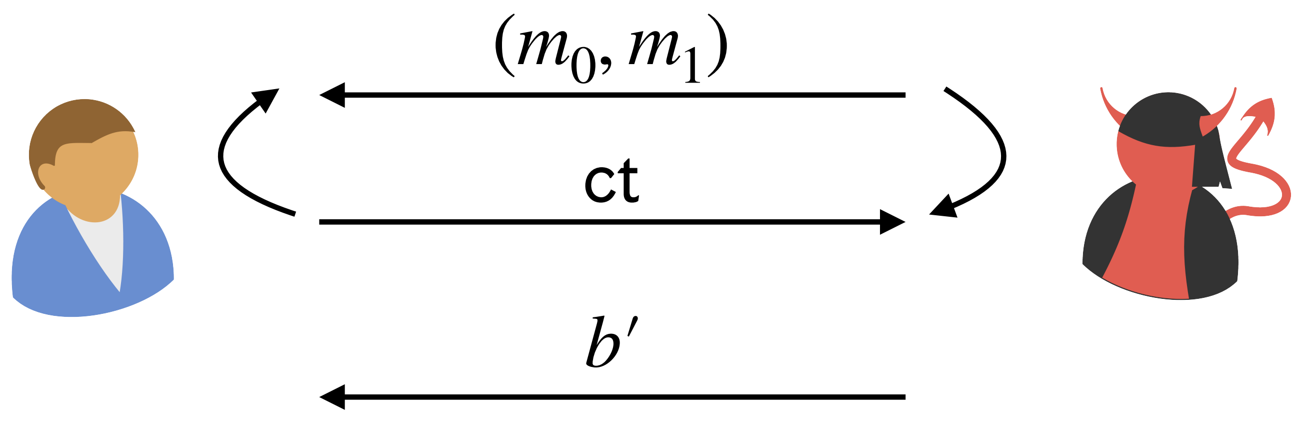
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$



Proof of Security

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m_0, x)$

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

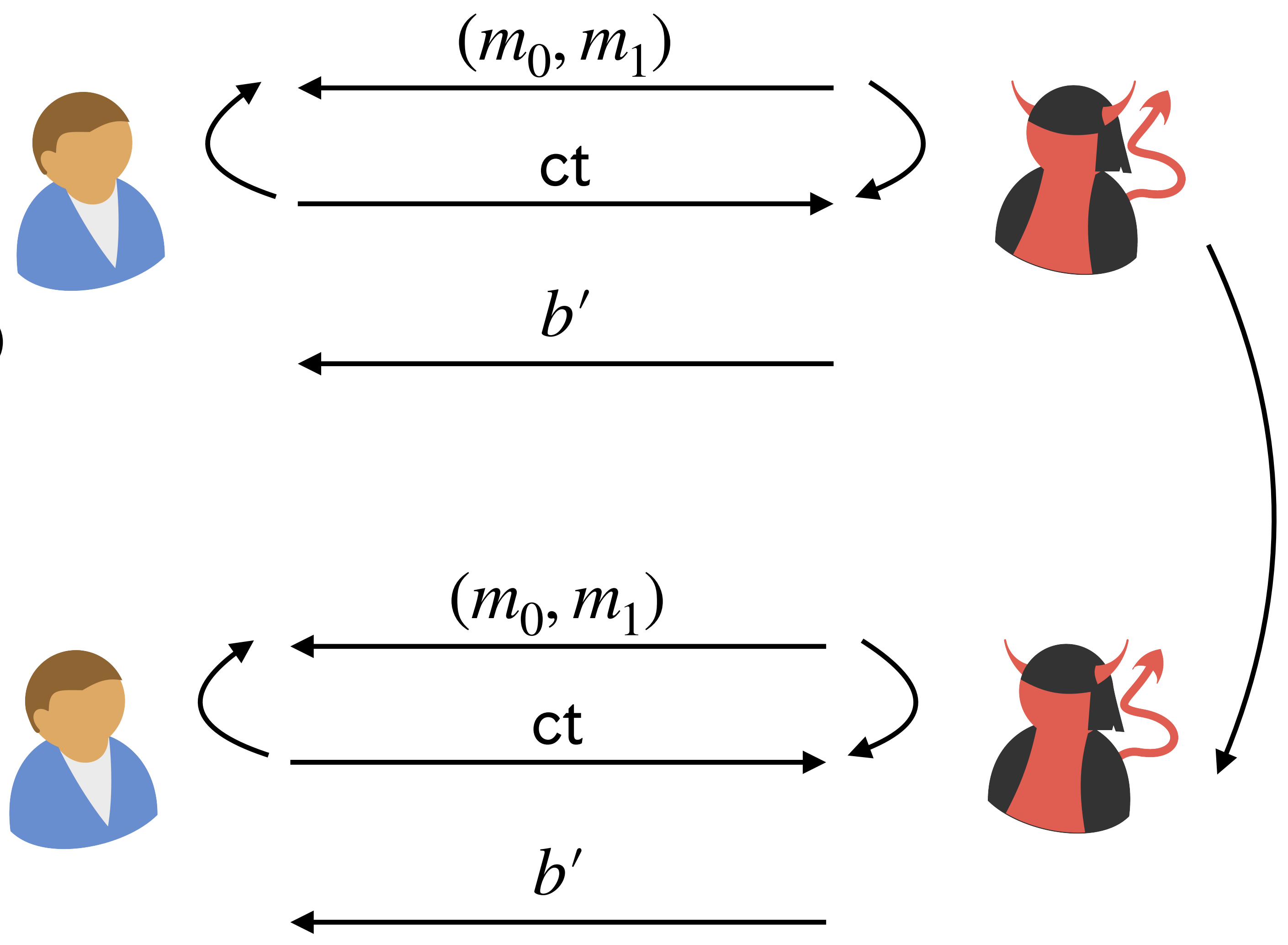
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

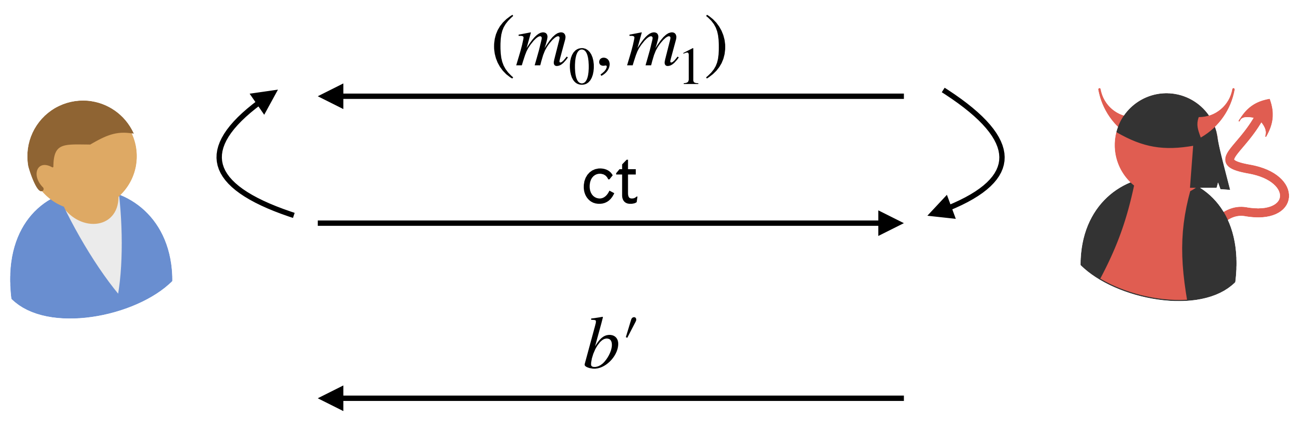
Proof of Security

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m_0, x)$



H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

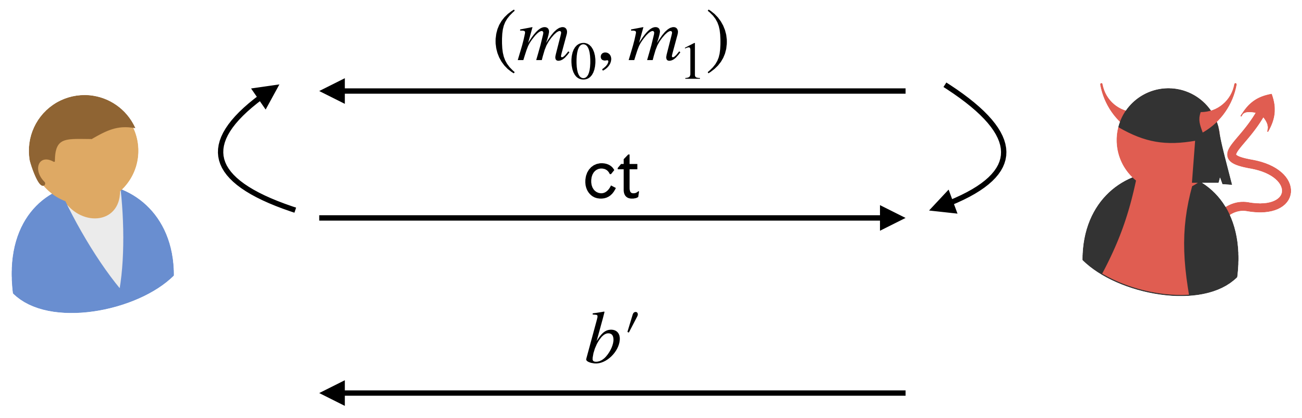
if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

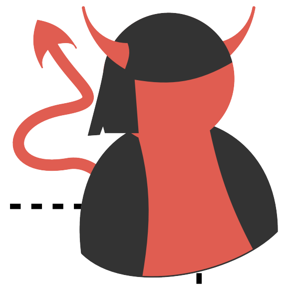
Ch_F

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

A_F



A_{H_0,H_1}

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(k, m) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

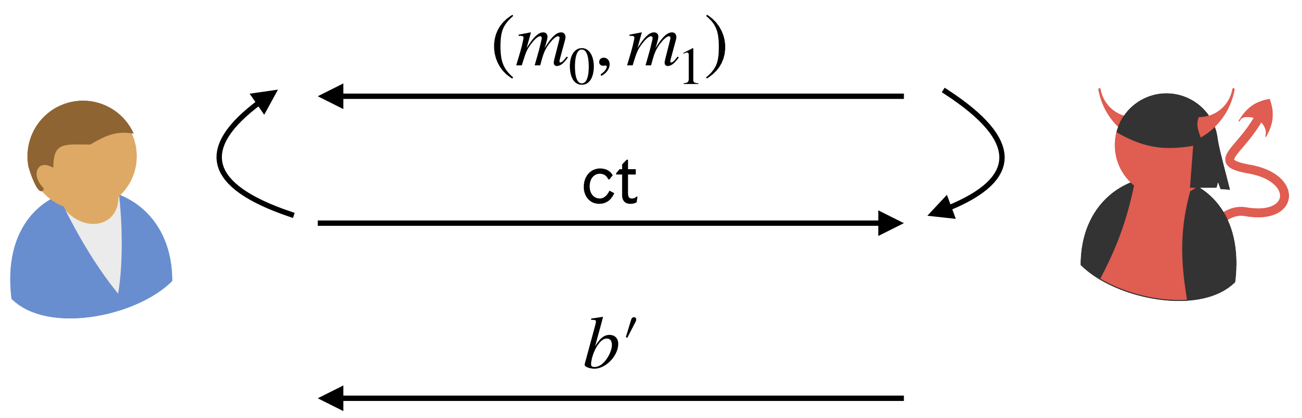
$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

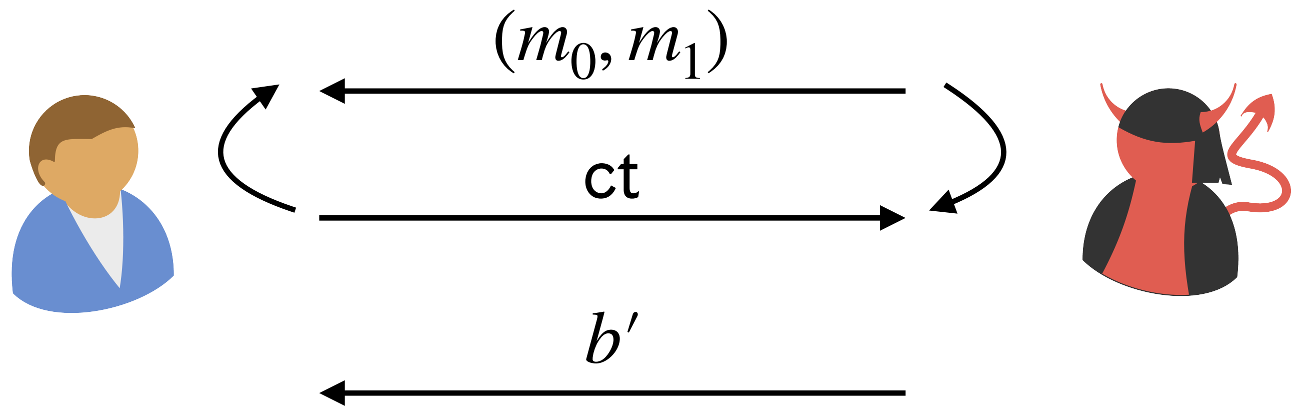
if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

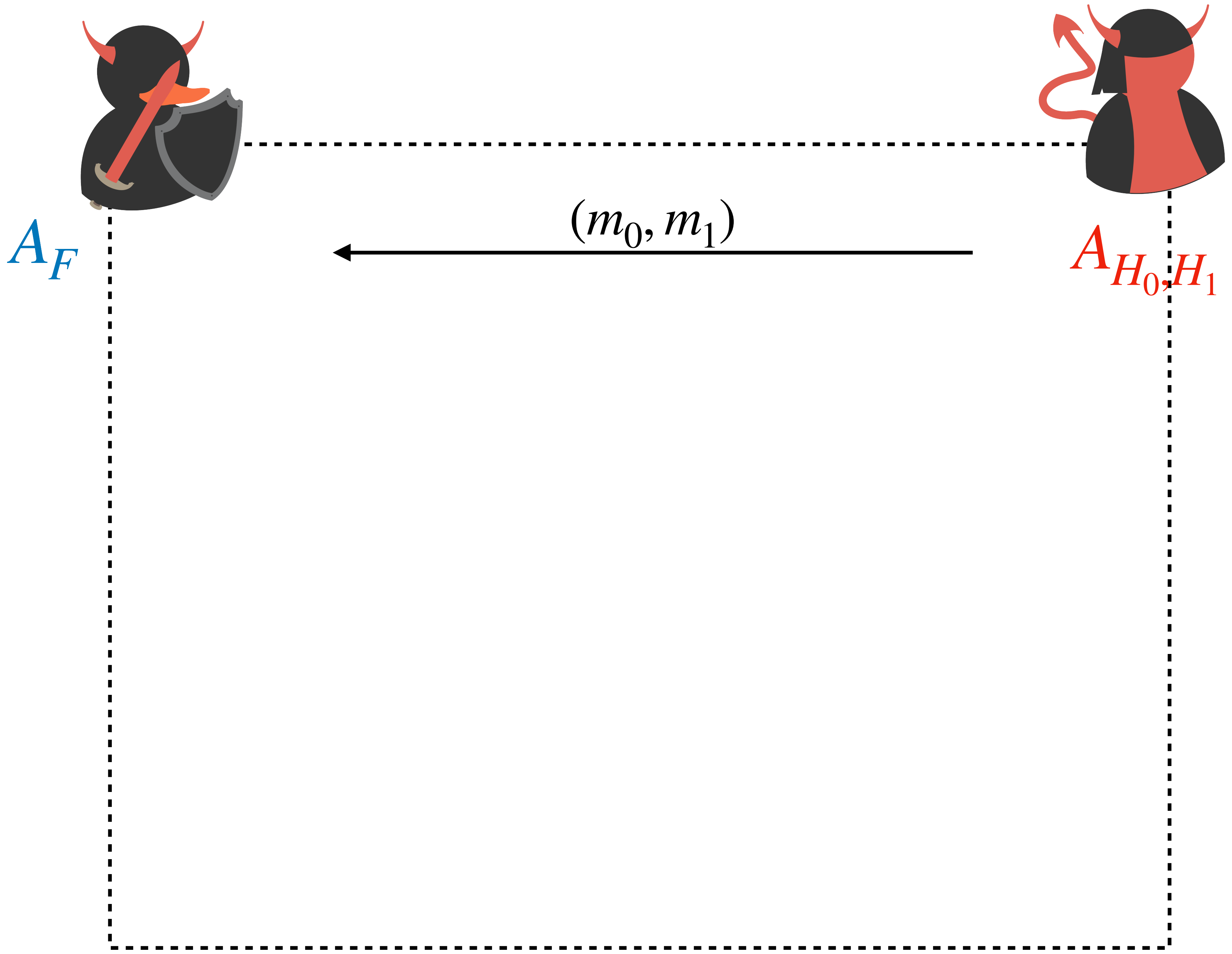
$T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$



Ch_F



Proof of Security

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m_0, x)$

(m_0, m_1)

ct

b'

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$

(m_0, m_1)

ct

b'

Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

Ch_F

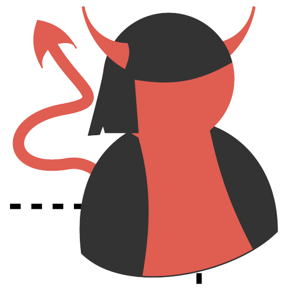
KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

A_F



A_{H_0,H_1}

(m_0, m_1)

$x \xleftarrow{\$} \{0,1\}^\lambda$

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

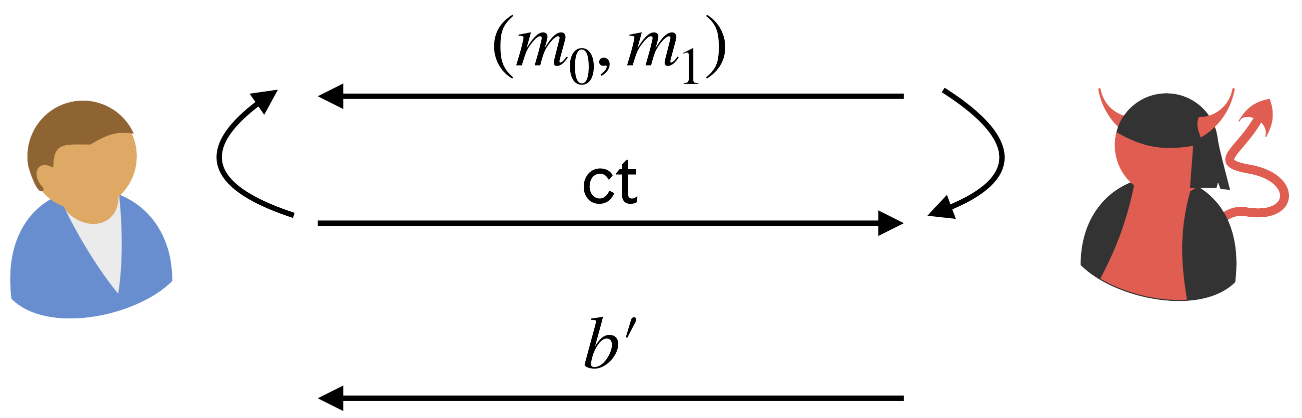
$\mathsf{Enc}(\mathsf{k}, \mathsf{m})$: $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m_0, x)$

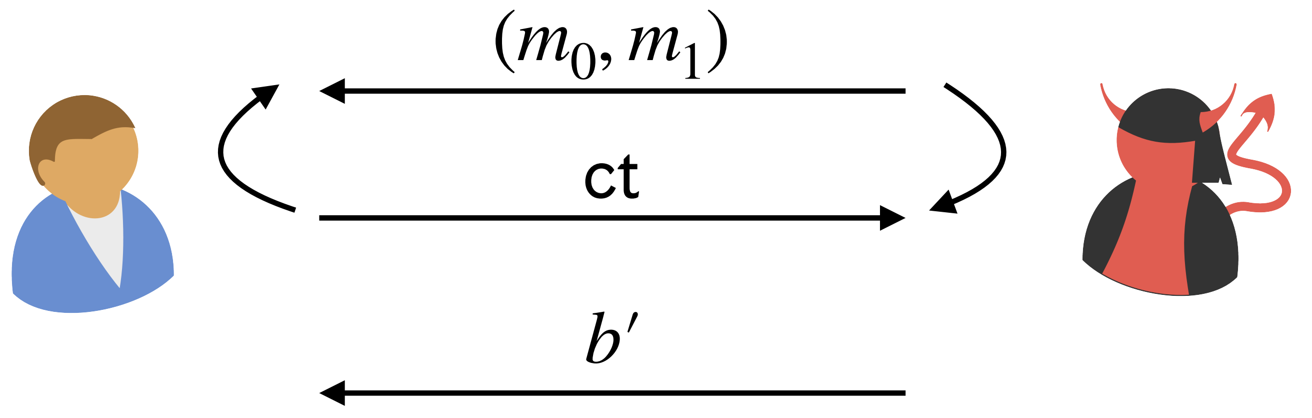


H_1

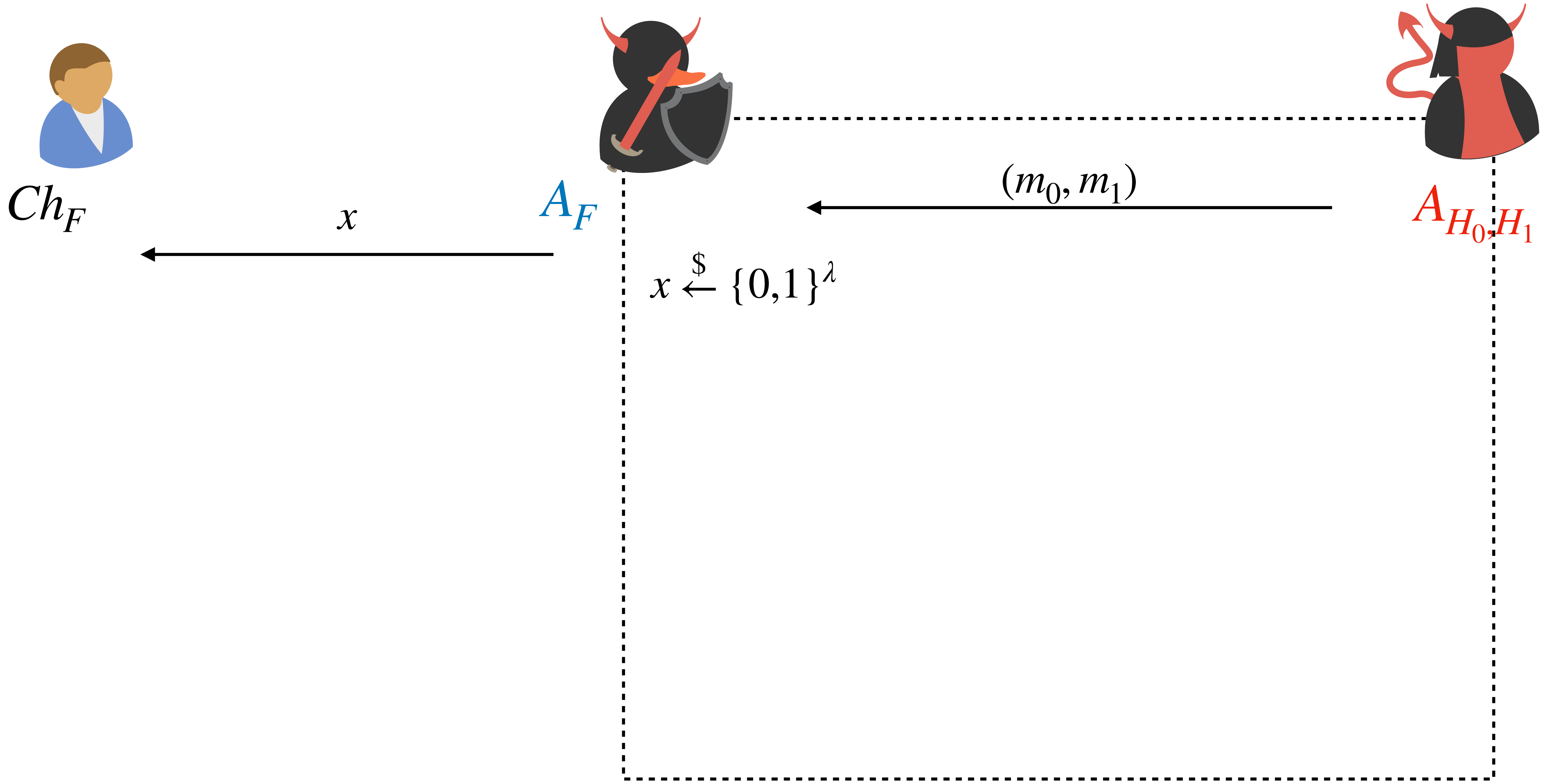
$T = \{\}$
$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$
$r \xleftarrow{\$} \{0,1\}^\lambda$
$T[x] = r$
$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F



Proof of Security

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

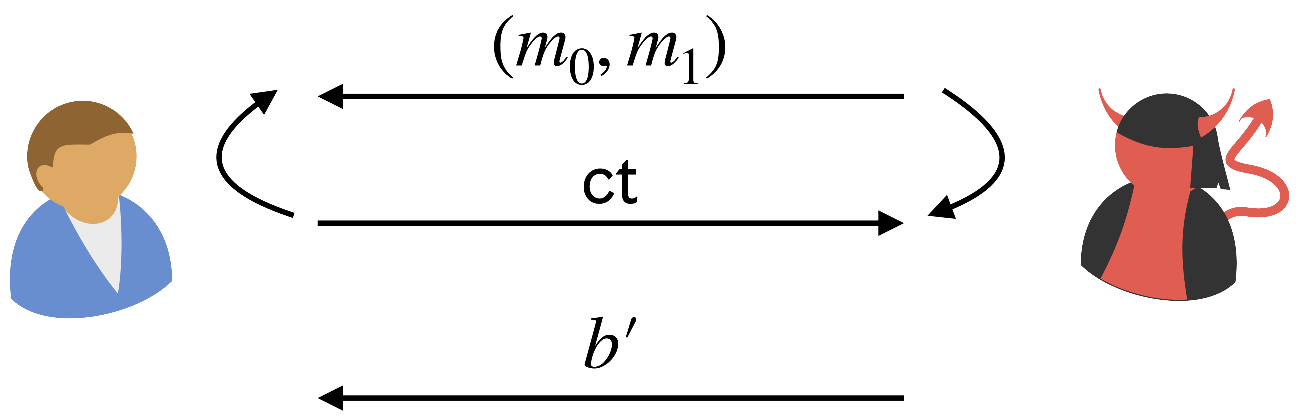
Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m_0, x)$



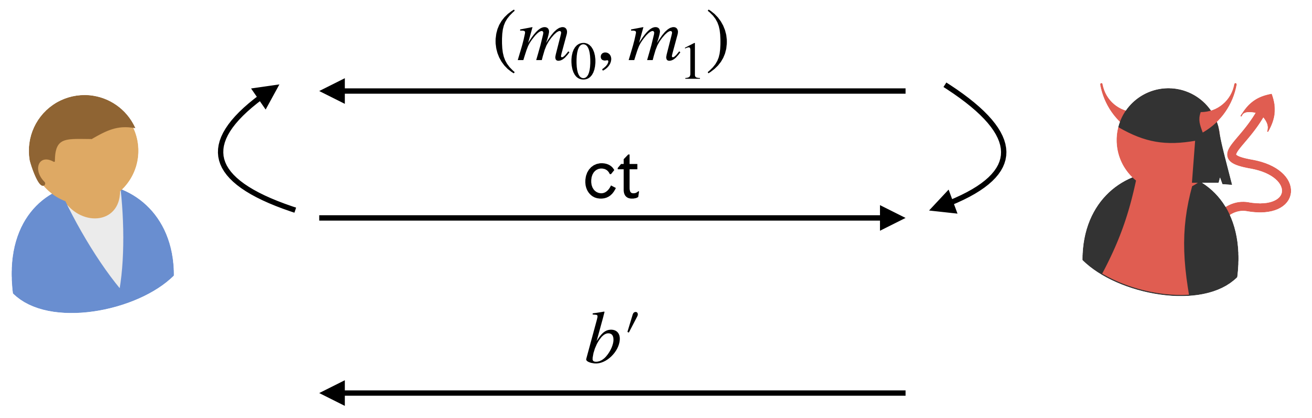
Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

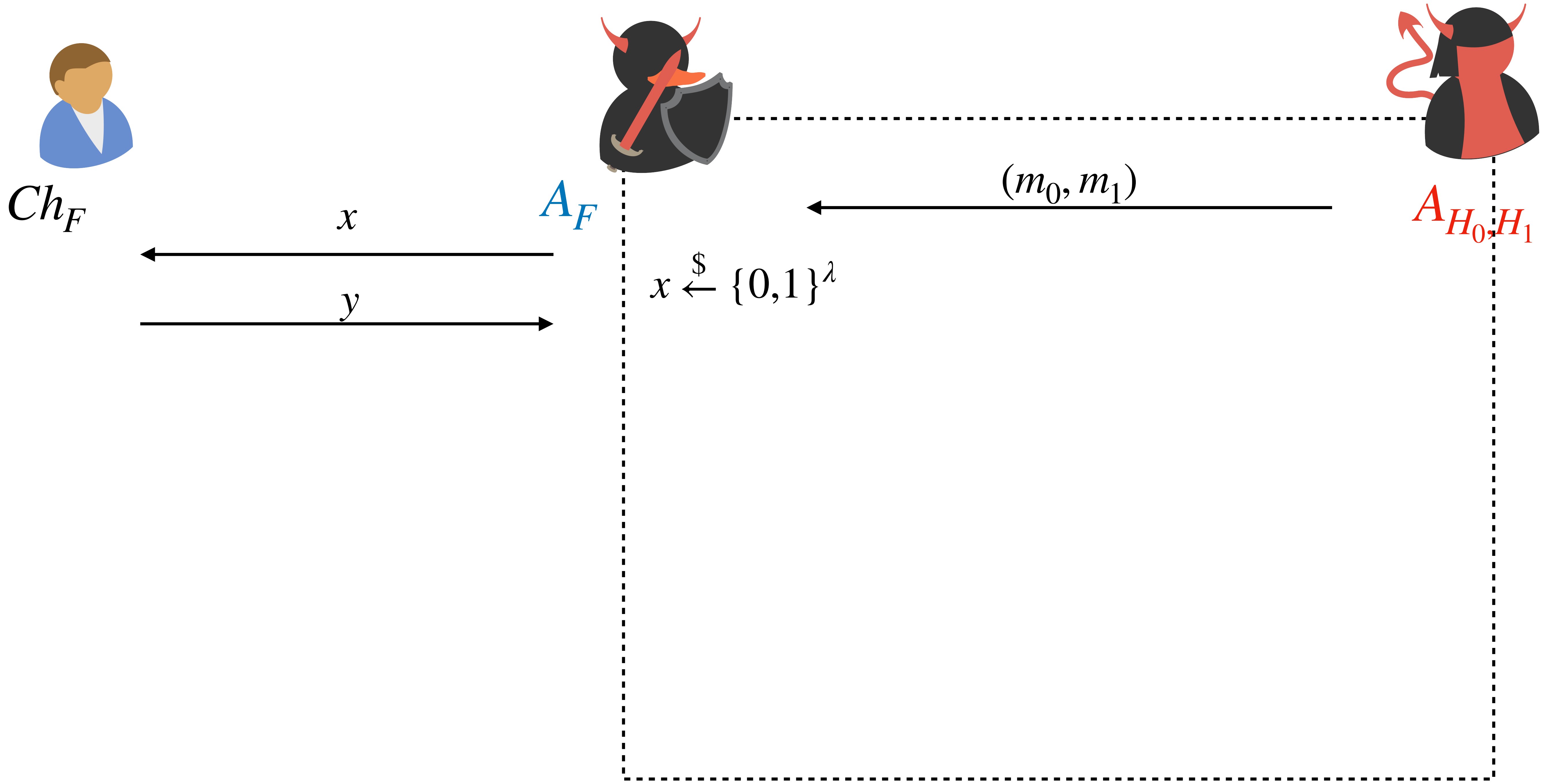
H_1

$T = \{\}$
$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$
$r \xleftarrow{\$} \{0,1\}^\lambda$
$T[x] = r$
$y = T[x]$

ct $:= (y \oplus m_0, x)$





Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m_0, x)$

(m_0, m_1)

ct

b'

H_1

$T = \{\}$
$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$
$r \xleftarrow{\$} \{0,1\}^\lambda$
$T[x] = r$
$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$

(m_0, m_1)

ct

b'

Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

Ch_F

x

y

A_F

A_{H_0,H_1}

(m_0, m_1)

$x \xleftarrow{\$} \{0,1\}^\lambda$

$(y \oplus m_0, x)$

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

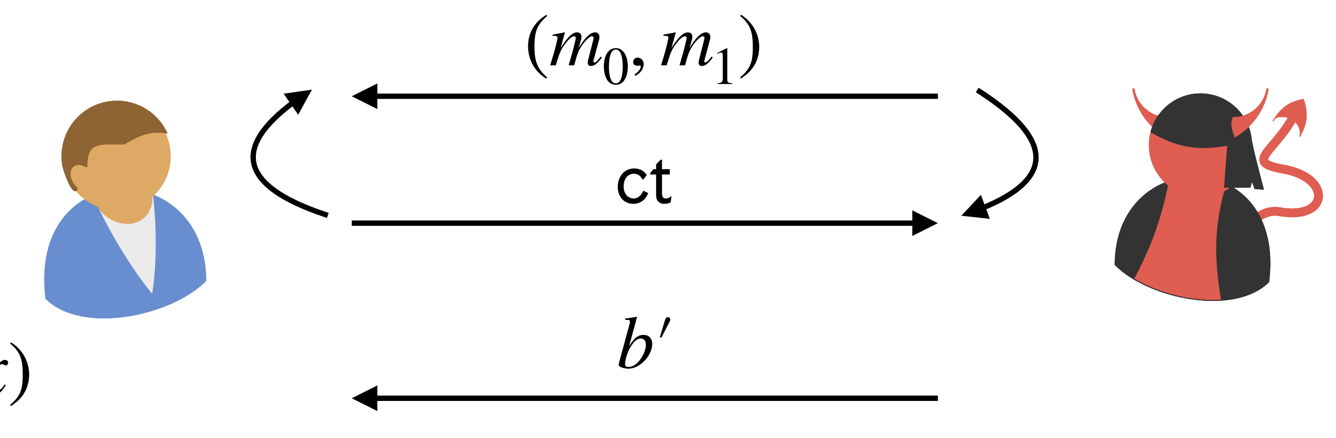
$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m_0, x)$

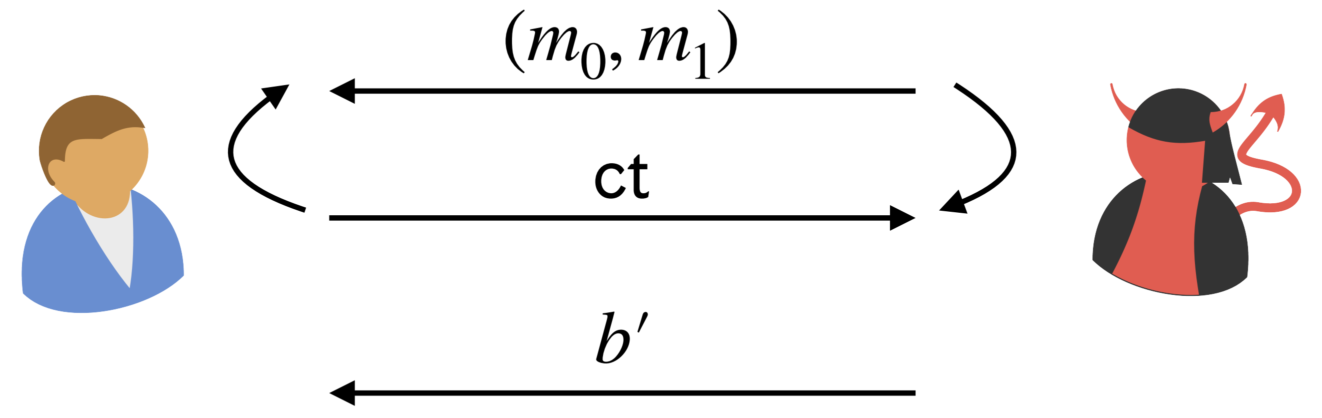


H_1

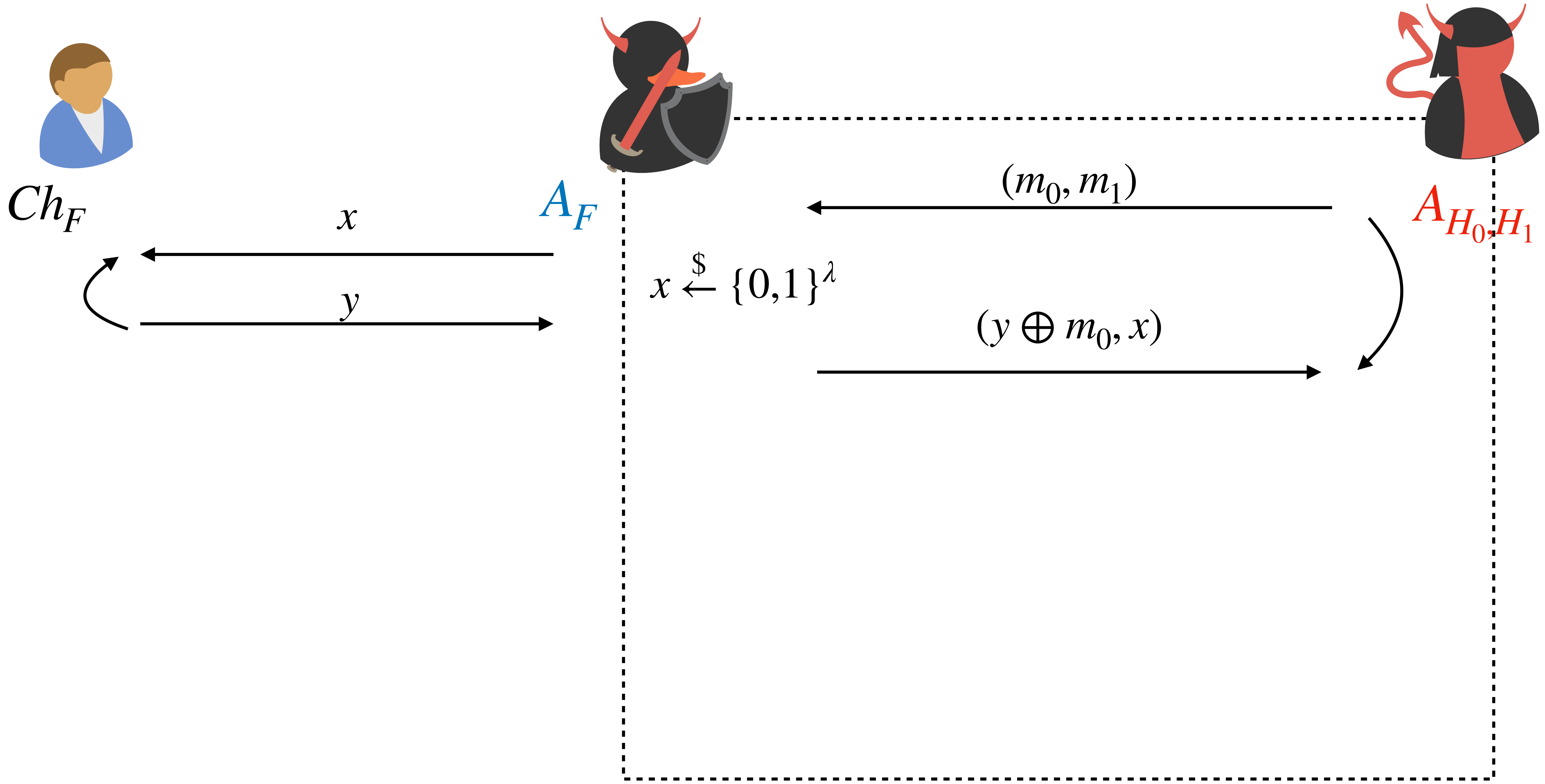
$T = \{\}$
$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$
$r \xleftarrow{\$} \{0,1\}^\lambda$
$T[x] = r$
$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F



Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(k, m) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

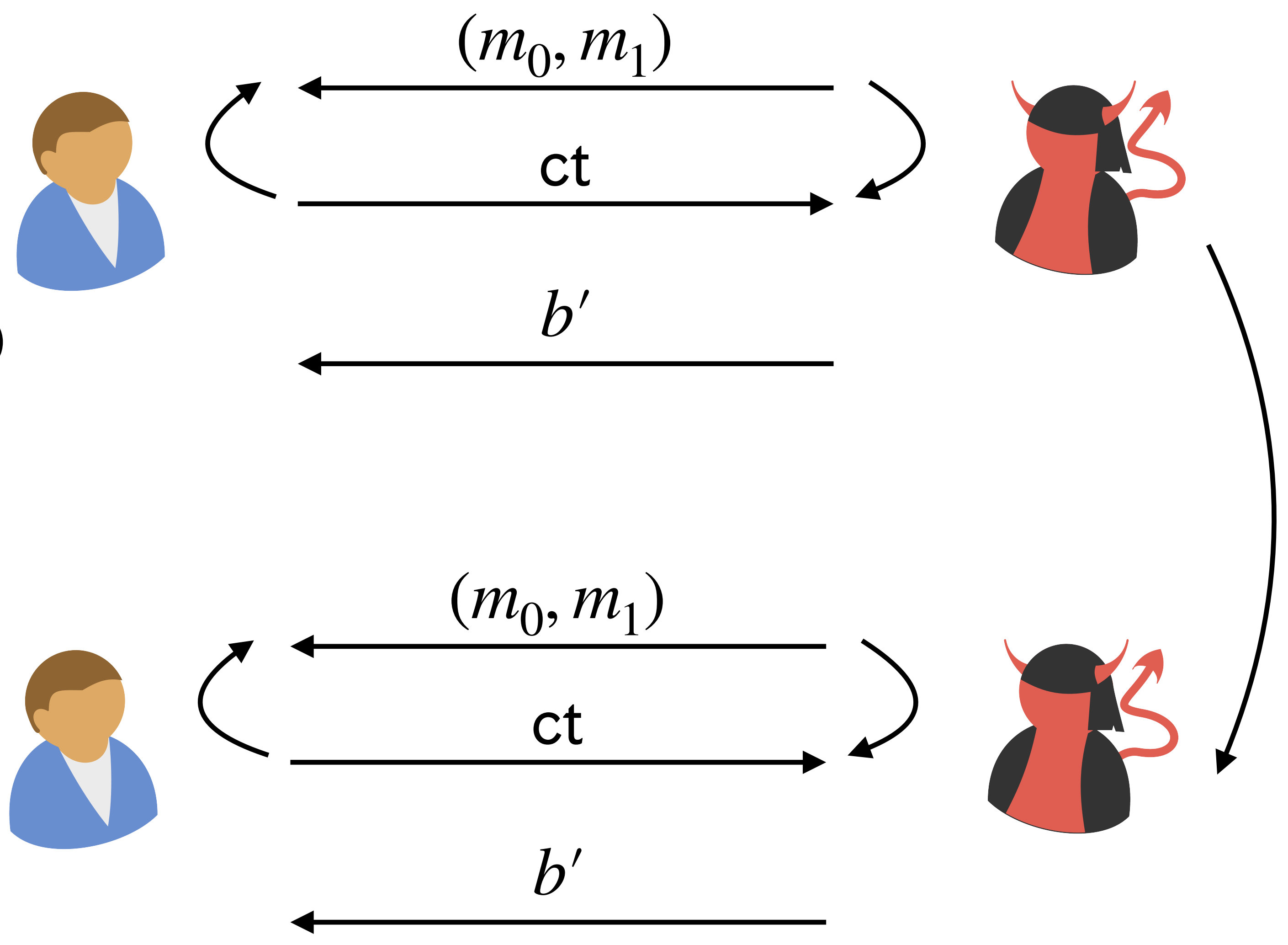
$x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m_0, x)$

H_1

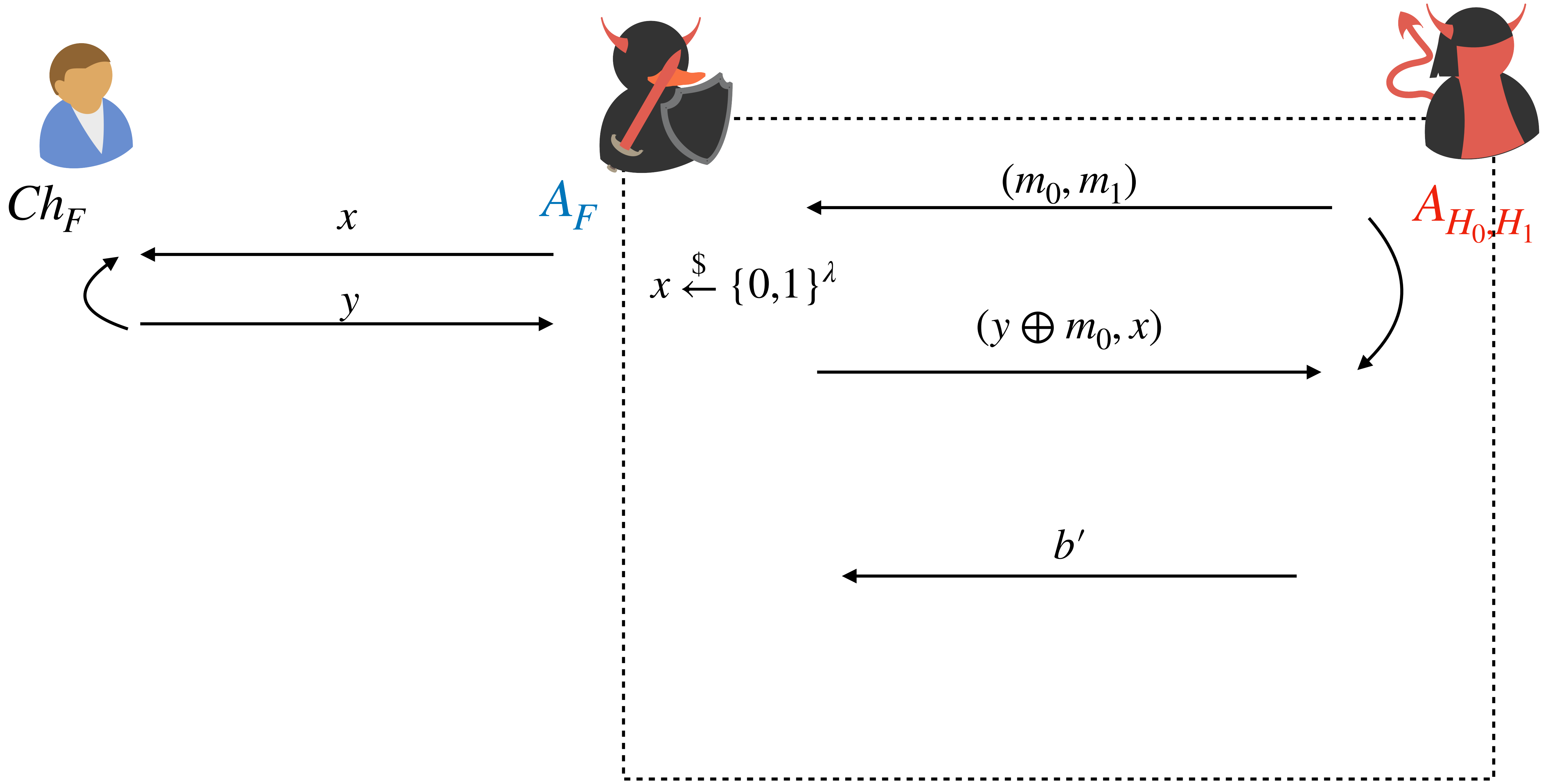
$T = \{\}$
$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$
  $r \xleftarrow{\$} \{0,1\}^\lambda$
  $T[x] = r$
$y = T[x]$

ct $:= (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F



Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(k, m) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$

(m_0, m_1)

ct

b'

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$

(m_0, m_1)

ct

b'

Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

Ch_F

A_F

A_{H_0,H_1}

x

y

(m_0, m_1)

$x \xleftarrow{\$} \{0,1\}^\lambda$

$(y \oplus m_0, x)$

b'

b'

Proof of Security

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m_0, x)$

(m_0, m_1)

ct

b'

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$

(m_0, m_1)

ct

b'

Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F

Ch_F

A_F

A_{H_0,H_1}

x

y

b'

(m_0, m_1)

$x \xleftarrow{\$} \{0,1\}^\lambda$

$(y \oplus m_0, x)$

b'

<u>Input Mapping</u>

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

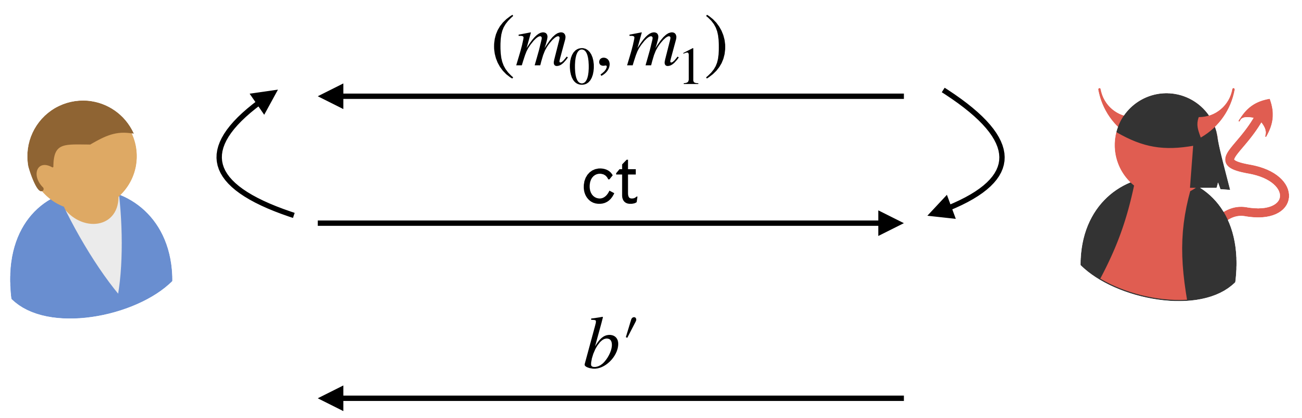
$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$



H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

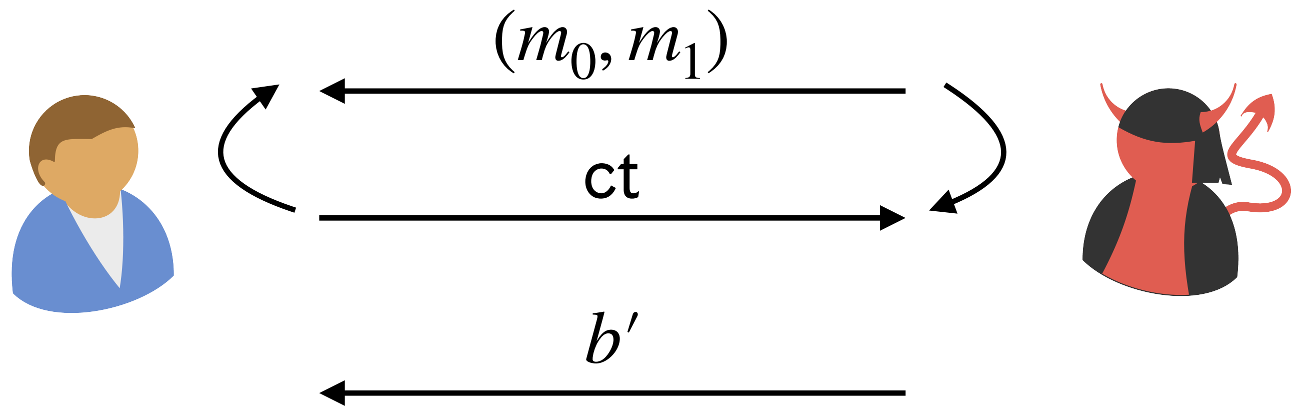
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

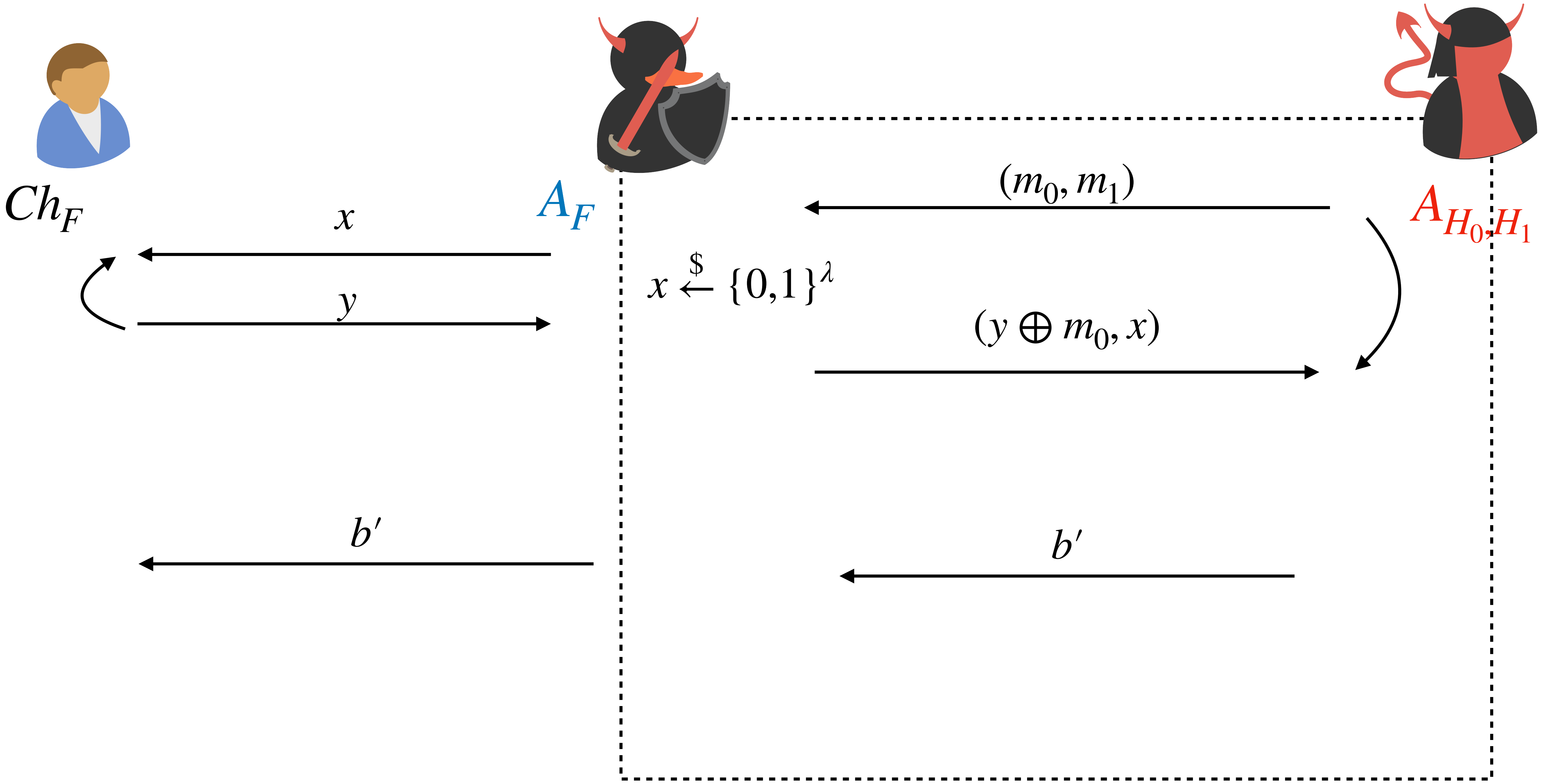
$\quad T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F



<u>Input Mapping</u>

$b = 0$: $(y \oplus m_0, x) = (F_k(x) \oplus m_0, x)$

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(k, m) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$; $\mathsf{ct} := (F_k(x) \oplus m, x)$

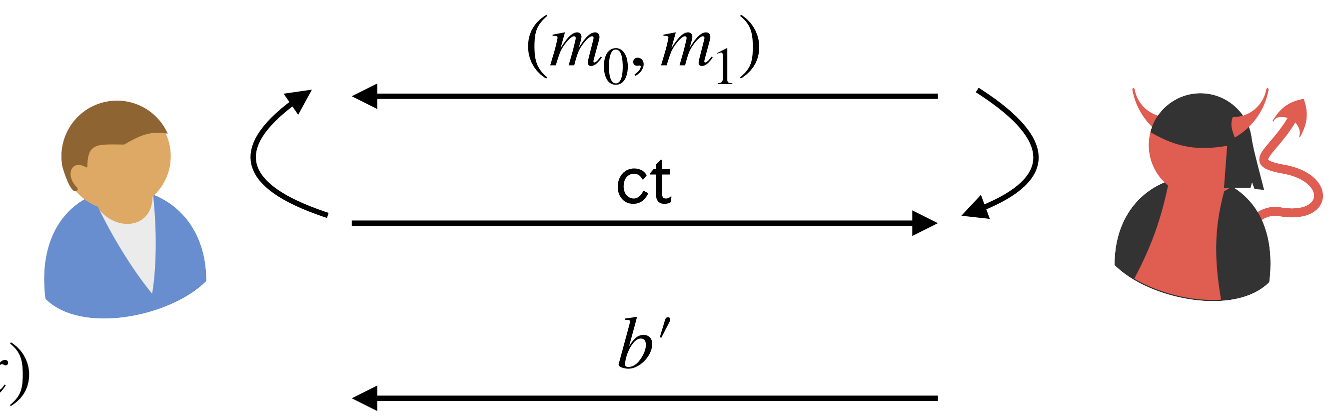
$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$



H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

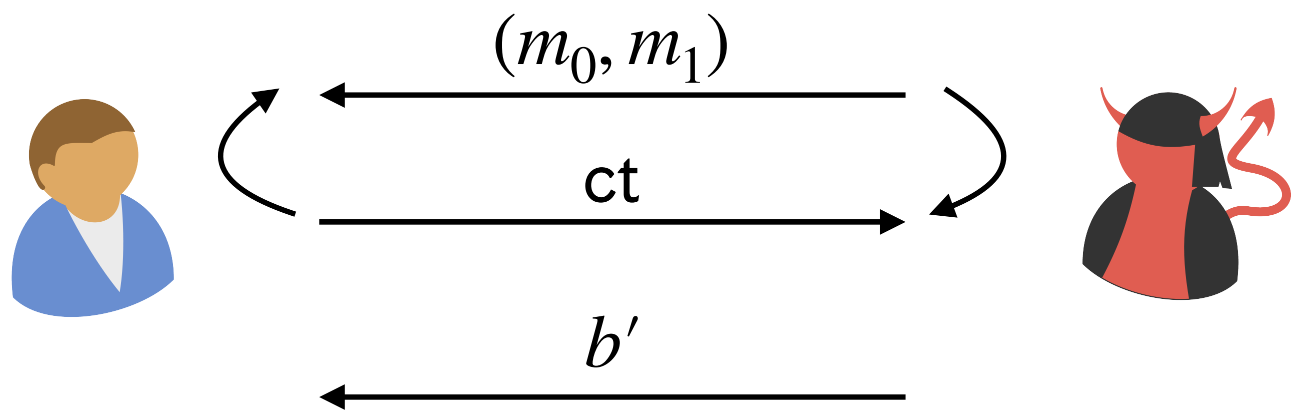
if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

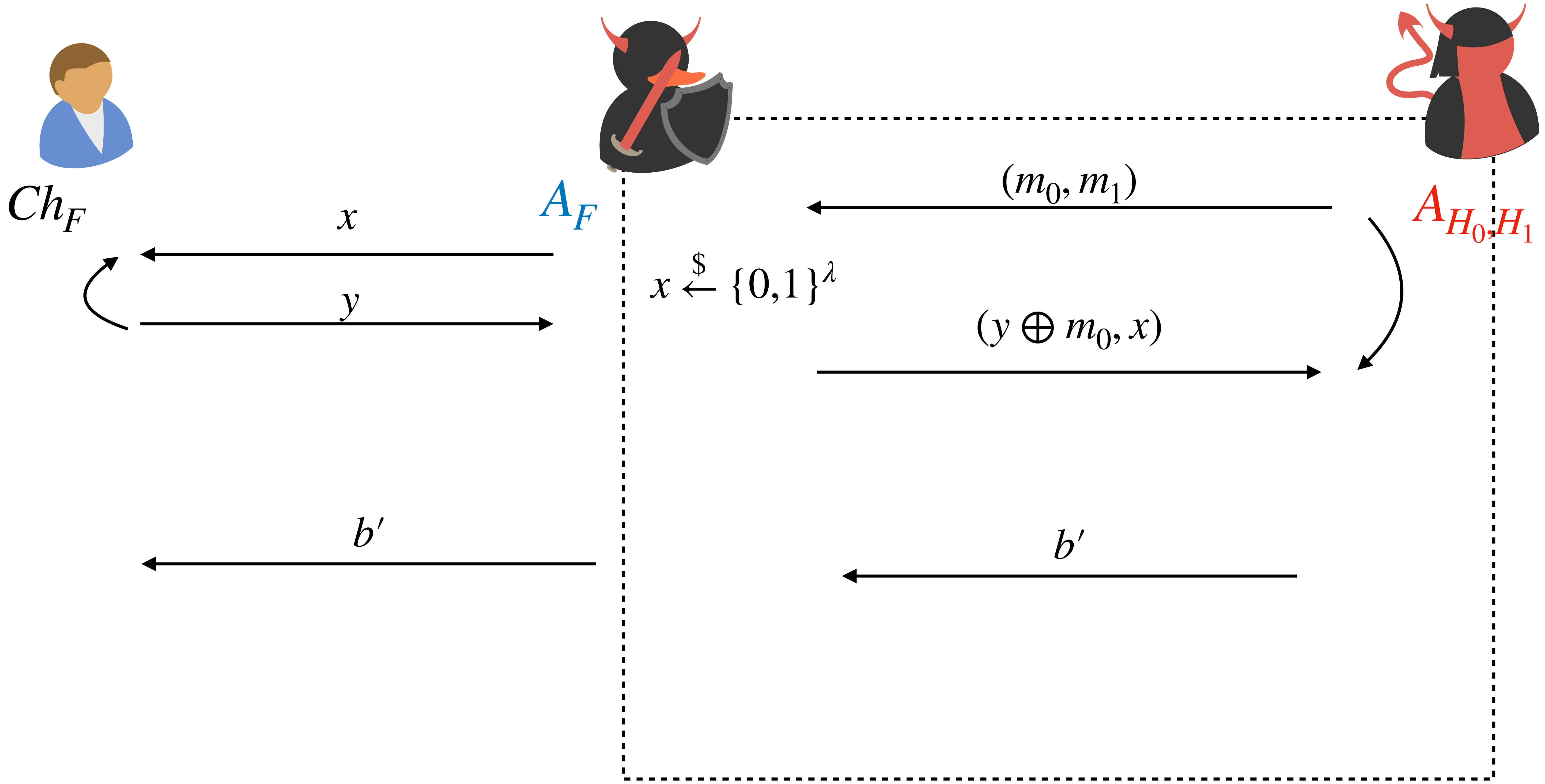
$T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_0, x)$



Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F



<u>Input Mapping</u>

$b = 0$: $(y \oplus m_0, x) = (F_k(x) \oplus m_0, x)$

$b = 1$: $(y \oplus m_0, x) = (T[x] \oplus m_0, x)$

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(k, m) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$, $\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$

(m_0, m_1)

ct

b'

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

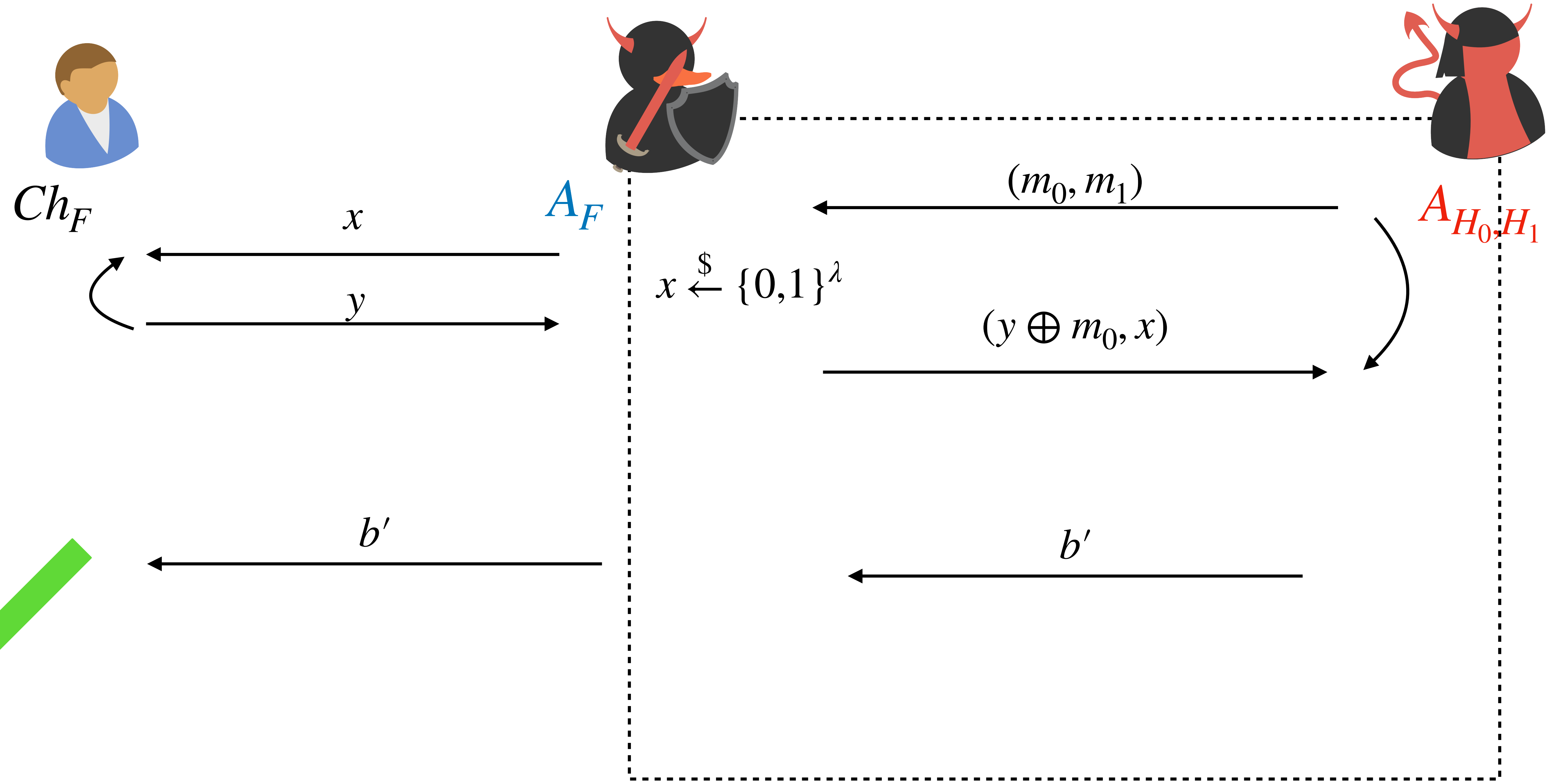
$\mathsf{ct} := (y \oplus m_0, x)$

(m_0, m_1)

ct

b'

Prove $H_0 \overset{c}{\approx} H_1$ via a reduction to the security of F



Input Mapping

$b = 0$: $(y \oplus m_0, x) = (F_k(x) \oplus m_0, x)$ ✓

$b = 1$: $(y \oplus m_0, x) = (T[x] \oplus m_0, x)$

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

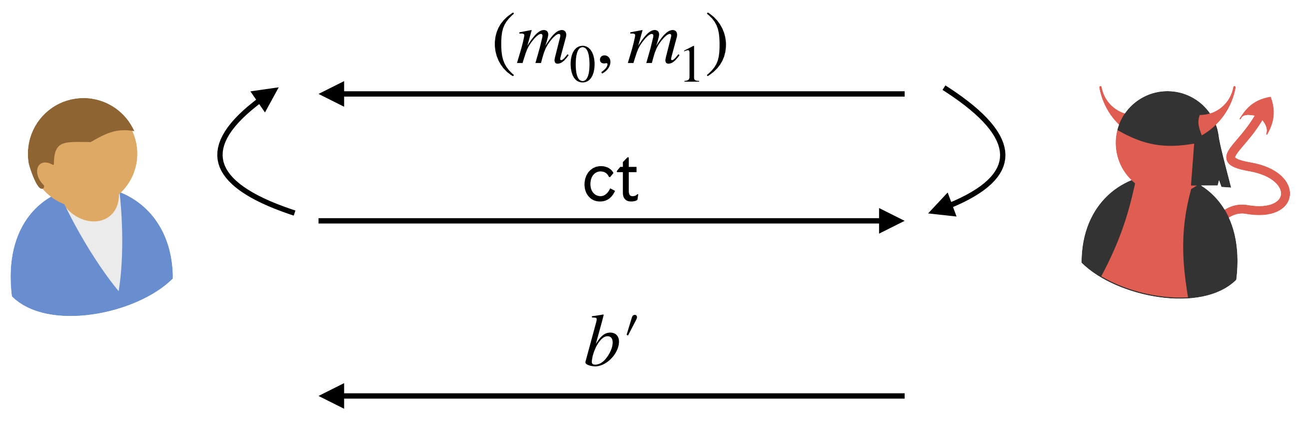
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

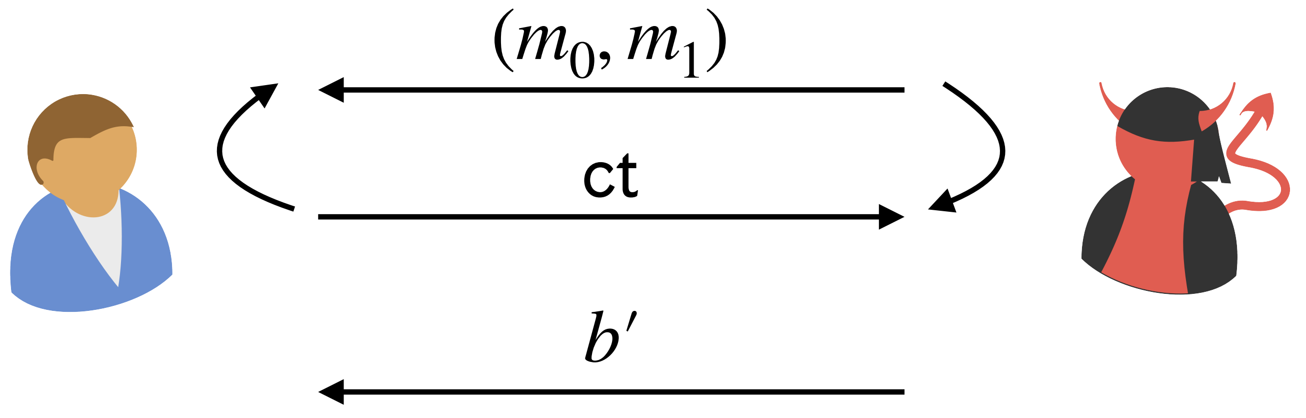
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



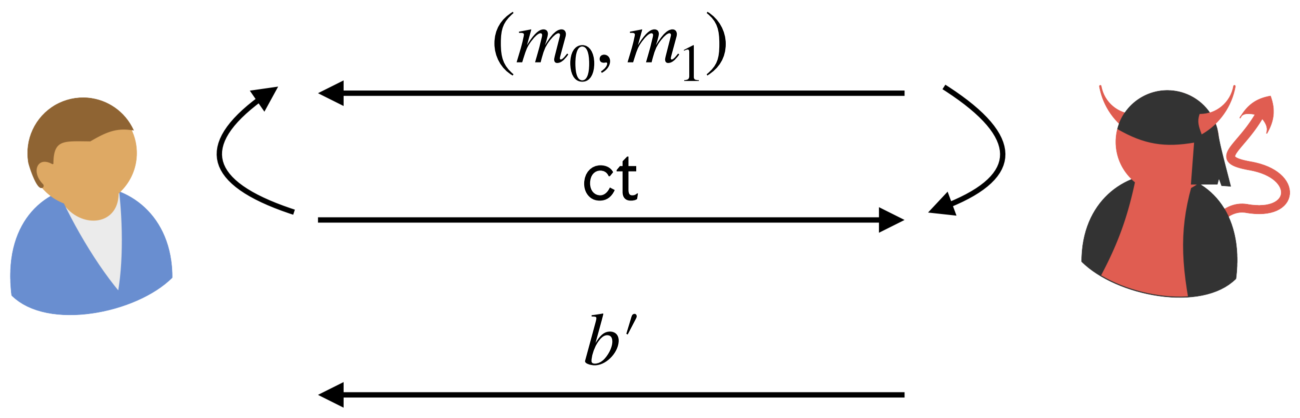
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, x)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$, ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

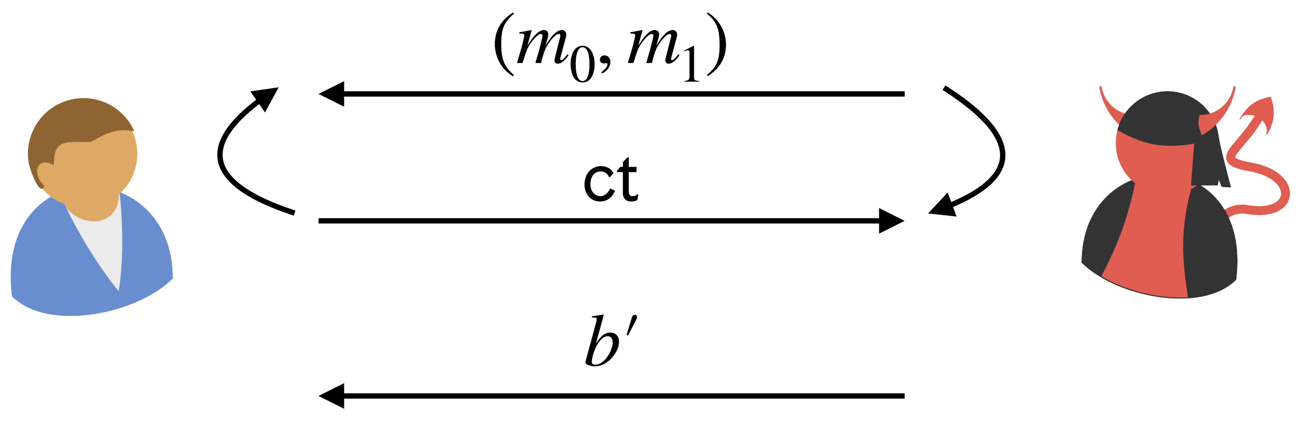
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



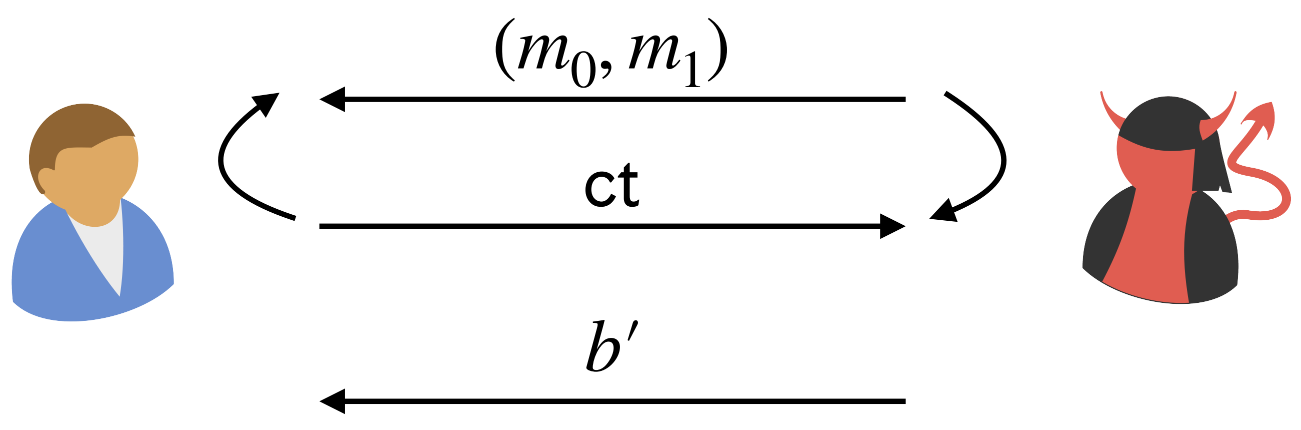
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, x)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

When does H_2 behave differently from H_1?

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

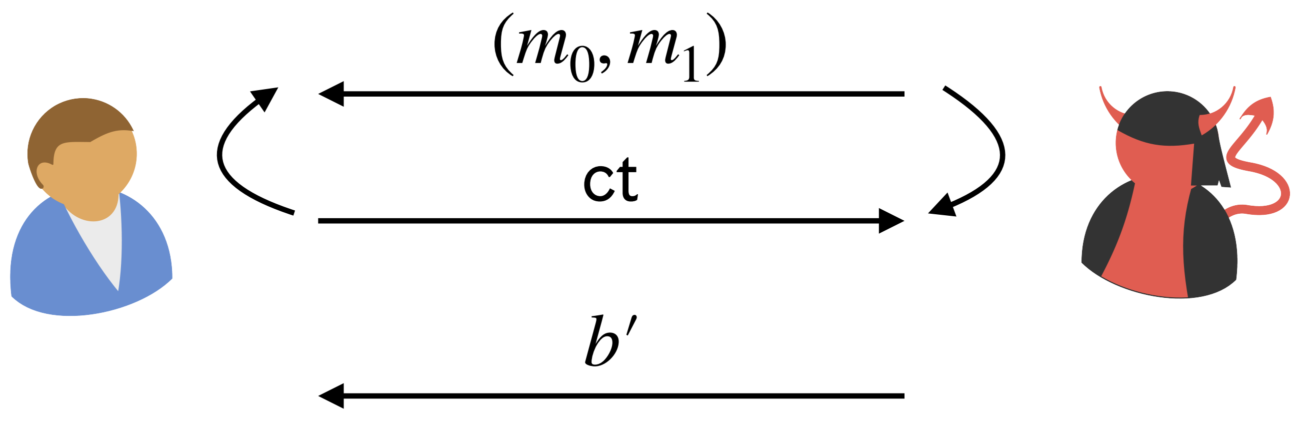
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



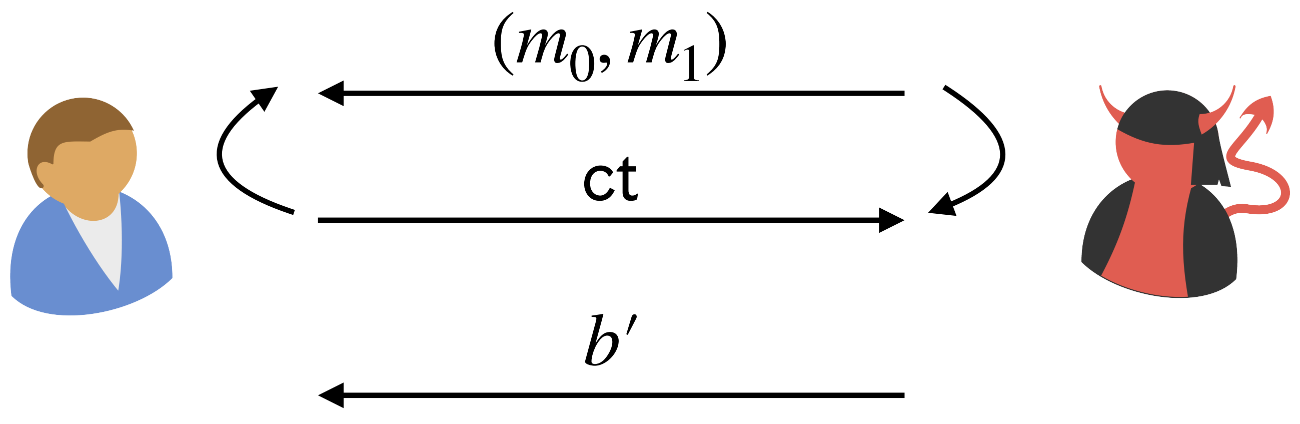
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, x)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

When does H_2 behave differently from H_1?

When the challenger samples the same x value twice!

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

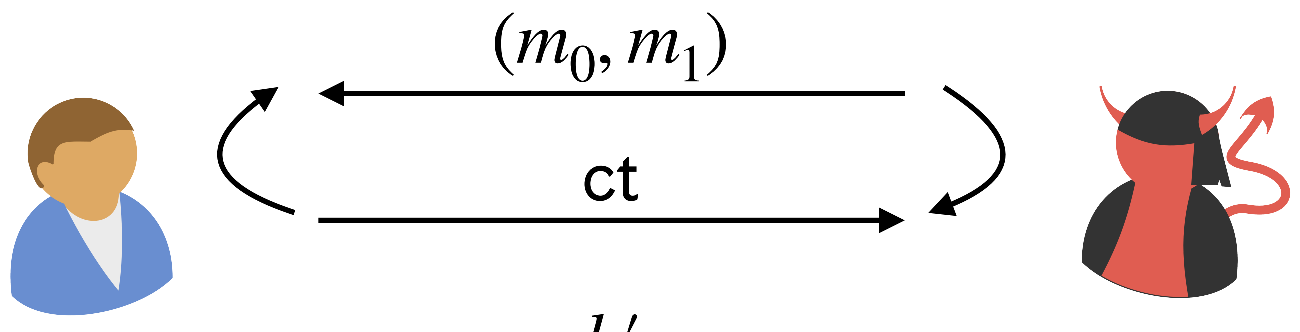
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



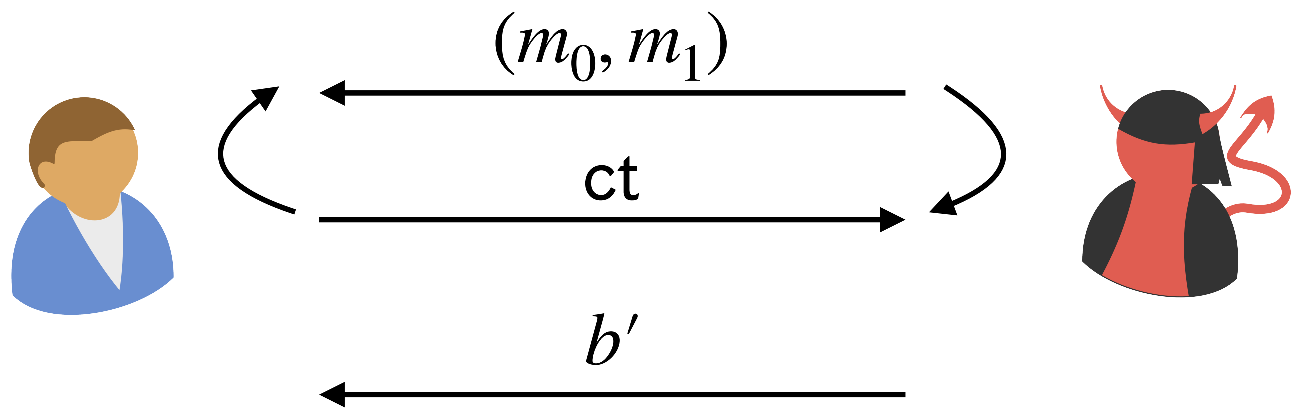
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, x)$



<u>Birthday Bound</u>: when sampling q elements from a space of size 2^λ, the probability of sampling some element twice is $P(q) \approx 1 - e^{\frac{-q^2}{2^{\lambda+1}}}$

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

When does H_2 behave differently from H_1?

When the challenger samples the same x value twice!

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

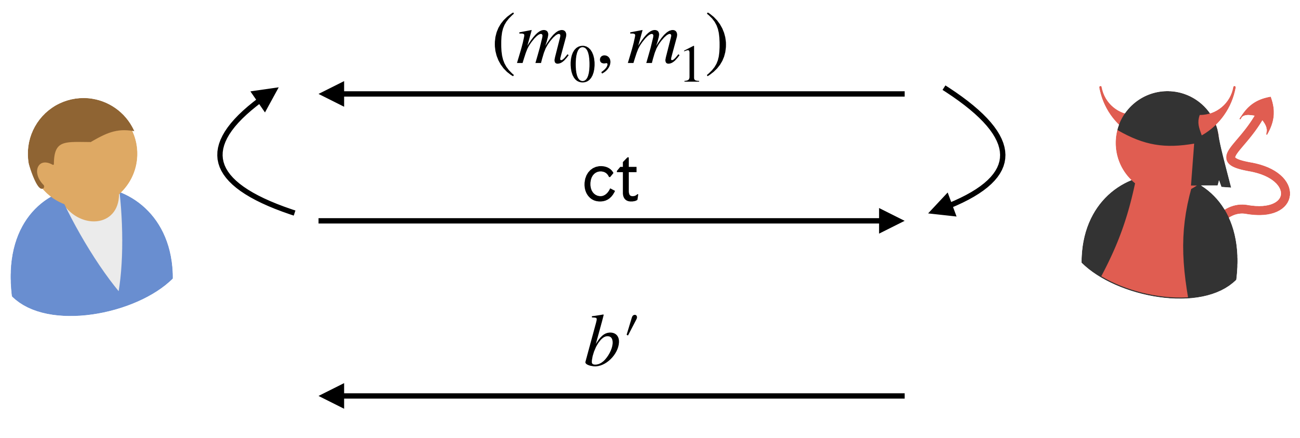
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



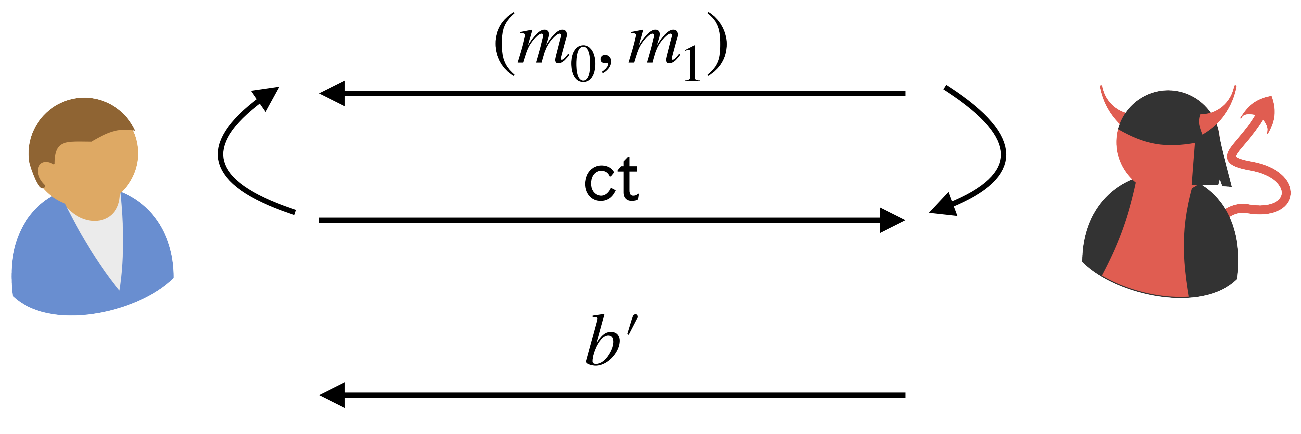
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, x)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

When does H_2 behave differently from H_1?

When the challenger samples the same x value twice!

Birthday Bound: when sampling q elements from a space of size 2^λ, the probability of sampling some element twice is $P(q) \approx 1 - e^{\frac{-q^2}{2^{\lambda+1}}}$

q is polynomial in λ, so
$P(q) \approx 1 - e^{-\mathsf{negl}(\lambda)} \approx 1 - e^0 = 1 - 1 = 0$

Proof of Security

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

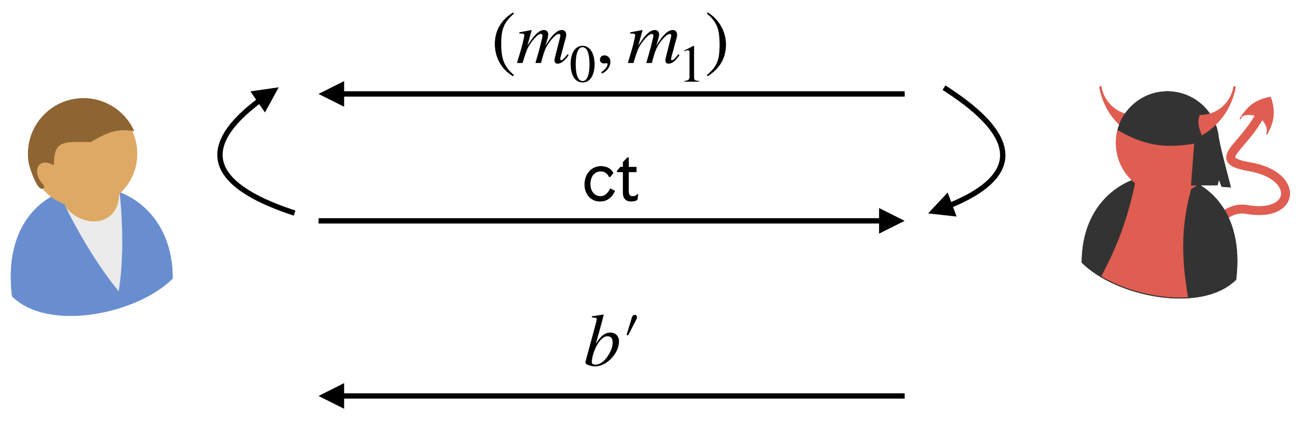
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



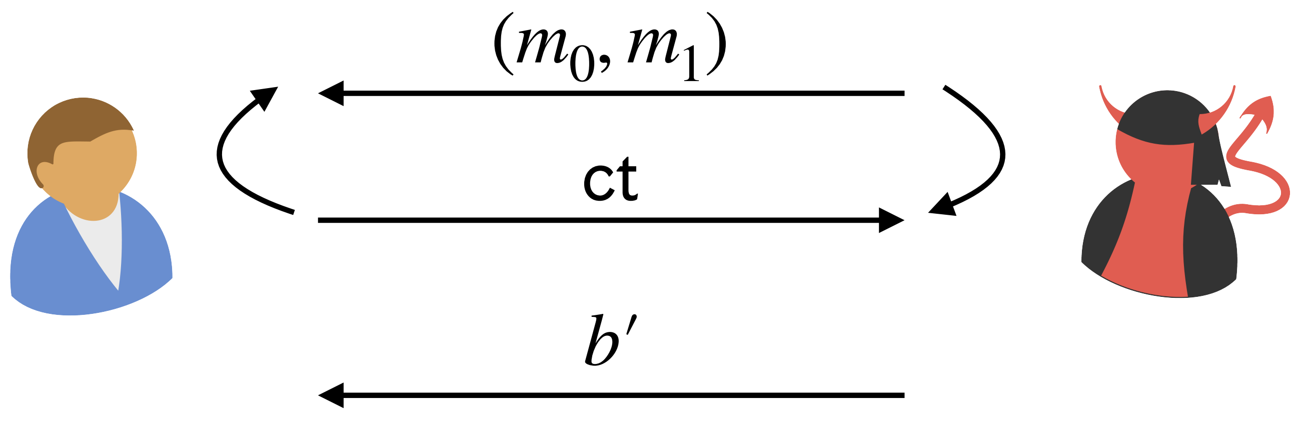
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, r)$



<u>Birthday Bound</u>: when sampling q elements from a space of size 2^λ, the probability of sampling some element twice is $P(q) \approx 1 - e^{\frac{-q^2}{2^{\lambda+1}}}$

q is polynomial in λ, so $P(q) \approx 1 - e^{-\mathsf{negl}(\lambda)} \approx 1 - e^0 = 1 - 1 = 0$

Proof of Security

<u>Difference Lemma</u>: for events A, B, F, if $\Pr[A \mid F] = \Pr[B \mid F]$, then $\left|\Pr[A] - \Pr[B]\right| \leq \Pr[F]$

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

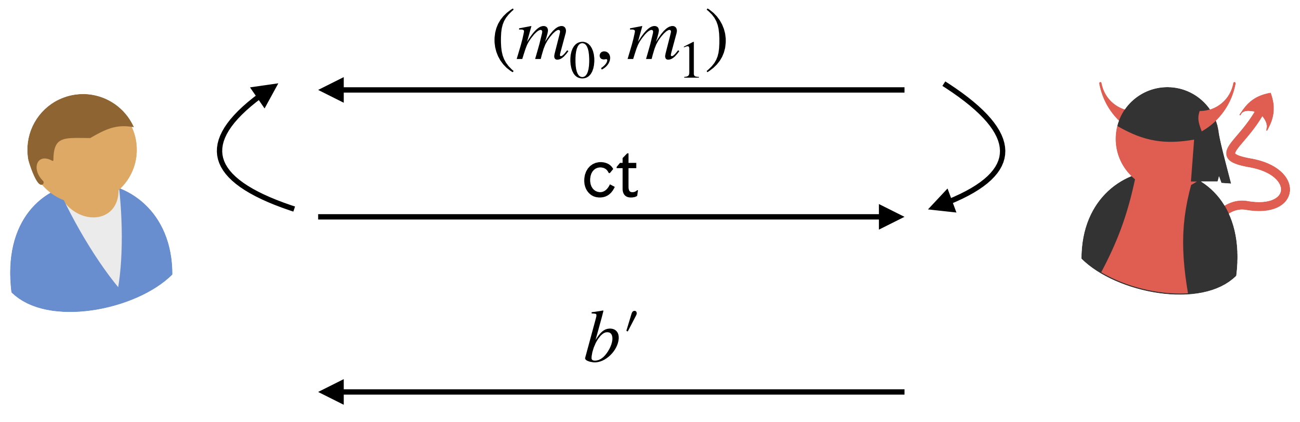
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



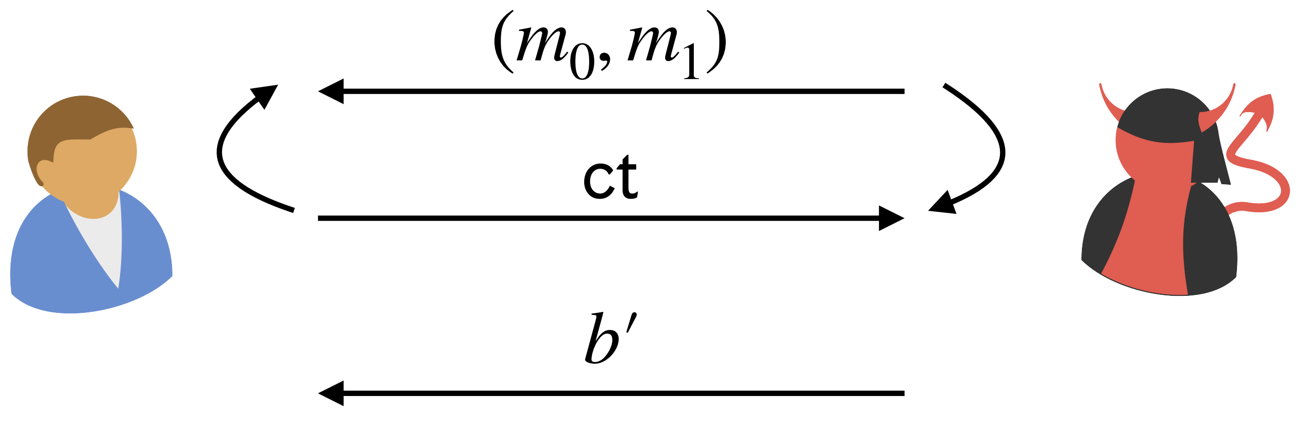
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, r)$



<u>Birthday Bound</u>: when sampling q elements from a space of size 2^λ, the probability of sampling some element twice is $P(q) \approx 1 - e^{\frac{-q^2}{2^{\lambda+1}}}$

q is polynomial in λ, so

$P(q) \approx 1 - e^{-\mathsf{negl}(\lambda)} \approx 1 - e^0 = 1 - 1 = 0$

Proof of Security

Difference Lemma: for events A, B, F, if $\Pr[A \mid F] = \Pr[B \mid F]$, then $\left|\Pr[A] - \Pr[B]\right| \le \Pr[F]$

H_1

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

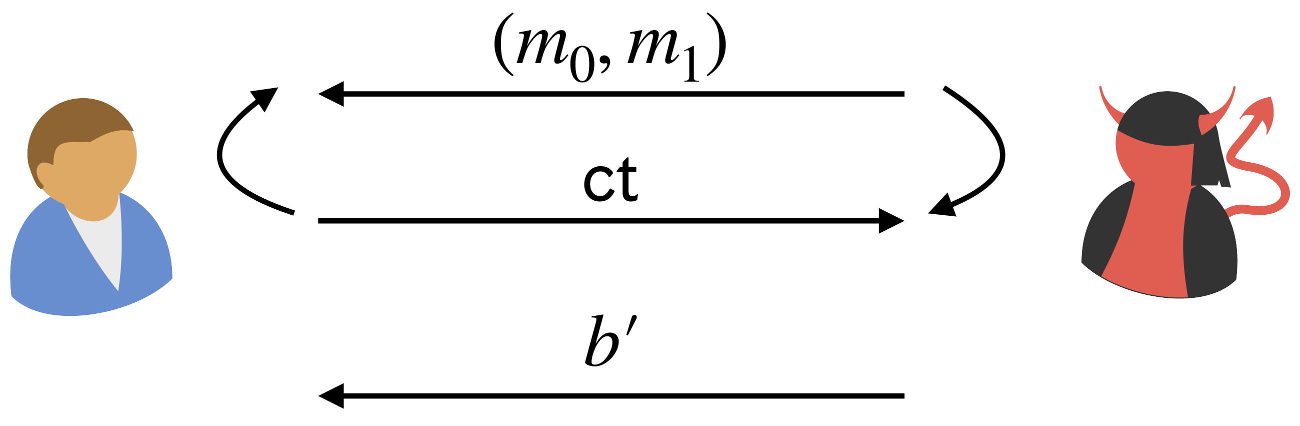
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_0, x)$



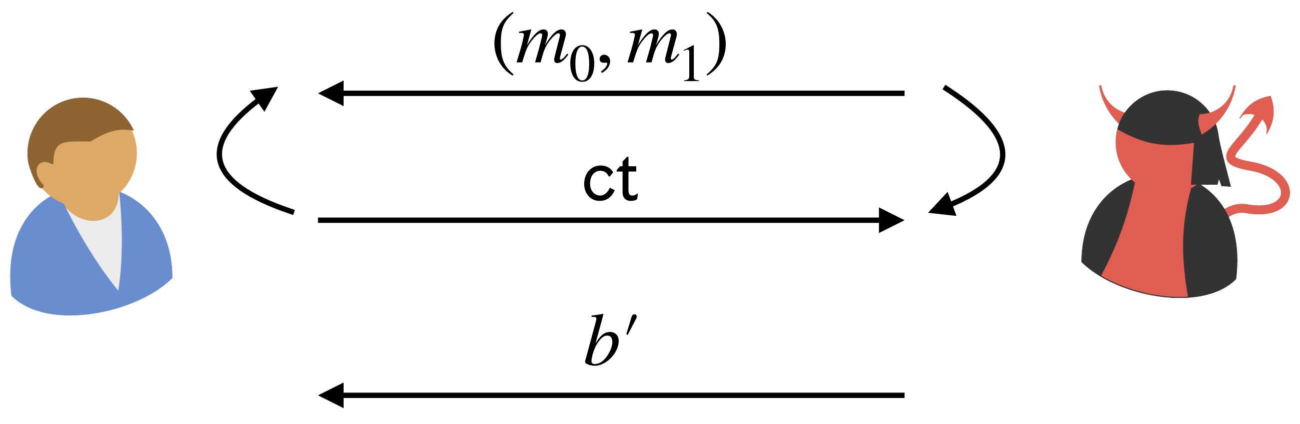
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, r)$



$$\left|\Pr\left[W_1\right] - \Pr\left[W_2\right]\right|$$

$$\le \Pr\left[C_{H_2} \text{ samples the same } x \text{ value twice}\right]$$

$$= P(q) = \mathsf{negl}(\lambda)$$

Birthday Bound: when sampling q elements from a space of size 2^λ, the probability of sampling some element twice is $P(q) \approx 1 - e^{\frac{-q^2}{2^{2\lambda+1}}}$

q is polynomial in λ, so

$P(q) \approx 1 - e^{-\mathsf{negl}(\lambda)} \approx 1 - e^0 = 1 - 1 = 0$

Proof of Security

H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_0, r)$

(m_0, m_1)

ct

b'

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k},\mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k,(c,x)) : m' := F_k(x) \oplus c$$

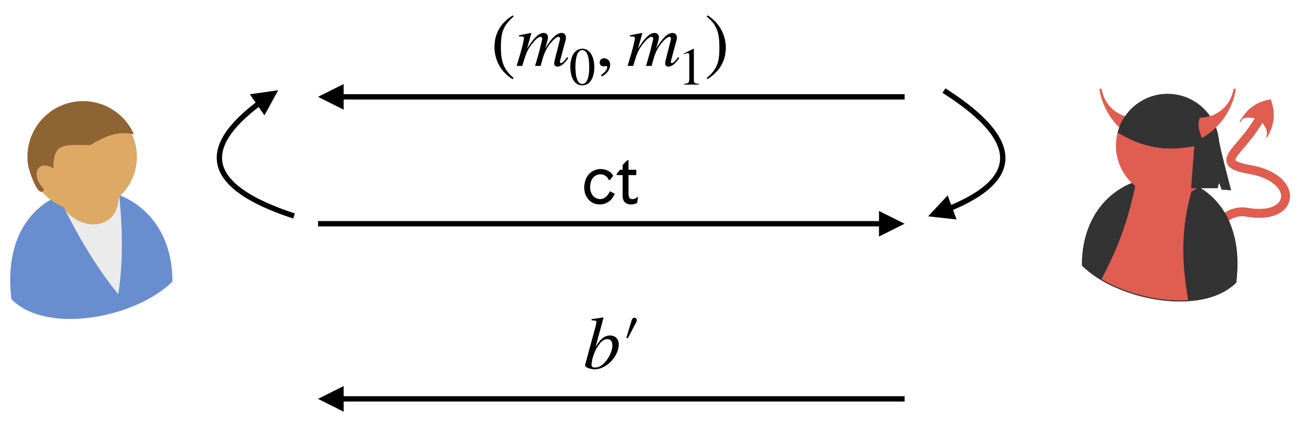
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (y \oplus m_0, r)$



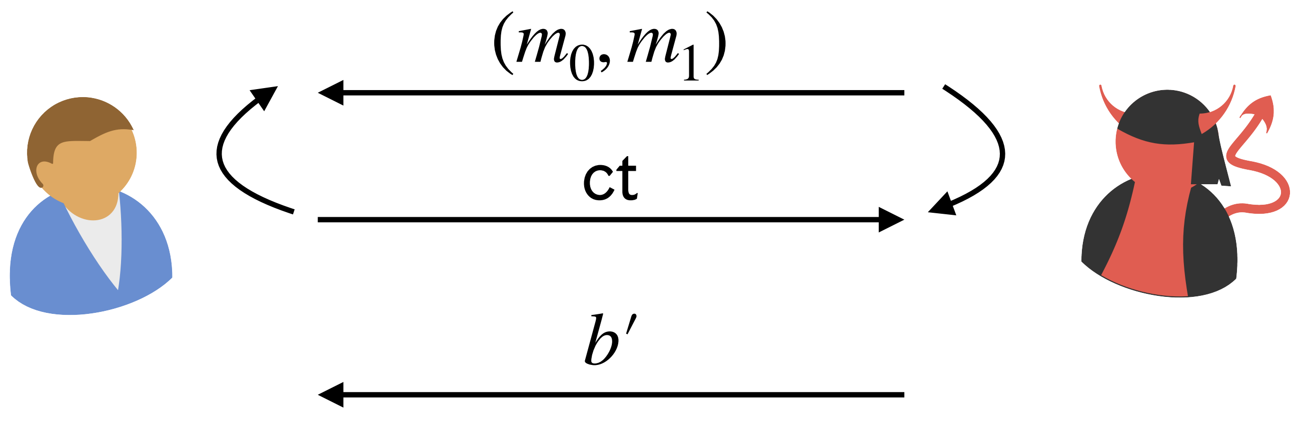
H_3

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (y \oplus m_1, r)$



Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

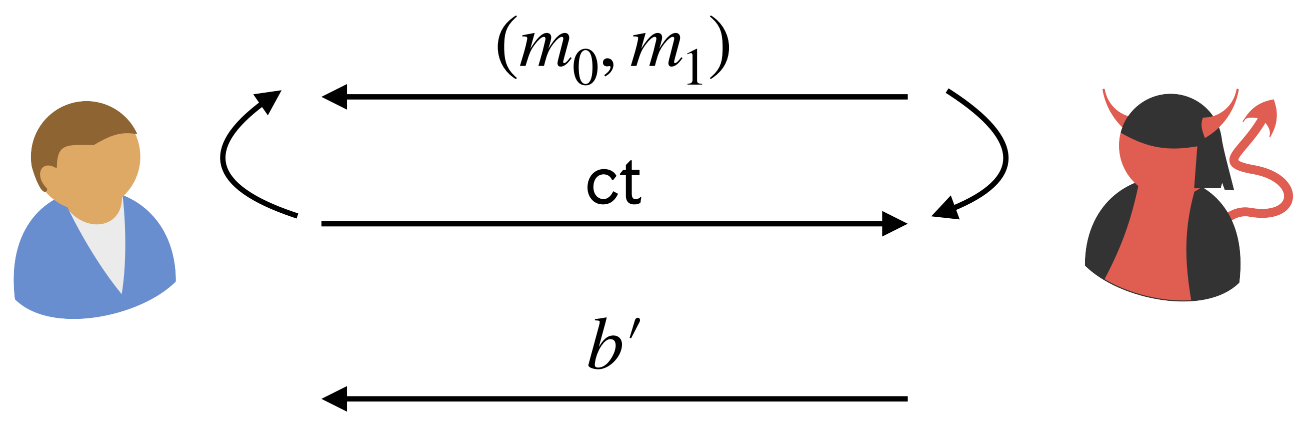
H_2

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (y \oplus m_0, r)$



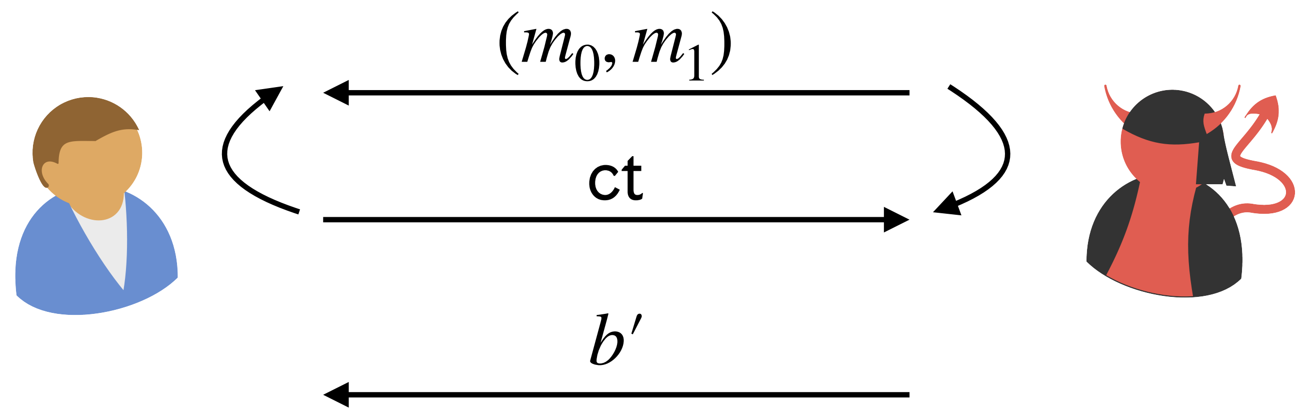
H_3

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (y \oplus m_1, r)$



Messages are all one-time-padded, can just swap them!

Proof of Security

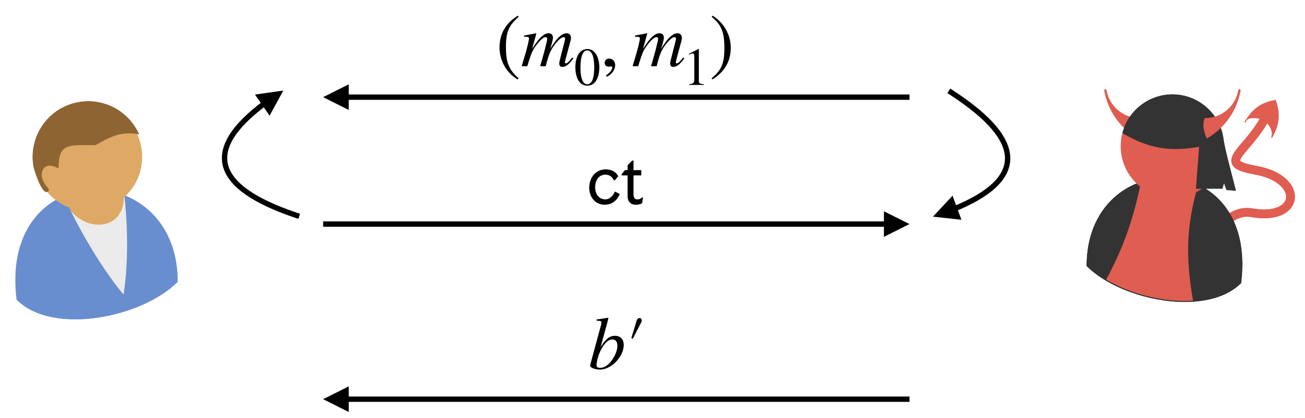
H_3

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_1, r)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

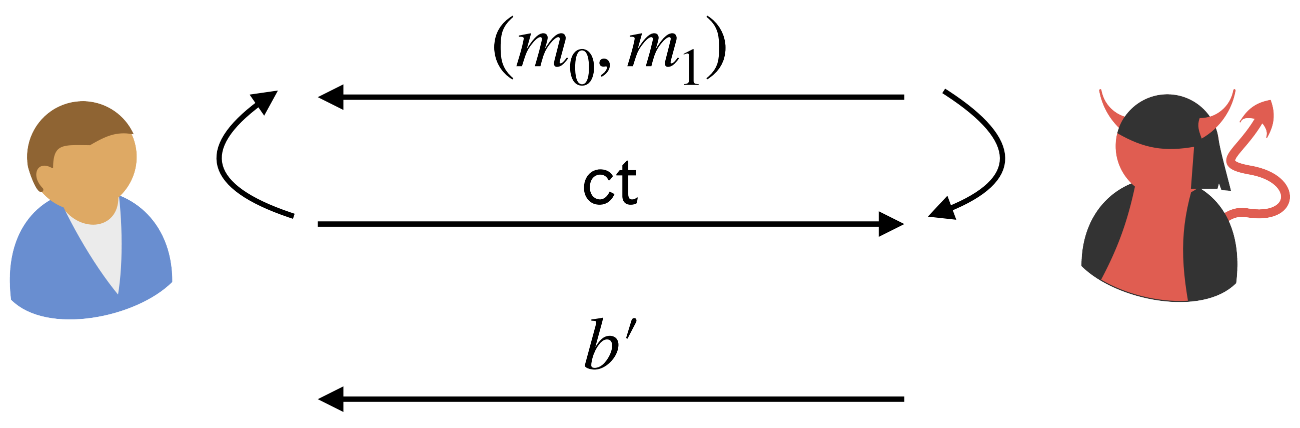
H_3

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (y \oplus m_1, r)$



H_4

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

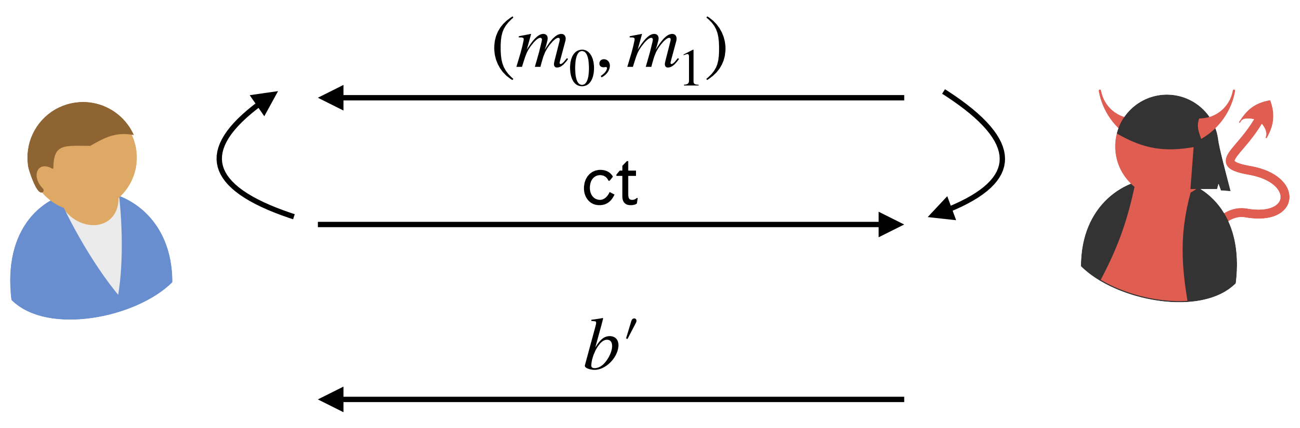
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_1, r)$



Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc(k, m)} : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

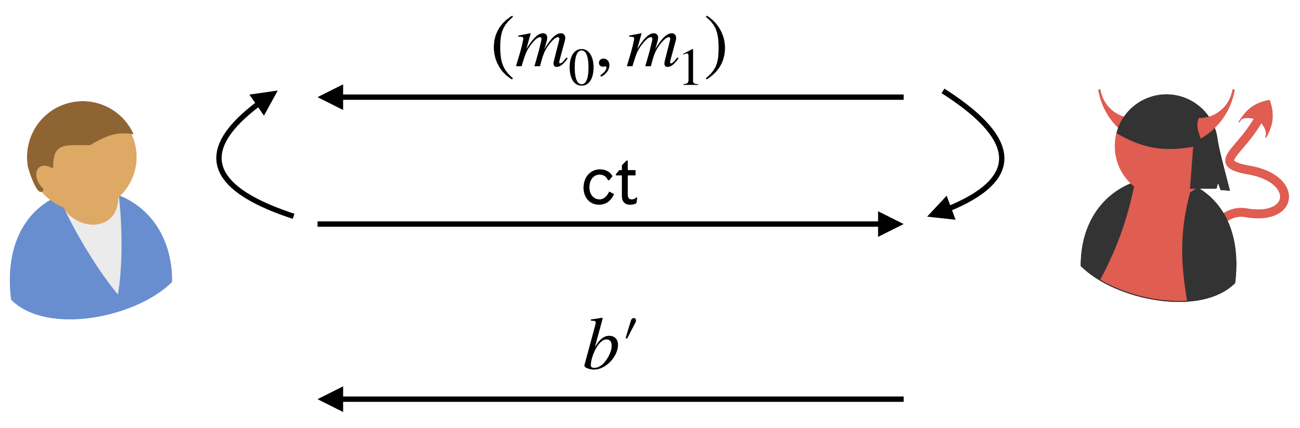
H_3

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$y \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (y \oplus m_1, r)$



H_4

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

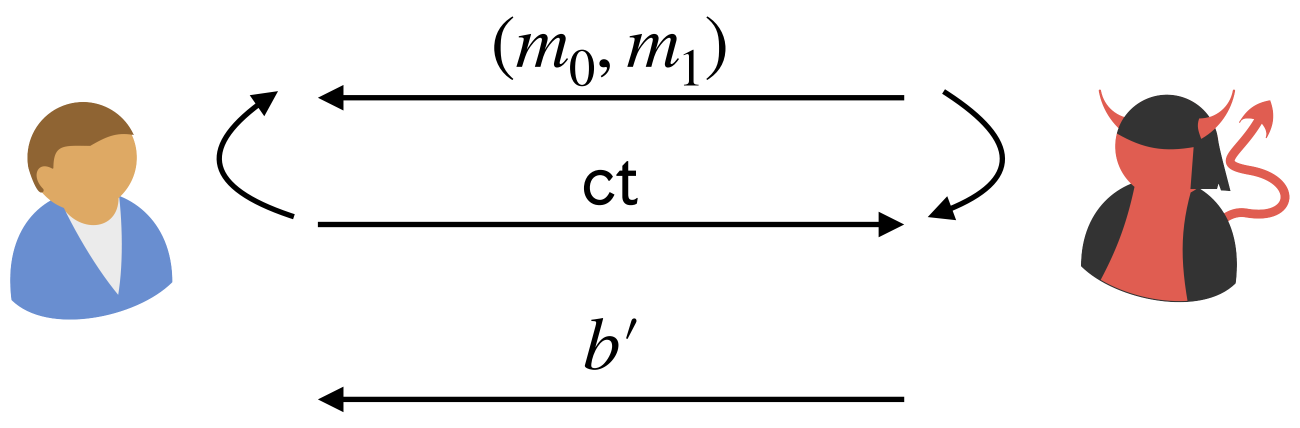
if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

ct $:= (y \oplus m_1, r)$



Exactly the same transition as H_1 to H_2!

Proof of Security

KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

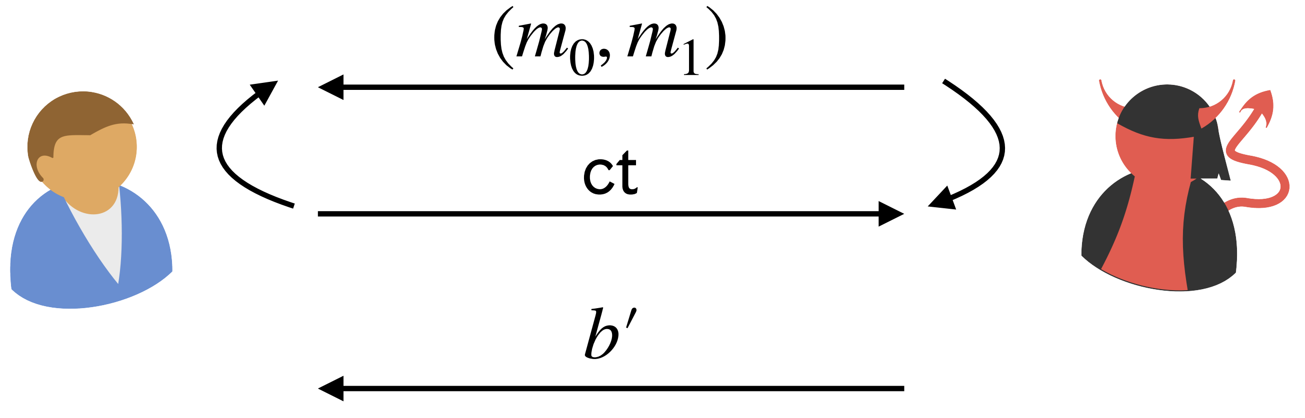
Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

H_4

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$



ct $:= (y \oplus m_1, r)$

Proof of Security

H_4

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

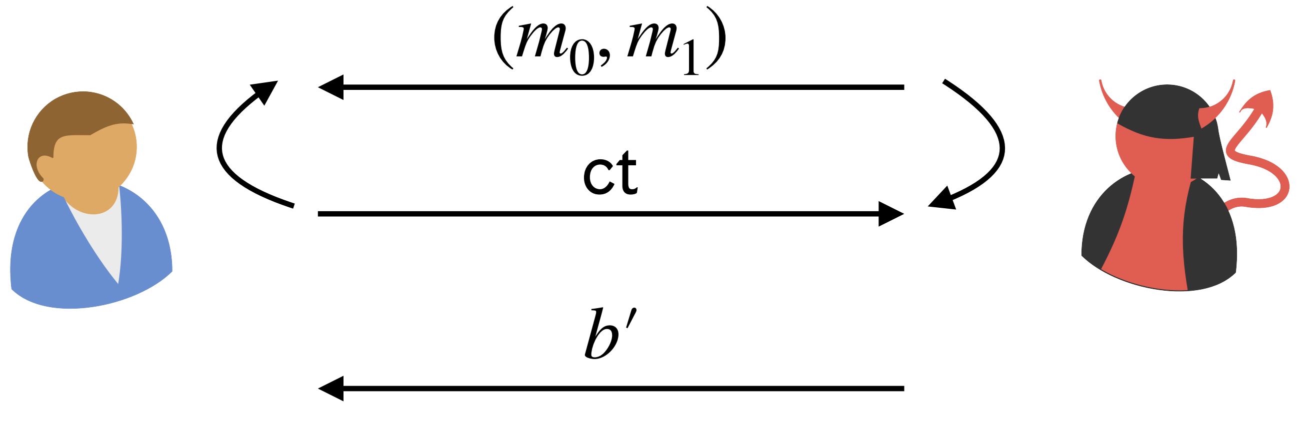
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_1, r)$



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \quad x \xleftarrow{\$} \{0,1\}^\lambda$

$\quad \mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

H_4

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

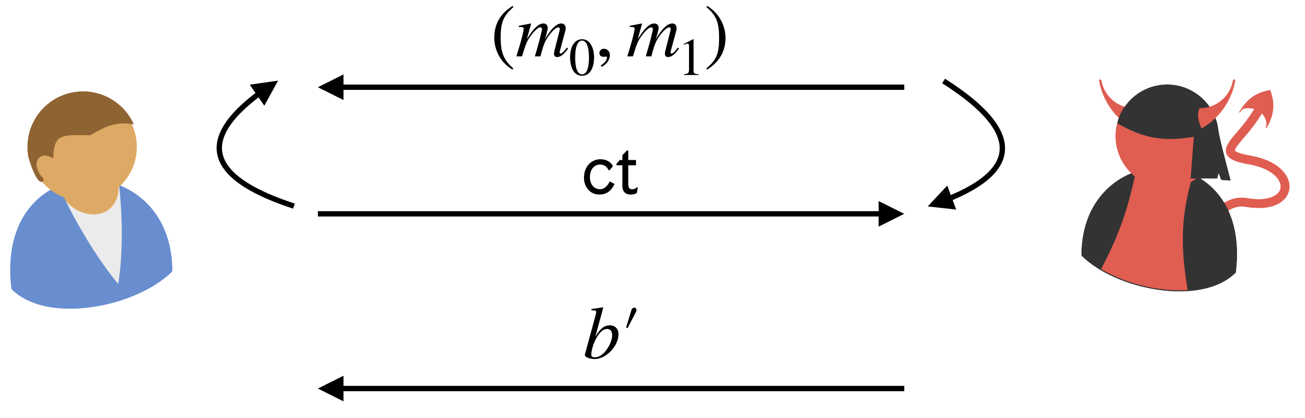
if $x \notin T$

$\quad r \xleftarrow{\$} \{0,1\}^\lambda$

$\quad T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_1, r)$

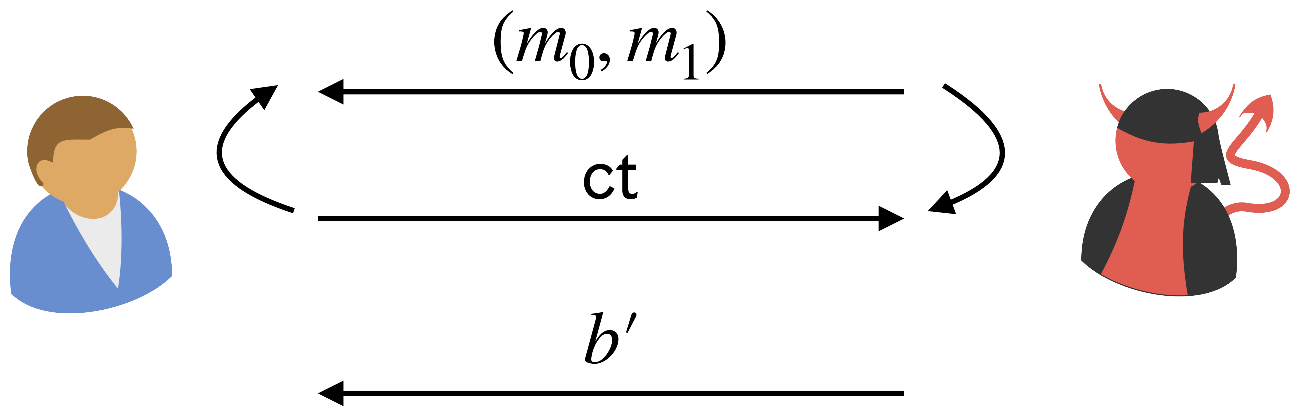


H_5

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_1, x)$



Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

H_4

$T = \{\}$

$x \xleftarrow{\$} \{0,1\}^\lambda$

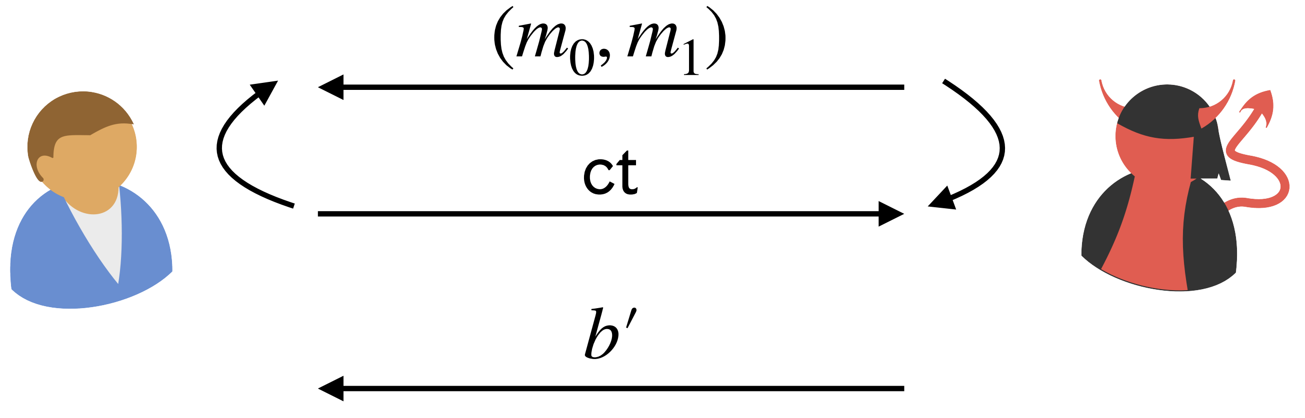
if $x \notin T$

$r \xleftarrow{\$} \{0,1\}^\lambda$

$T[x] = r$

$y = T[x]$

$\mathsf{ct} := (y \oplus m_1, r)$

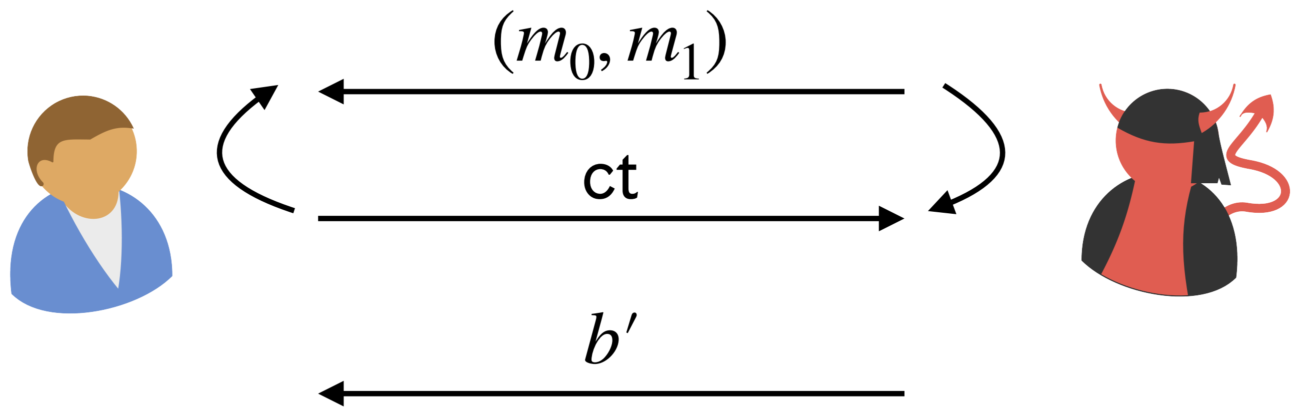


H_5

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_1, x)$



Exactly the same transition as H_0 to H_1!

Proof of Security

$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

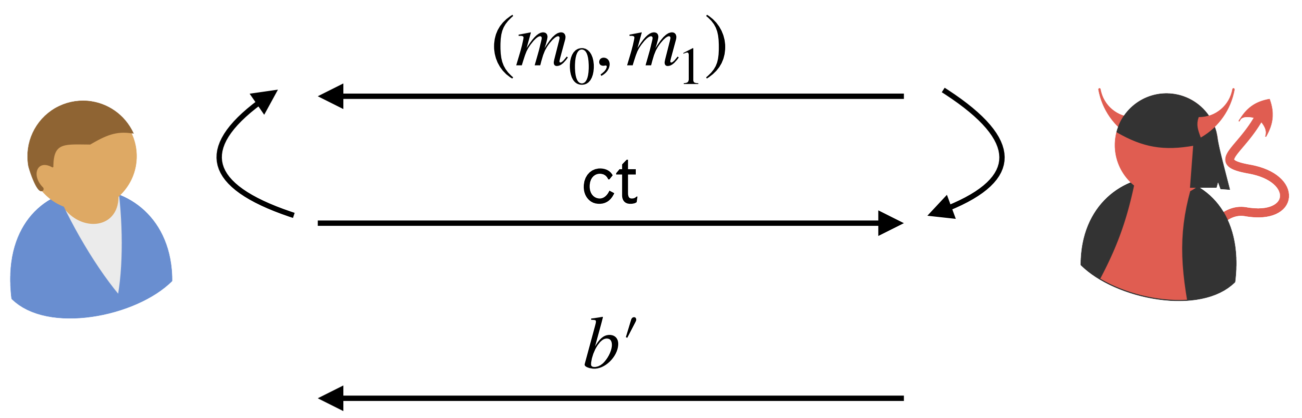
Proof of Security

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

ct $:= (F_k(x) \oplus m_0, x)$



KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Enc(k, m) : $x \xleftarrow{\$} \{0,1\}^\lambda$
ct $:= (F_k(x) \oplus m, x)$

Dec$(k, (c, x))$: $m' := F_k(x) \oplus c$

Proof of Security

$$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$$

$$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) : \begin{array}{l} x \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := (F_k(x) \oplus m, x) \end{array}$$

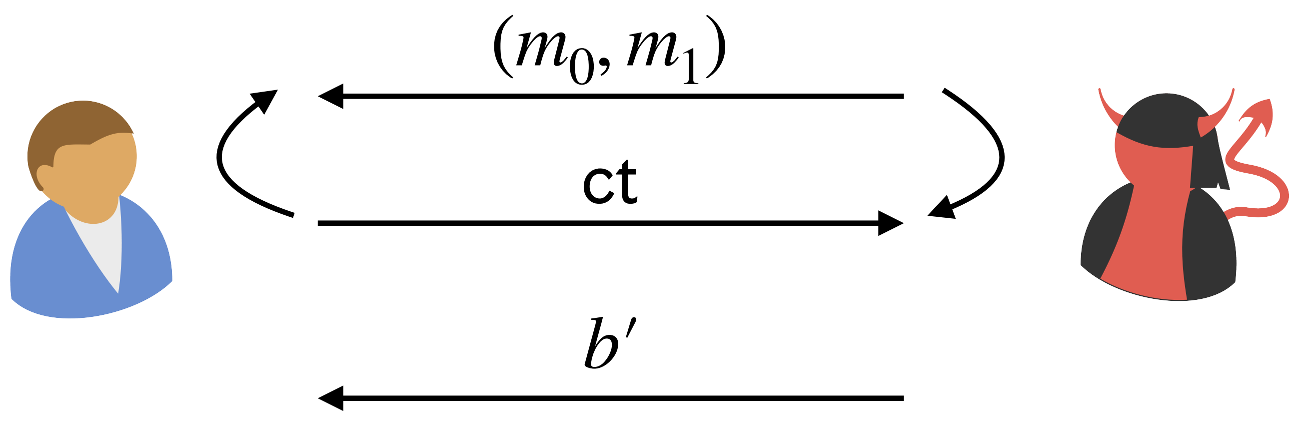
$$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_0, x)$

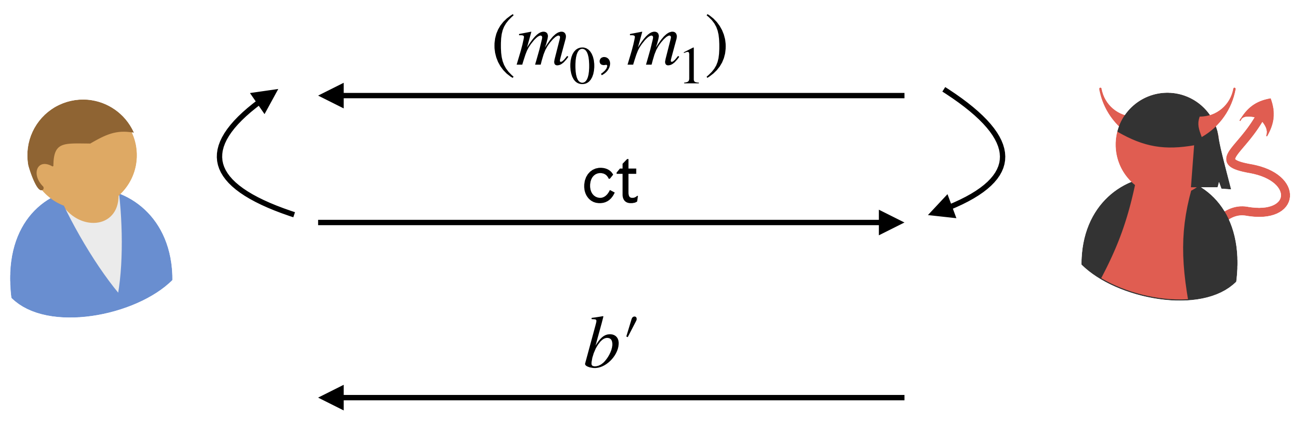


H_5

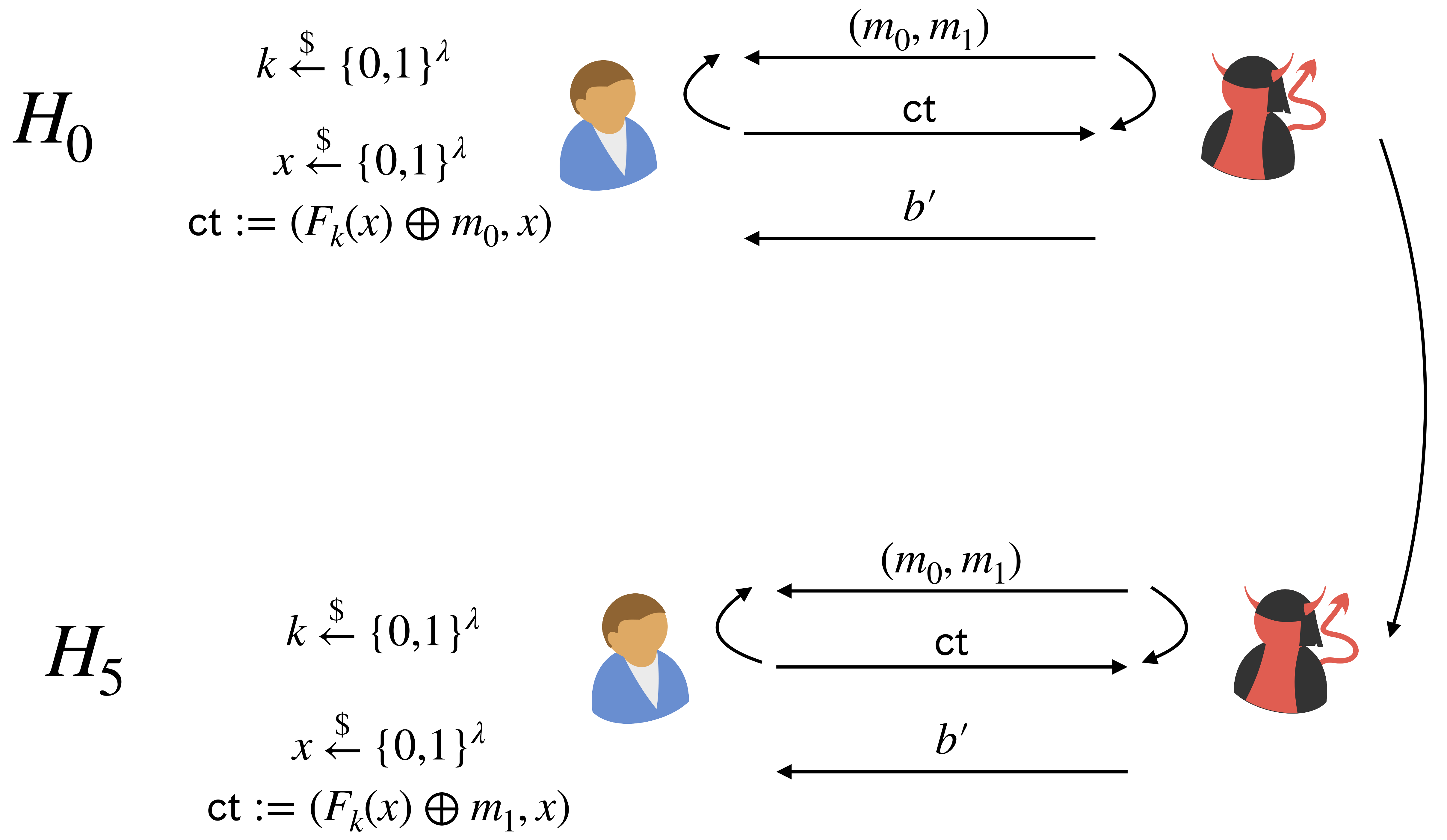
$k \xleftarrow{\$} \{0,1\}^\lambda$

$x \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{ct} := (F_k(x) \oplus m_1, x)$



Proof of Security



$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Enc}(\mathsf{k}, \mathsf{m}) :$ $x \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{ct} := (F_k(x) \oplus m, x)$

$\mathsf{Dec}(k, (c, x)) : m' := F_k(x) \oplus c$

By the hybrid lemma $H_0 \overset{c}{\approx} H_5$, and so our encryption scheme satisfies IND-CPA security

Public Key Encryption (PKE)

Public Key Encryption (PKE)

- What if Alice and Bob <u>don't</u> share a secret? Can they still communicate securely?

Public Key Encryption (PKE)

- What if Alice and Bob <u>don't</u> share a secret? Can they still communicate securely?
- Public key setting:

Public Key Encryption (PKE)

- What if Alice and Bob don't share a secret? Can they still communicate securely?
- Public key setting:
  - Both parties have a *public key* pk, and a *secret key* sk

Public Key Encryption (PKE)

- What if Alice and Bob <u>don't</u> share a secret? Can they still communicate securely?
- Public key setting:
    - Both parties have a *public key* pk, and a *secret key* sk
    - Knowing *only Bob's public key*, Alice can encrypt a message to Bob

Public Key Encryption (PKE)

- What if Alice and Bob <u>don't</u> share a secret? Can they still communicate securely?
- Public key setting:
  - Both parties have a *public key* pk, and a *secret key* sk
  - Knowing *only Bob's public key*, Alice can encrypt a message to Bob
  - Bob can decrypt the ciphertext using *sk*

Public Key Encryption (PKE)

- What if Alice and Bob <u>don't</u> share a secret? Can they still communicate securely?
- Public key setting:
  - Both parties have a *public key* pk, and a *secret key* sk
  - Knowing *only Bob's public key*, Alice can encrypt a message to Bob
  - Bob can decrypt the ciphertext using *sk*
- We may also refer to this as the *asymmetric* setting, and private key encryption as the *symmetric* setting

Public Key Encryption (PKE)

Public Key Encryption Scheme Syntax

A *public key encryption scheme* consists of three (possibly probabilistic) algorithms:

Public Key Encryption (PKE)

Public Key Encryption Scheme Syntax

A *public key encryption scheme* consists of three (possibly probabilistic) algorithms:

- KeyGen$(1^\lambda) \to (sk, pk)$ outputs a secret key $sk \in \mathscr{K}_s$ and a public key $pk \in \mathscr{K}_p$

Public Key Encryption (PKE)

Public Key Encryption Scheme Syntax

A *public key encryption scheme* consists of three (possibly probabilistic) algorithms:

- KeyGen$(1^\lambda) \rightarrow (sk, pk)$ outputs a secret key $sk \in \mathscr{K}_s$ and a public key $pk \in \mathscr{K}_p$
- Enc$(pk, m) \rightarrow$ ct takes a public key pk and a message $m \in \mathscr{M}$ and outputs a ciphertext ct $\in \mathscr{C}$

Public Key Encryption (PKE)

Public Key Encryption Scheme Syntax

A *public key encryption scheme* consists of three (possibly probabilistic) algorithms:

- KeyGen$(1^\lambda) \rightarrow (sk, pk)$ outputs a secret key $sk \in \mathcal{K}_s$ and a public key $pk \in \mathcal{K}_p$
- Enc$(pk, m) \rightarrow$ ct takes a public key pk and a message $m \in \mathcal{M}$ and outputs a ciphertext ct $\in \mathcal{C}$
- Dec$(sk,$ ct$) \rightarrow m$ takes a secret key sk and a ciphertext ct and outputs a message m

PKE Security

KeyGen $\rightarrow (pk, sk)$

Enc$(pk, m) \rightarrow$ ct

Dec$(sk, \text{ct}) \rightarrow m$

Wins if $b' = b$

PKE Security

KeyGen $\rightarrow (pk, sk)$

Enc$(pk, m) \rightarrow$ ct

Dec$(sk, \text{ct}) \rightarrow m$

Wins if $b' = b$

PKE Security

KeyGen $\rightarrow (pk, sk)$

$\mathsf{Enc}(pk, m) \rightarrow \mathsf{ct}$

$\mathsf{Dec}(sk, \mathsf{ct}) \rightarrow m$

$$b \xleftarrow{\$} \{0,1\}$$

$$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$$

Wins if $b' = b$

PKE Security

KeyGen $\rightarrow (pk, sk)$
Enc$(pk, m) \rightarrow$ ct
Dec$(sk,$ ct$) \rightarrow m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\xrightarrow{\quad pk \quad}$

Wins if $b' = b$

PKE Security

KeyGen $\rightarrow (pk, sk)$

Enc$(pk, m) \rightarrow$ ct

Dec$(sk,$ ct$) \rightarrow m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow$ KeyGen(1^λ)

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

Wins if $b' = b$

PKE Security

KeyGen $\to (pk, sk)$

$\mathsf{Enc}(pk, m) \to \mathsf{ct}$

$\mathsf{Dec}(sk, \mathsf{ct}) \to m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(pk, m_b)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

Wins if $b' = b$

PKE Security

$\mathsf{KeyGen} \to (pk, sk)$

$\mathsf{Enc}(pk, m) \to \mathsf{ct}$

$\mathsf{Dec}(sk, \mathsf{ct}) \to m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(pk, m_b)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \mathsf{ct}_b \quad}$

Wins if $b' = b$

PKE Security

KeyGen $\to (pk, sk)$

Enc$(pk, m) \to$ ct

Dec$(sk, \text{ct}) \to m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

$\text{ct}_b \leftarrow \text{Enc}(pk, m_b)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \text{ct}_b \quad}$

$\xleftarrow{\quad b' \quad}$

Wins if $b' = b$

PKE Security

KeyGen $\to (pk, sk)$

$\mathsf{Enc}(pk, m) \to \mathsf{ct}$

$\mathsf{Dec}(sk, \mathsf{ct}) \to m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(pk, m_b)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \mathsf{ct}_b \quad}$

$\xleftarrow{\quad b' \quad}$

Wins if $b' = b$

Adversary only makes a *single* query!

PKE Security

KeyGen $\to (pk, sk)$

Enc$(pk, m) \to$ ct

Dec$(sk, \text{ct}) \to m$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

$\text{ct}_b \leftarrow \text{Enc}(pk, m_b)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \text{ct}_b \quad}$

$\xleftarrow{\quad b' \quad}$

Wins if $b' = b$

Adversary only makes a *single* query!

In the public key setting, security for a single message implies multi-message security

PKE Security

IND-CPA Security

$$b \xleftarrow{\$} \{0,1\}$$

$$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$$

$$\mathsf{ct}_b \leftarrow \mathsf{Enc}(pk, m_b)$$

$$\xrightarrow{\quad pk \quad}$$

$$\xleftarrow{\quad (m_0, m_1) \quad}$$

$$\xrightarrow{\quad \mathsf{ct}_b \quad}$$

$$\xleftarrow{\quad b' \quad}$$

Wins if $b' = b$

PKE Security

IND-CPA Security

A public key encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$b \xleftarrow{\$} \{0,1\}$$

$$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$$

$$\mathsf{ct}_b \leftarrow \mathsf{Enc}(pk, m_b)$$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \mathsf{ct}_b \quad}$

$\xleftarrow{\quad b' \quad}$

Wins if $b' = b$

PKE Security

IND-CPA Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins GuessGame}] \leq \frac{1}{2} + \nu(\lambda)$$

$b \xleftarrow{\$} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(pk, m_b)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \mathsf{ct}_b \quad}$

$\xleftarrow{\quad b' \quad}$

Wins if $b' = b$

PKE Security

<u>IND-CPA Security</u>

$$\Big|\Pr[\mathscr{A} \text{ outputs } 1 \text{ in } \mathsf{Game}_0] - \Pr[\mathscr{A} \text{ outputs } 1 \text{ in } \mathsf{Game}_1]\Big| \leq \nu(\lambda)$$

Game_0

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(pk, m_0)$

$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

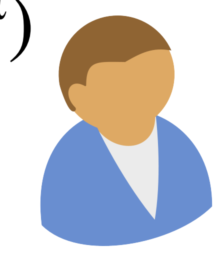
$\xrightarrow{\quad \mathsf{ct} \quad}$

$\xleftarrow{\quad b' \quad}$

Game_1

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(pk, m_1)$



$\xrightarrow{\quad pk \quad}$

$\xleftarrow{\quad (m_0, m_1) \quad}$

$\xrightarrow{\quad \mathsf{ct} \quad}$

$\xleftarrow{\quad b' \quad}$

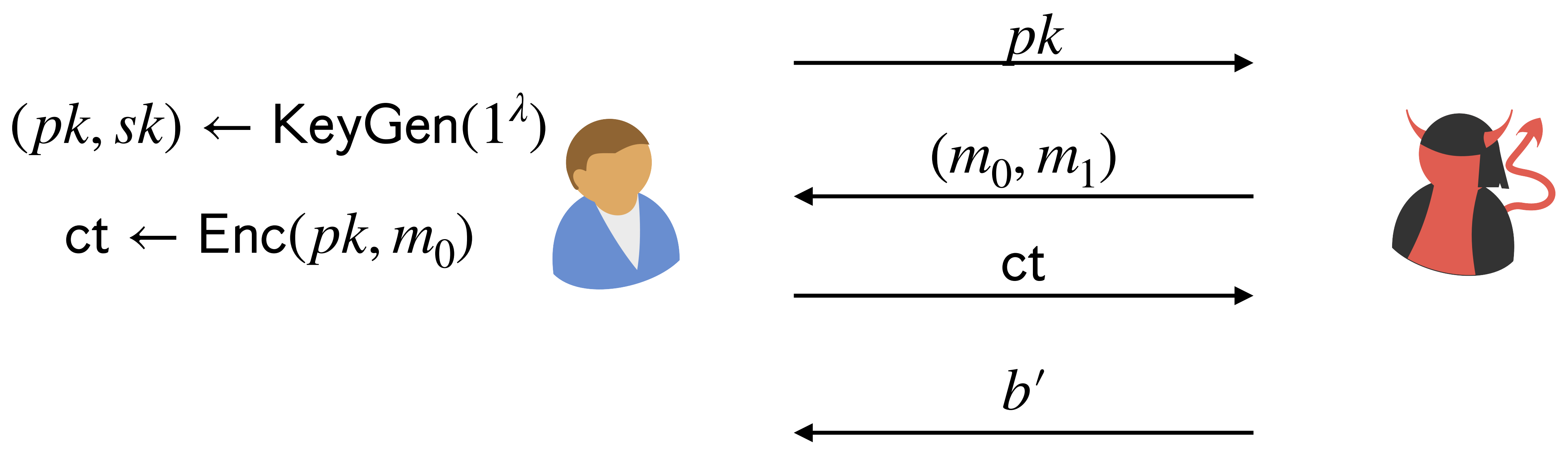
PKE Security

IND-CPA Security

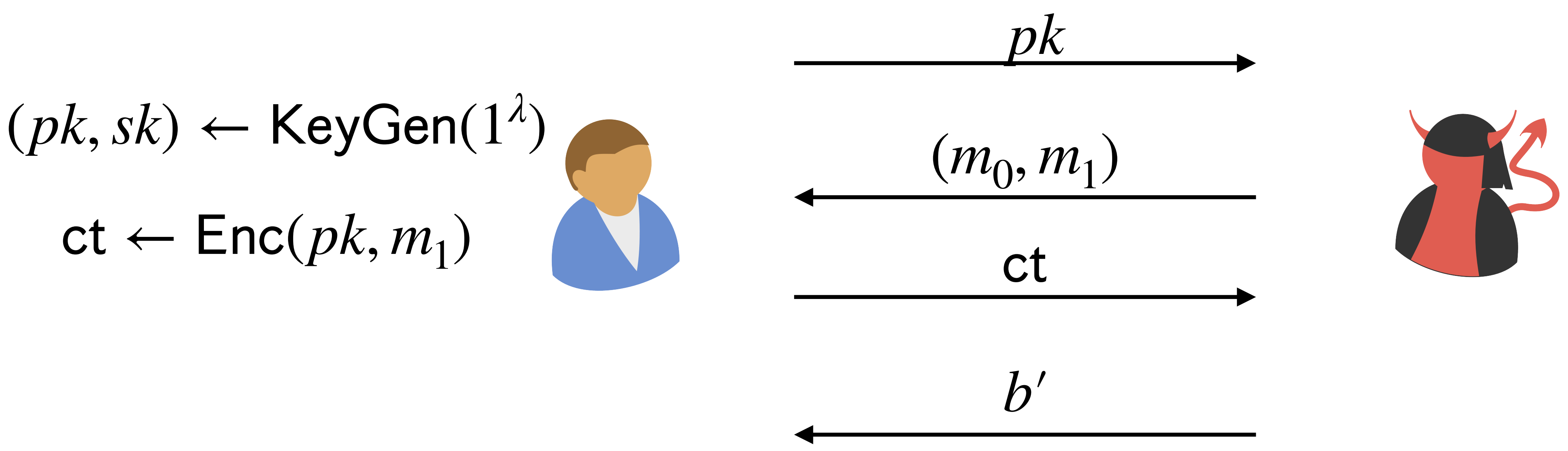
A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Big| \Pr[\mathscr{A} \text{ outputs } 1 \text{ in Game}_0] - \Pr[\mathscr{A} \text{ outputs } 1 \text{ in Game}_1] \Big| \leq \nu(\lambda)$$

Game$_0$



Game$_1$



Constructing PKE

Constructing PKE

- We built a secret key IND-CPA secure encryption assuming only the existence of PRGs.

Constructing PKE

- We built a secret key IND-CPA secure encryption assuming only the existence of PRGs.
- Building PKE seems to require some sort of *new* assumption.

Constructing PKE

- We built a secret key IND-CPA secure encryption assuming only the existence of PRGs.
- Building PKE seems to require some sort of *new* assumption.
    - RSA, Diffie-Hellman, LWE, etc.

Groups

Groups

- A group $\mathscr{G}$ is defined by:

Groups

- A group $\mathscr{G}$ is defined by:
  - A set of elements G

Groups

- A group $\mathscr{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \rightarrow G$

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $*: G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$

Groups

- A group $\mathscr{G}$ is defined by:
  - A set of elements G
  - An operation $*: G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
  - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
  - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$
- Abelian Groups: $a * b = b * a$

Groups

- A group $\mathcal{G}$ is defined by:
    - A set of elements G
    - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
    - Closure: For all $a, b \in G$, we have that $a * b \in G$
    - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
    - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
    - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$
- Abelian Groups: $a * b = b * a$

Example Group: $(\mathbb{Z}, +)$

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
  - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$
- Abelian Groups: $a * b = b * a$

Example Group: $(\mathbb{Z}, +)$

- 4 + 5 = 9

Groups

- A group $\mathscr{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
  - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$
- Abelian Groups: $a * b = b * a$

Example Group: $(\mathbb{Z}, +)$

- 4 + 5 = 9
- (4 + 5) + 6 = 4 + (5 + 6)

Groups

- A group $\mathcal{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
  - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$
- Abelian Groups: $a * b = b * a$

Example Group: $(\mathbb{Z}, +)$

- 4 + 5 = 9
- (4 + 5) + 6 = 4 + (5 + 6)
- 4 + 0 = 4

Groups

- A group $\mathscr{G}$ is defined by:
  - A set of elements G
  - An operation $* : G \times G \to G$
- $(G, *)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have that $a * b \in G$
  - Associativity: For all $a, b, c \in G$, we have that $(a * b) * c = a * (b * c)$
  - Identity: There exists an element $e \in G$ such that for all $a \in G$, we have that $e * a = a$
  - Inverse: For all $a \in G$, there exists $b \in G$ such that $a * b = e$
- Abelian Groups: $a * b = b * a$

Example Group: $(\mathbb{Z}, +)$

- 4 + 5 = 9
- (4 + 5) + 6 = 4 + (5 + 6)
- 4 + 0 = 4
- 4 + (-4) = 0

Cyclic Groups

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$
- Notation: let n be the *order* of the group, i.e. $n = |G|$

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$
- Notation: let n be the *order* of the group, i.e. $n = |G|$
- A group $(G, *)$ is a *cyclic* group if there it is generated by a single element

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$
- Notation: let n be the *order* of the group, i.e. $n = |G|$
- A group $(G, *)$ is a *cyclic* group if there it is generated by a single element
  - $G = \left\{1 = e = g^0, g^1, g^2, \ldots, g^{n-1}\right\}$

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$
- Notation: let n be the *order* of the group, i.e. $n = |G|$
- A group $(G, *)$ is a *cyclic* group if there it is generated by a single element
  - $G = \{1 = e = g^0, g^1, g^2, \ldots, g^{n-1}\}$
  - g is a *generator* of G

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$
- Notation: let n be the *order* of the group, i.e. $n = |G|$
- A group $(G, *)$ is a *cyclic* group if there it is generated by a single element
  - $G = \{1 = e = g^0, g^1, g^2, \ldots, g^{n-1}\}$
  - g is a *generator* of G
- We write this as $G = \langle g \rangle$

Cyclic Groups

- Notation: for $g \in G$, we have that $g^3 = g * g * g$
- Notation: let n be the *order* of the group, i.e. $n = |G|$
- A group $(G, *)$ is a *cyclic* group if there it is generated by a single element
  - $G = \{1 = e = g^0, g^1, g^2, \ldots, g^{n-1}\}$
  - g is a *generator* of G
- We write this as $G = \langle g \rangle$
- Our assumption: given g^a for some randomly sampled a, it should be hard to find a

Discrete Logarithm Assumption

Discrete Logarithm Assumption

Discrete Logarithm Assumption

$a \xleftarrow{\$} \{0, \ldots, p-1\}$

Discrete Logarithm Assumption

$a \xleftarrow{\$} \{0, \ldots, p-1\}$

$$\xrightarrow{(G, p, g, g^a)}$$

Discrete Logarithm Assumption

$a \xleftarrow{\$} \{0, \ldots, p-1\}$

$$\xrightarrow{(G, p, g, g^a)}$$

$$\xleftarrow{a'}$$

Discrete Logarithm Assumption

$a \xleftarrow{\$} \{0, \ldots, p-1\}$

$$(G, p, g, g^a) \longrightarrow$$

$$\xleftarrow{\quad a' \quad}$$

Wins if $a' = a$

Discrete Logarithm Assumption

Discrete Logarithm Assumption

Let $(G, *)$ be a cyclic group of order p (where p is a safe prime) with generator g, then for every NUPPT adversary $\mathscr{A}$, there exists a negligible function ν such that

$$\Pr\left[\mathscr{A} \text{ wins DLGame}\right] \leq \nu(\lambda)$$

$a \xleftarrow{\$} \{0, \ldots, p-1\}$

$\xrightarrow{(G, p, g, g^a)}$

$\xleftarrow{a'}$

Wins if $a' = a$

Discrete Logarithm Assumption

$p = 2q + 1$ for some large prime q

<u>Discrete Logarithm Assumption</u>

Let $(G, *)$ be a cyclic group of order p (where p is a safe prime) with generator g, then for every NUPPT adversary $\mathscr{A}$, there exists a negligible function ν such that

$$\Pr\left[\mathscr{A} \text{ wins DLGame}\right] \leq \nu(\lambda)$$

$a \xleftarrow{\$} \{0,\ldots,p-1\}$

$\xrightarrow{(G, p, g, g^a)}$

$\xleftarrow{a'}$

Wins if $a' = a$

Discrete Logarithm Assumption

$p = 2q + 1$ for some large prime q

<u>Discrete Logarithm Assumption</u>

Let $(G, *)$ be a cyclic group of order p (where p is a safe prime) with generator g, then for every NUPPT adversary $\mathscr{A}$, there exists a negligible function ν such that

$$\Pr\left[\mathscr{A} \text{ wins DLGame}\right] \leq \nu(\lambda)$$

$a \xleftarrow{\$} \{0, \ldots, p-1\}$

$$\xrightarrow{(G, p, g, g^a)}$$

$$\xleftarrow{a'}$$

Wins if $a' = a$

Note: This is a *search problem*. $\mathscr{A}$ needs to actually *find* the discrete logarithm