Authentication I

601.442/642 Modern Cryptography

10th March 2026

Logistics

Logistics

- HW 6 due Thursday

Logistics

- HW 6 due Thursday
- Isn’t Goldreich-Levin cool?

Logistics

- HW 6 due Thursday
- Isn’t Goldreich-Levin cool?
  - A generic way to turn *search* problems into *decision* problems

Logistics

- HW 6 due Thursday
- Isn’t Goldreich-Levin cool?
    - A generic way to turn *search* problems into *decision* problems
    - Search: given $y = f(x)$, find x

Logistics

- HW 6 due Thursday
- Isn't Goldreich-Levin cool?
  - A generic way to turn *search* problems into *decision* problems
  - Search: given $y = f(x)$, find x
  - Decision: given $y = f(x)$, *predict* $\mathsf{hc}(x)$

A Note on Notation

A Note on Notation

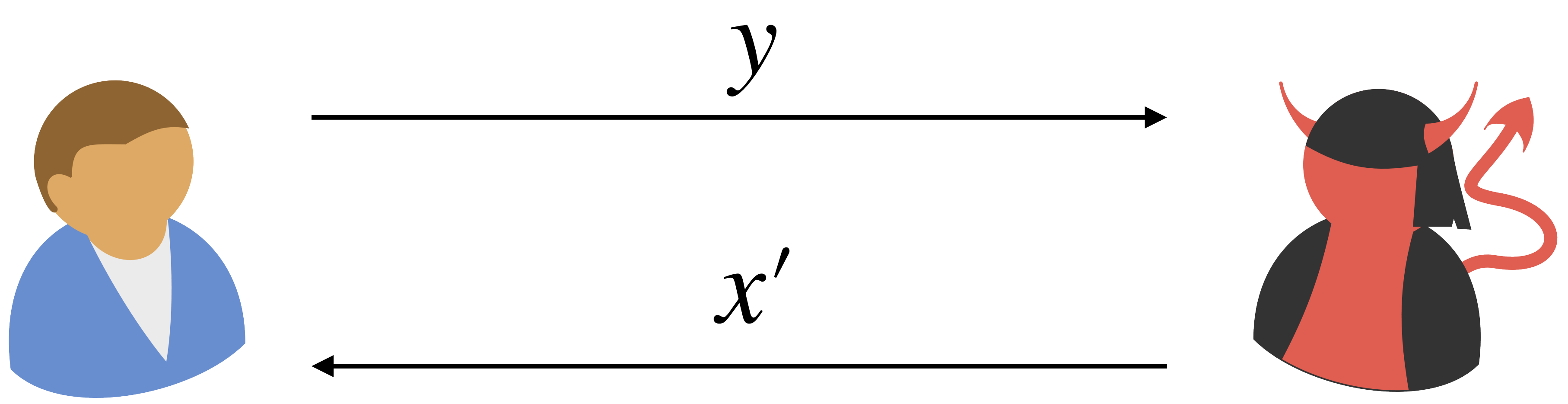
$$\Pr\left[f(x') = y \;:\; \begin{array}{r} x \xleftarrow{\$} \{0,1\}^\lambda \\ y := f(x) \\ x' \xleftarrow{\$} \mathscr{A}(1^\lambda, y) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

A Note on Notation

$$\Pr\left[f(x') = y \ : \ \begin{array}{r} x \xleftarrow{\$} \{0,1\}^\lambda \\ y := f(x) \\ x' \xleftarrow{\$} \mathscr{A}(1^\lambda, y) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

$$\Pr\left[\mathscr{A} \text{ wins OWFGame}\right] \leq \mathsf{negl}(\lambda)$$

$$x \xleftarrow{\$} \{0,1\}^\lambda$$
$$y := f(x)$$

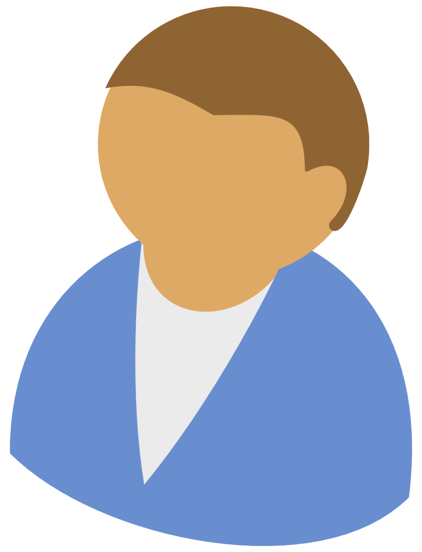


Wins if $f(x') = y$

Authentication

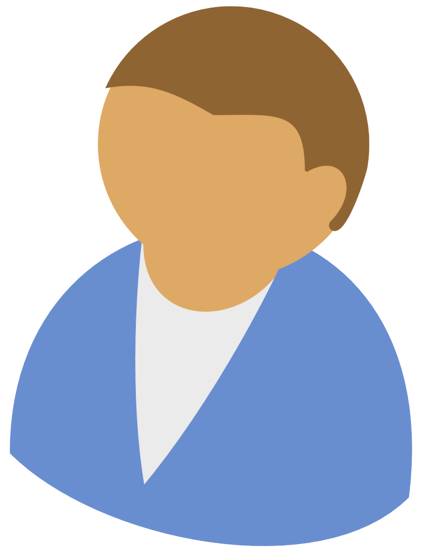
Authentication

Authentication



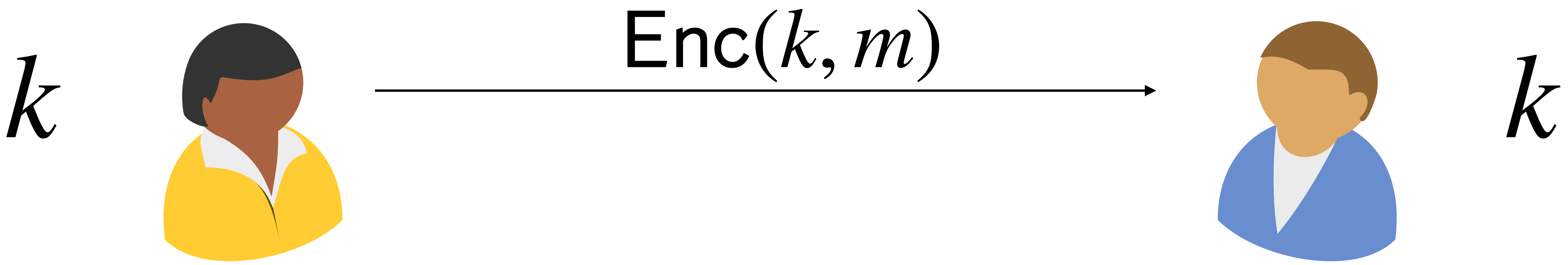
Authentication

k

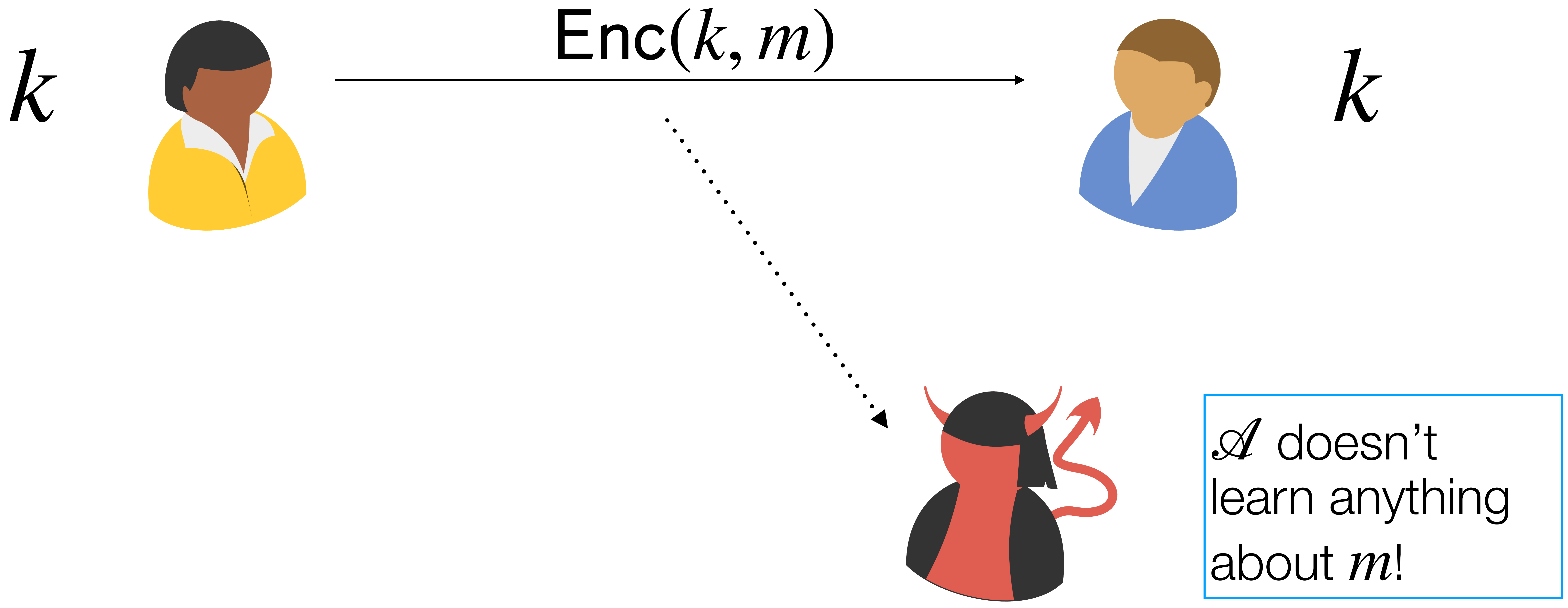


k

Authentication



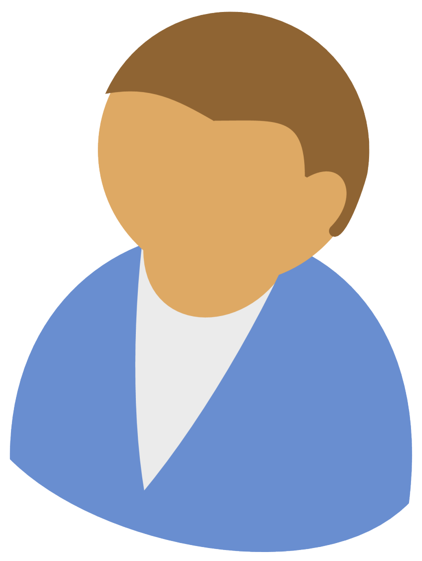
Authentication



Authentication

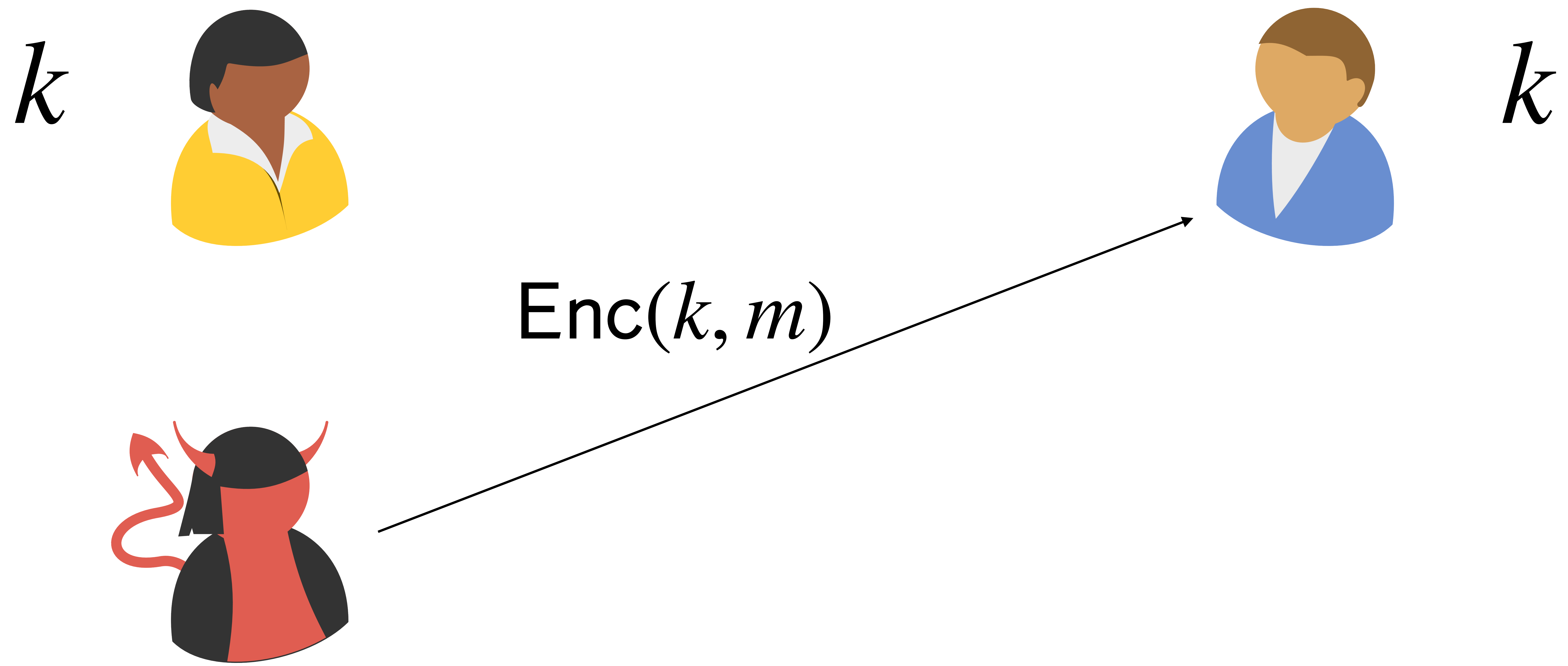
Authentication

k 

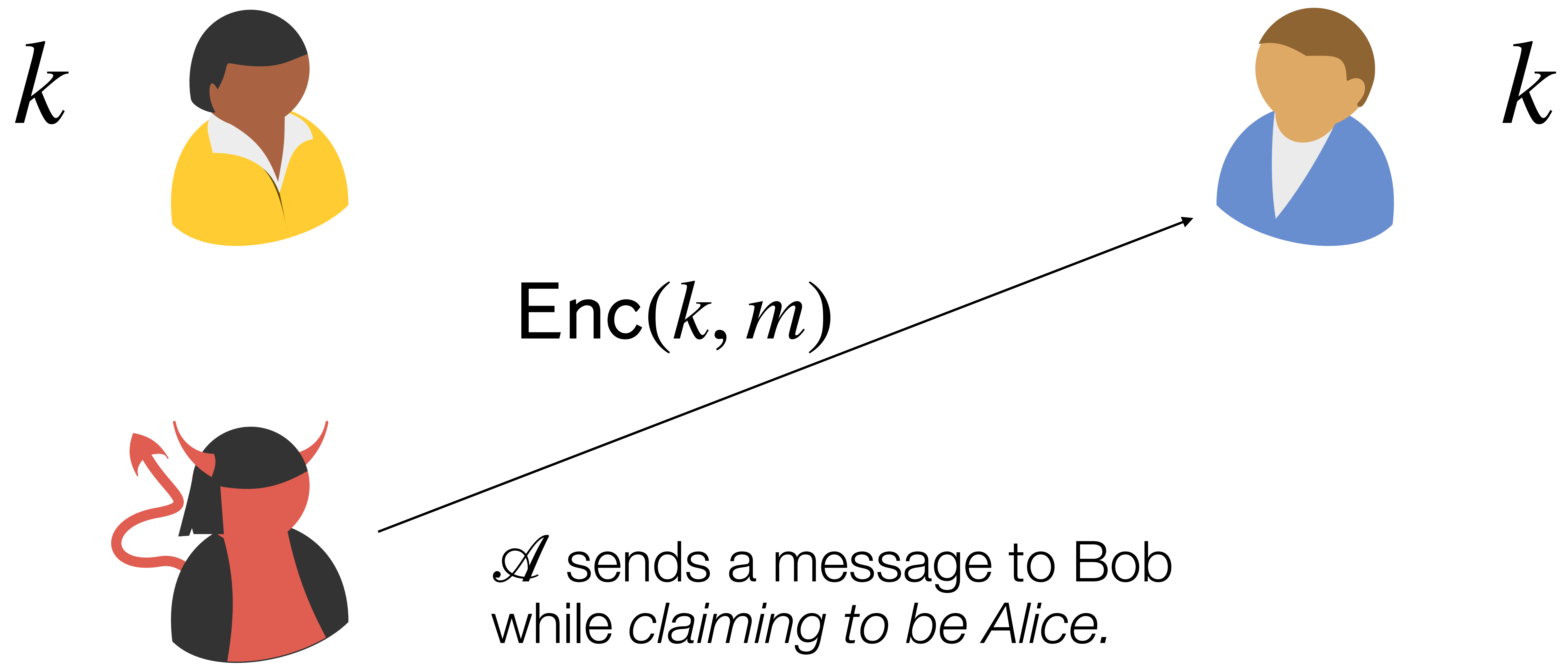
 k

Authentication

Authentication

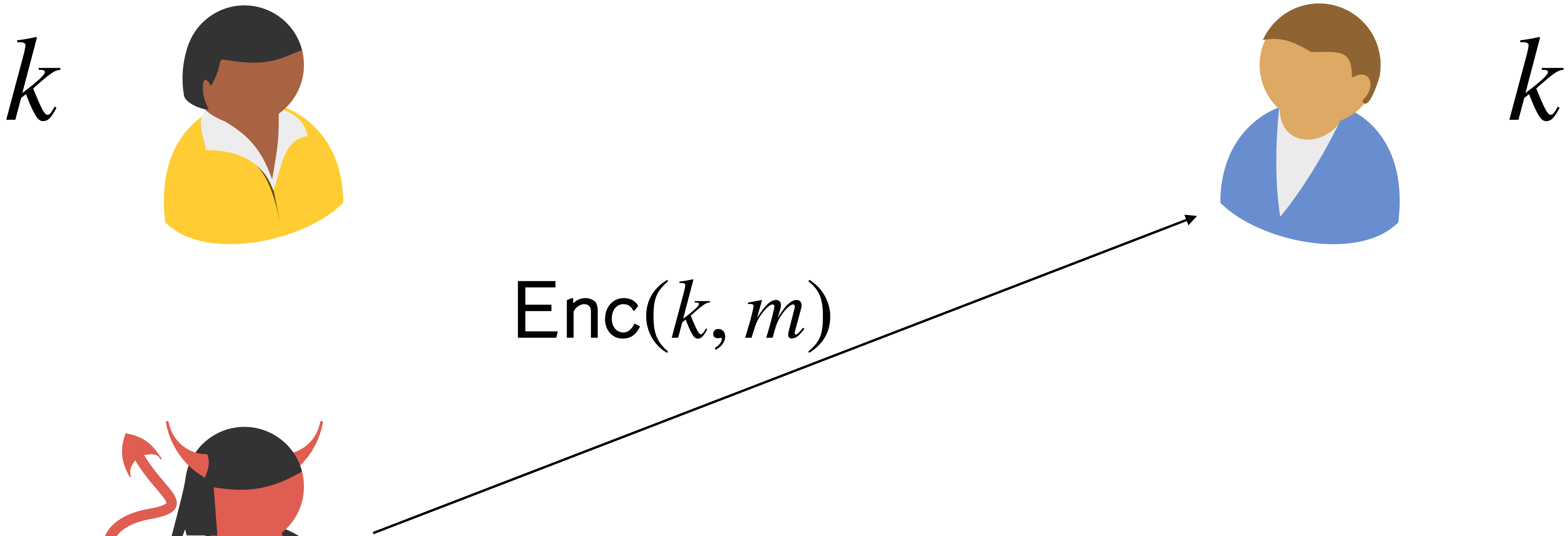


Authentication



$\mathscr{A}$ sends a message to Bob
while *claiming to be Alice.*

Authentication

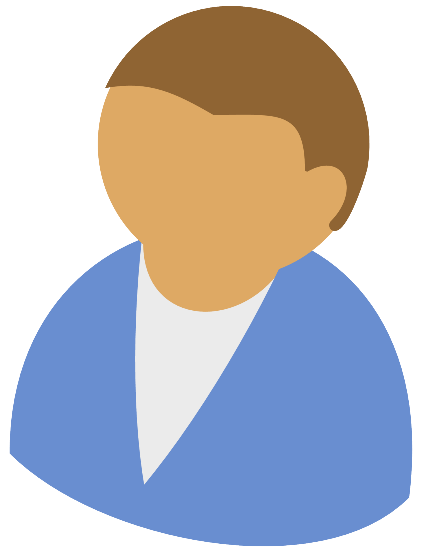


$\mathscr{A}$ sends a message to Bob while *claiming to be Alice.*

How can Bob tell that a message *really did* come from Alice?

Authentication

k



k

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

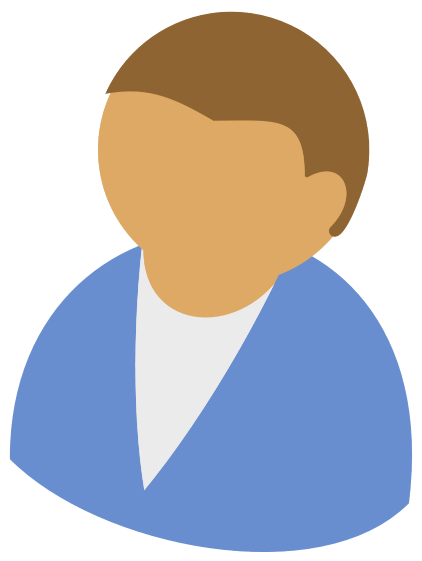
$\mathscr{A}$ sends a message to Bob while *claiming to be Alice.*

How can Bob tell that a message *really did* come from Alice?

Goal: something like a *signature*

Authentication

k



k

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

$\mathscr{A}$ sends a message to Bob while *claiming to be Alice.*

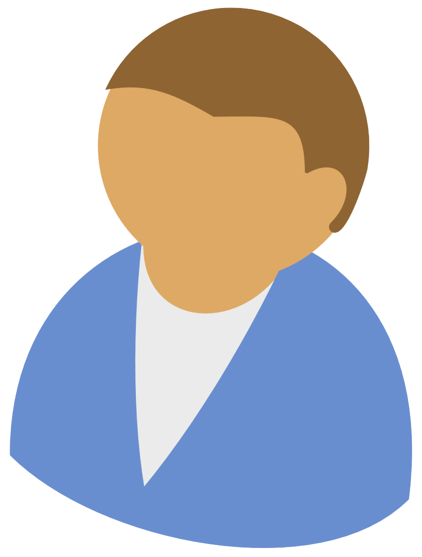
How can Bob tell that a message *really did* come from Alice?

Goal: something like a *signature*

- Alice can "sign" a message m to produce a signature σ

Authentication

k



k

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

$\mathscr{A}$ sends a message to Bob while *claiming to be Alice.*

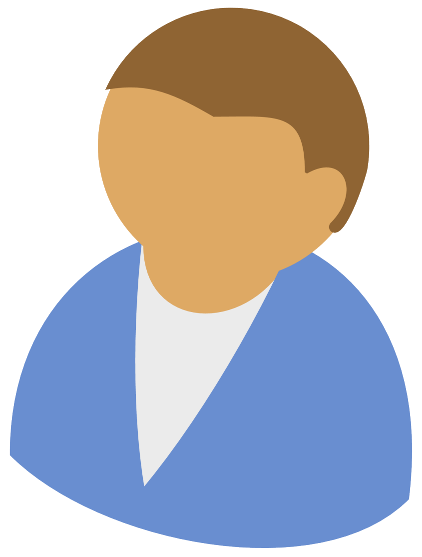
How can Bob tell that a message *really did* come from Alice?

Goal: something like a *signature*

- Alice can "sign" a message m to produce a signature σ
- Bob can *verify* that σ is correct for m

Authentication

k



k

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

$\mathscr{A}$ sends a message to Bob while *claiming to be Alice.*

How can Bob tell that a message *really did* come from Alice?

Goal: something like a *signature*

- Alice can "sign" a message m to produce a signature σ
- Bob can *verify* that σ is correct for m
- $\mathscr{A}$ cannot *forge* a signature

Authentication

Authentication

- Our goal is authentication: the party sending a message includes a signature that proves it really is them sending it

Authentication

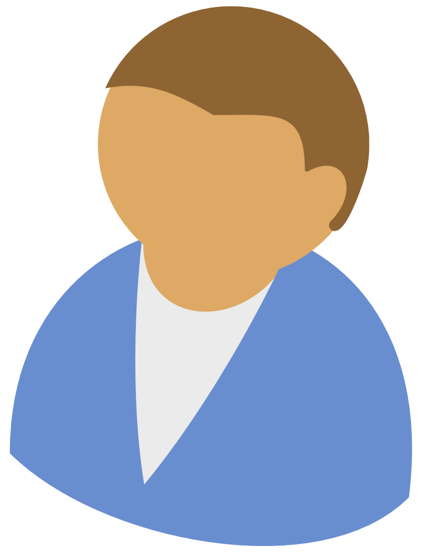
- Our goal is authentication: the party sending a message includes a signature that proves it really is them sending it
- Private Key: Message Authentication Codes (MACs)

Authentication

- Our goal is authentication: the party sending a message includes a signature that proves it really is them sending it
- Private Key: Message Authentication Codes (MACs)
- Public Key: Digital Signatures

Message Authentication Code (MAC)

Message Authentication Code (MAC)

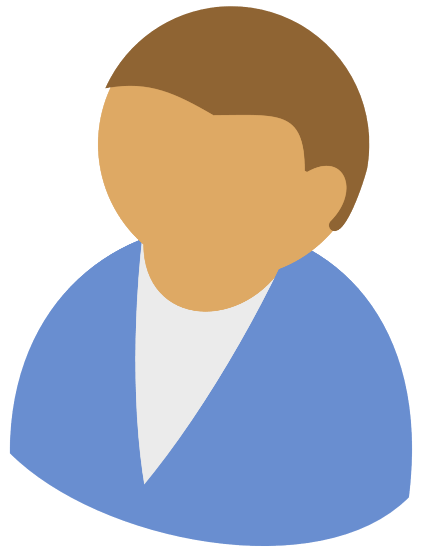


Message Authentication Code (MAC)

Message Authentication Code (MAC)

k

m



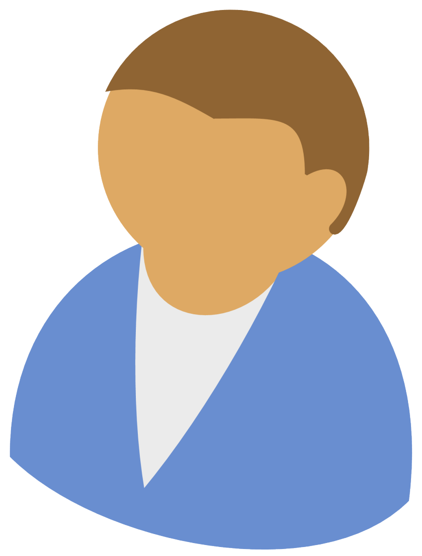
k

Message Authentication Code (MAC)

k

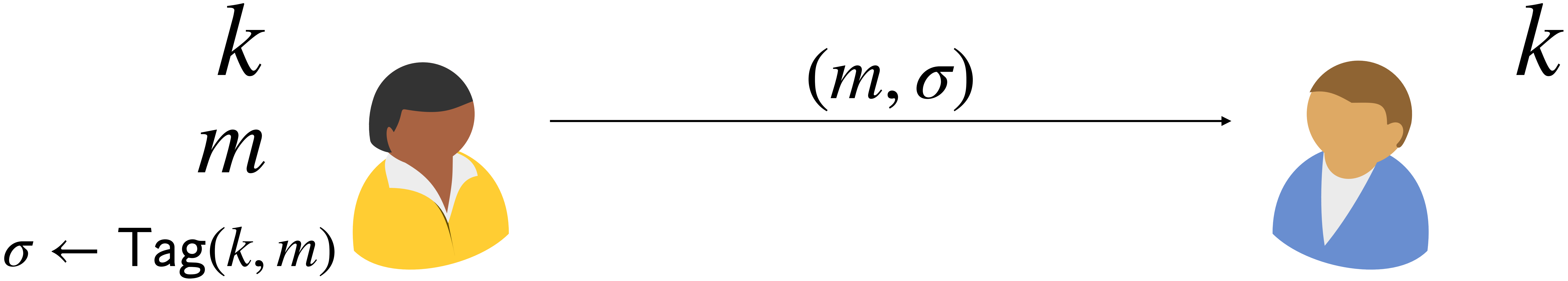
m

$\sigma \leftarrow \mathsf{Tag}(k, m)$



k

Message Authentication Code (MAC)



Message Authentication Code (MAC)

Message Authentication Code (MAC)

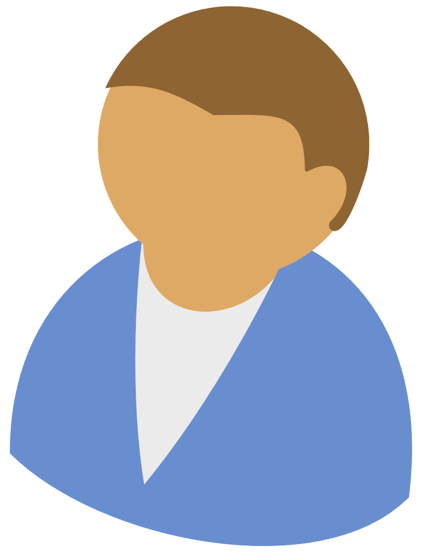
Message Authentication Code (MAC)

k

m

$\sigma \leftarrow \mathsf{Tag}(k, m)$

(m, σ)



k

If $\mathsf{Ver}(k, m, \sigma) = 1$, accept

else, reject

Message Authentication Code (MAC)

Message Authentication Code (MAC)

Message Authentication Code (MAC)

Goal: $\mathscr{A}$ can't produce a (m', σ') pair that verifies: even after seeing a bunch of valid signatures (m, σ)

Message Authentication Code (MAC)

Message Authentication Code Scheme Syntax

Message Authentication Code (MAC)

Message Authentication Code Scheme Syntax

A *message authentication code scheme* consists of three (possibly probabilistic) algorithms:

Message Authentication Code (MAC)

Message Authentication Code Scheme Syntax

A *message authentication code scheme* consists of three (possibly probabilistic) algorithms:

- KeyGen$(1^\lambda) \rightarrow k$ outputs a key $k \in \mathscr{K}$

Message Authentication Code (MAC)

Message Authentication Code Scheme Syntax

A *message authentication code scheme* consists of three (possibly probabilistic) algorithms:

- KeyGen$(1^\lambda) \to k$ outputs a key $k \in \mathscr{K}$
- Tag$(k, m) \to \sigma$ takes as input a key and a message $m \in \mathscr{M}$ and outputs a signature $\sigma \in \mathscr{T}$

Message Authentication Code (MAC)

Message Authentication Code Scheme Syntax

A *message authentication code scheme* consists of three (possibly probabilistic) algorithms:

- KeyGen$(1^\lambda) \to k$ outputs a key $k \in \mathcal{K}$
- Tag$(k, m) \to \sigma$ takes as input a key and a message $m \in \mathcal{M}$ and outputs a signature $\sigma \in \mathcal{T}$
- Ver$(k, m, \sigma) \to b$ takes as input a key, a message, and a signature and outputs a bit

Message Authentication Code (MAC)

Message Authentication Code Scheme Syntax

A *message authentication code scheme* consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}(1^\lambda) \rightarrow k$ outputs a key $k \in \mathcal{K}$
- $\mathsf{Tag}(k, m) \rightarrow \sigma$ takes as input a key and a message $m \in \mathcal{M}$ and outputs a signature $\sigma \in \mathcal{T}$
- $\mathsf{Ver}(k, m, \sigma) \rightarrow b$ takes as input a key, a message, and a signature and outputs a bit

Correctness: $$\Pr\left[\mathsf{Ver}(k, m, \sigma) = 1 \;:\; \begin{matrix} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \sigma \leftarrow \mathsf{Tag}(k, m) \end{matrix}\right] = 1$$

MAC Security

UF-CMA Security

MAC Security

UF-CMA Security

A MAC scheme $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

MAC Security

UF-CMA Security

A MAC scheme $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins } \mathsf{MACGame}] \leq \mathsf{negl}(\lambda)$$

MAC Security

UF-CMA Security

A MAC scheme $(\text{KeyGen}, \text{Tag}, \text{Ver})$ satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\text{negl}(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

MAC Security

UF-CMA Security

A MAC scheme $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \mathsf{negl}(\lambda)$$

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

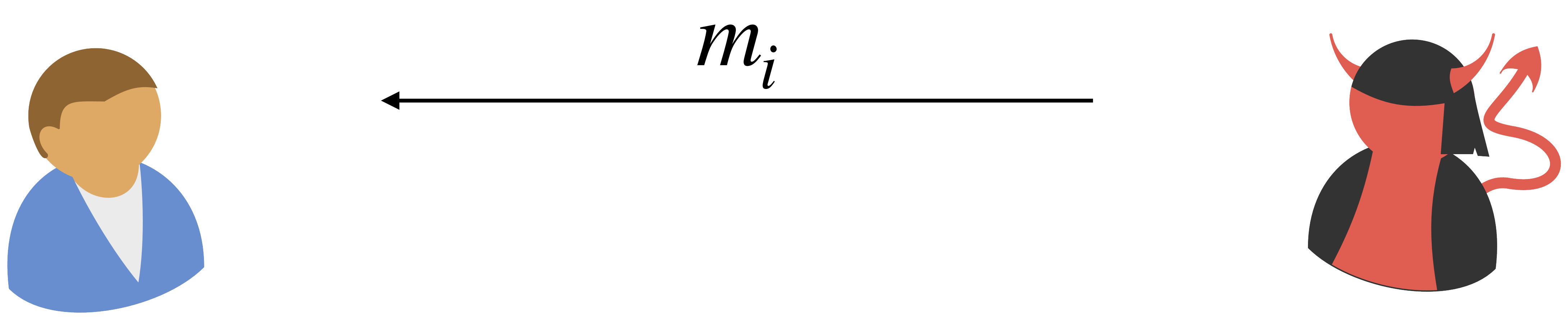
MAC Security

UF-CMA Security

A MAC scheme (KeyGen, Tag, Ver) satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function negl$(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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



MAC Security

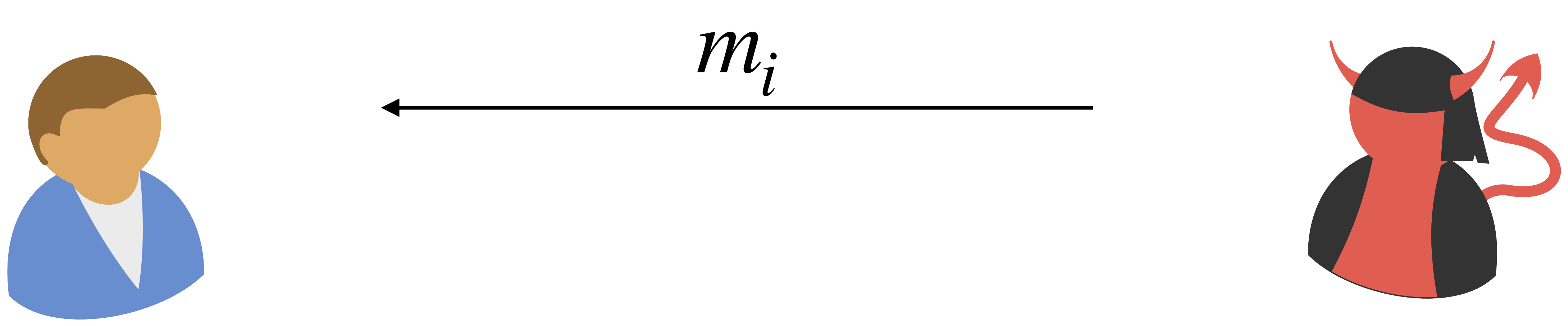
UF-CMA Security

A MAC scheme (KeyGen, Tag, Ver) satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function negl$(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



MAC Security

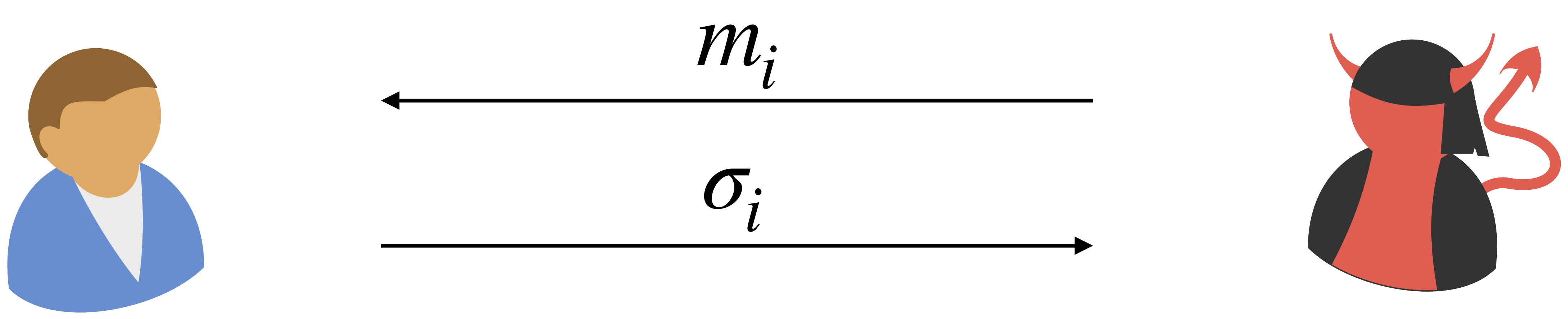
UF-CMA Security

A MAC scheme $(\text{KeyGen}, \text{Tag}, \text{Ver})$ satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\text{negl}(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



MAC Security

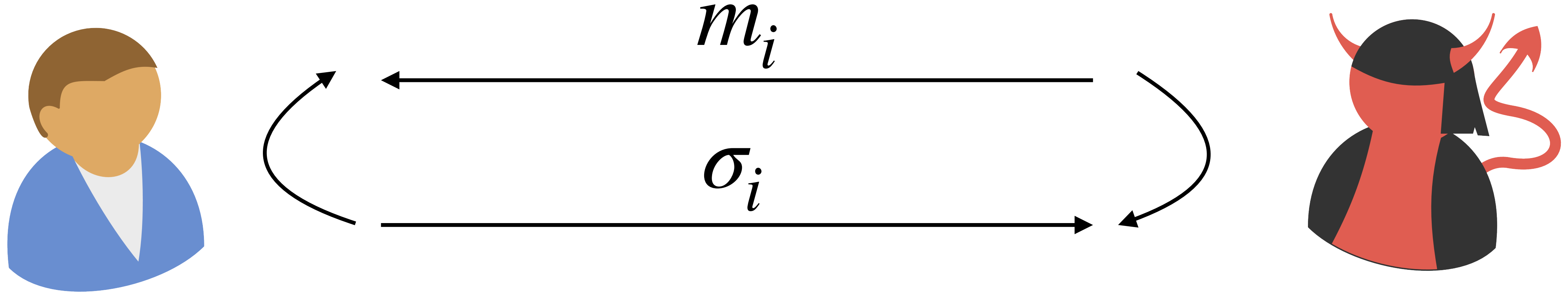
UF-CMA Security

A MAC scheme (KeyGen, Tag, Ver) satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function negl($\cdot$). such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



MAC Security

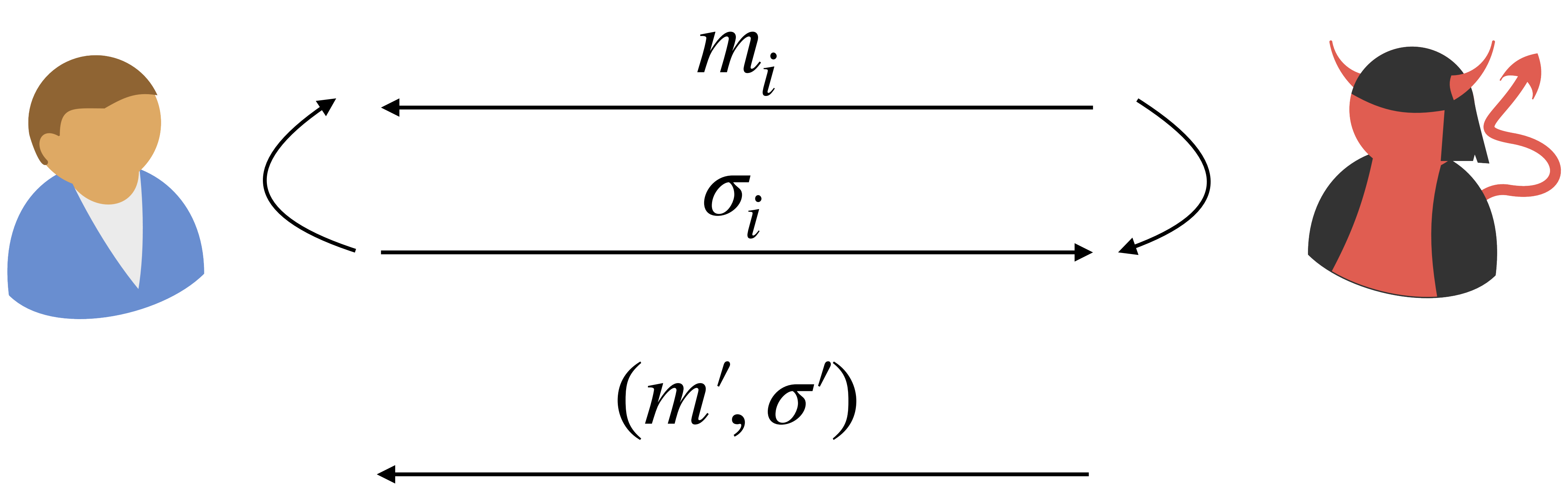
UF-CMA Security

A MAC scheme (KeyGen, Tag, Ver) satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function negl($\cdot$). such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



MAC Security

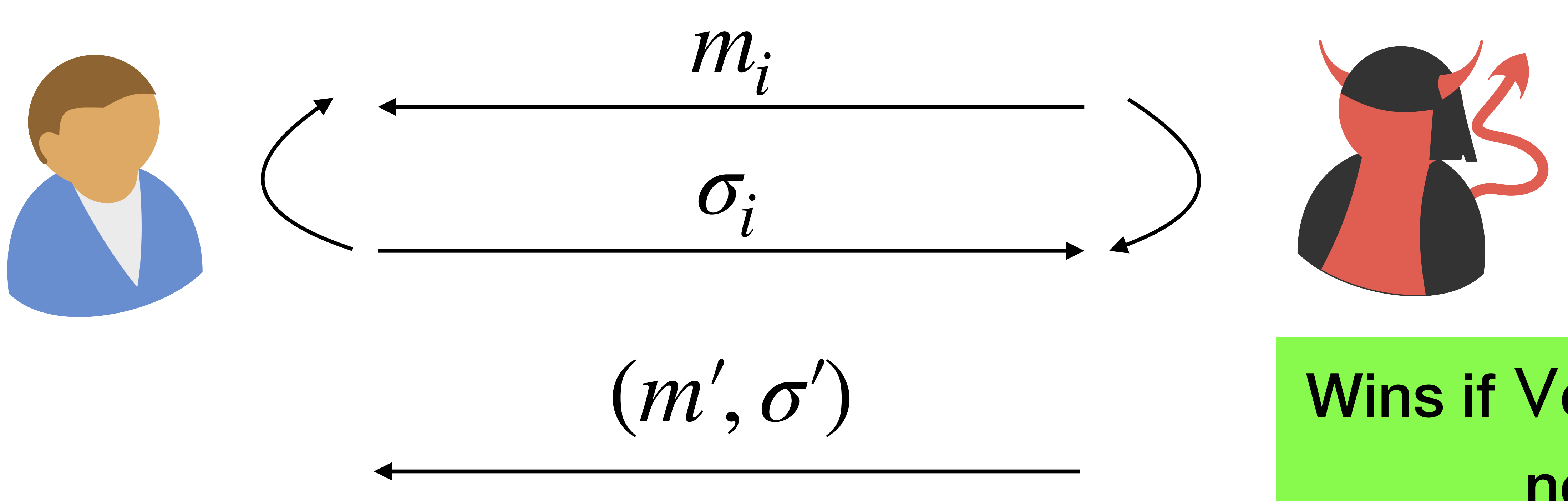
UF-CMA Security

A MAC scheme (KeyGen, Tag, Ver) satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function negl($\cdot$). such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



Wins if $\text{Ver}(k, m', \sigma') = 1$ **and** $\mathscr{A}$ **never queried** m'

MAC Security

This is a *search* problem!

UF-CMA Security

A MAC scheme (KeyGen, Tag, Ver) satisfies *unforgeability under chosen message attack* (UF-CMA) if for all NUPPT $\mathscr{A}$, there exists a negligible function negl$(\cdot)$. such that $\forall \lambda \in \mathbb{N}$:

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$

m_i

σ_i

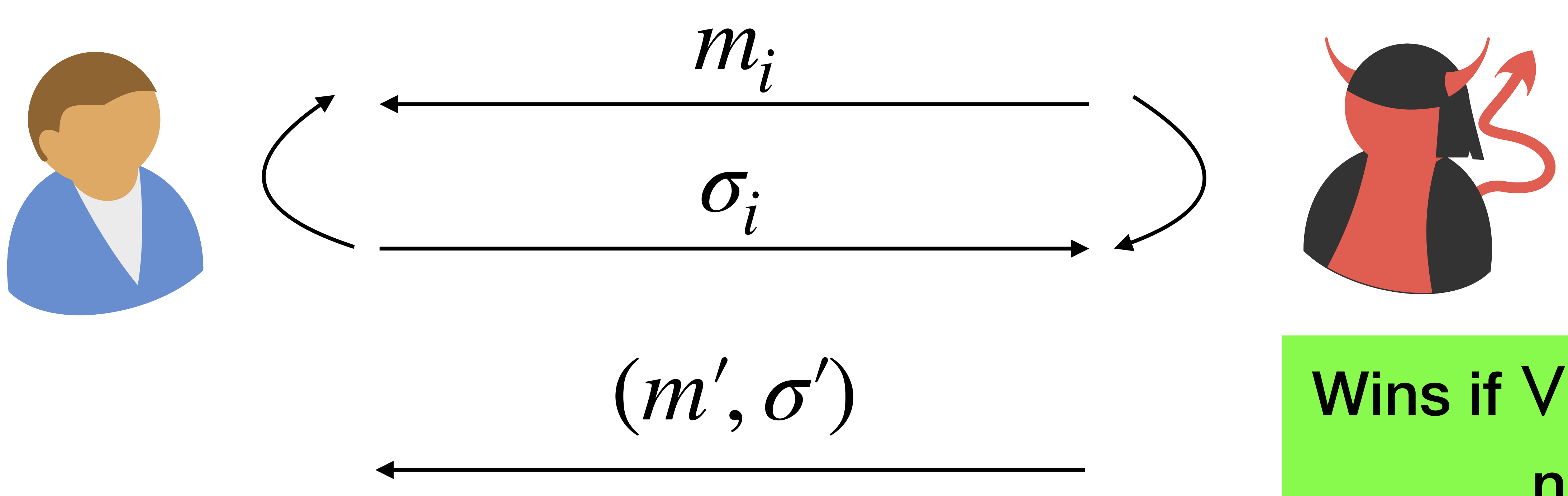
(m', σ')

Wins if $\text{Ver}(k, m', \sigma') = 1$ **and** $\mathscr{A}$ **never queried** m'

MAC Security

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



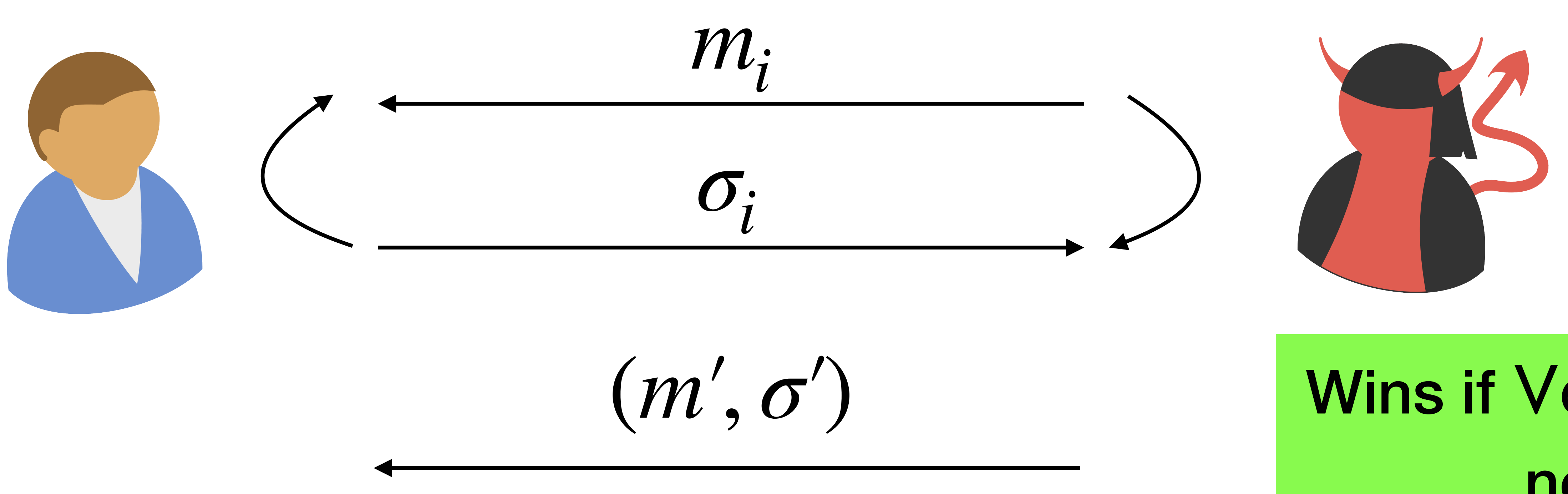
Wins if $\text{Ver}(k, m', \sigma') = 1$ and $\mathscr{A}$ never queried m'

MAC Security

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \text{negl}(\lambda)$$

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

$\sigma_i \leftarrow \text{Tag}(k, m_i)$



Wins if $\text{Ver}(k, m', \sigma') = 1$ and $\mathscr{A}$ never queried m'

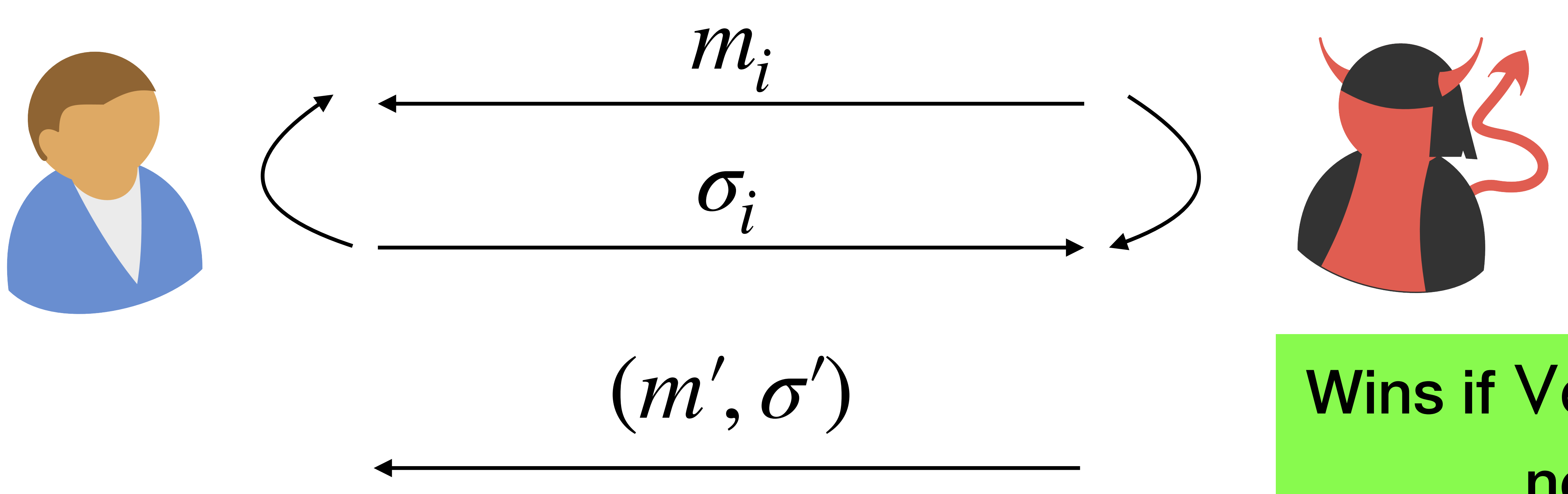
MAC Security

$$\Pr\left[\begin{array}{c}\mathsf{Ver}(k, m', \sigma') = 1 \\ \textbf{and } \mathscr{A} \text{ never queried } m'\end{array} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (m', \sigma') \leftarrow \mathscr{A}^{\mathsf{Tag}(k,\cdot)}(1^\lambda)\end{array}\right] \leq \mathsf{negl}(\lambda)$$

$$\Pr[\mathscr{A} \text{ wins MACGame}] \leq \mathsf{negl}(\lambda)$$

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

$\sigma_i \leftarrow \mathsf{Tag}(k, m_i)$



Wins if $\mathsf{Ver}(k, m', \sigma') = 1$ and $\mathscr{A}$ never queried m'

MAC Construction

MAC Construction

MACs can be built directly from PRFs

MAC Construction

MACs can be built directly from PRFs

PRF MAC

MAC Construction

MACs can be built directly from PRFs

PRF MAC

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

MAC Construction

MACs can be built directly from PRFs

PRF MAC

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

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

MAC Construction

MACs can be built directly from PRFs

PRF MAC

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

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$
- Tag(k, m): $\sigma := F_k(m)$

MAC Construction

MACs can be built directly from PRFs

PRF MAC

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

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$
- Tag(k, m): $\sigma := F_k(m)$
- Ver(k, m, σ): $F_k(m) \stackrel{?}{=} \sigma$

MAC Construction

MACs can be built directly from PRFs

PRF MAC

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

- KeyGen(1^λ): $k \xleftarrow{\$} \{0,1\}^\lambda$
- Tag(k, m): $\sigma := F_k(m)$
- Ver(k, m, σ): $F_k(m) \stackrel{?}{=} \sigma$

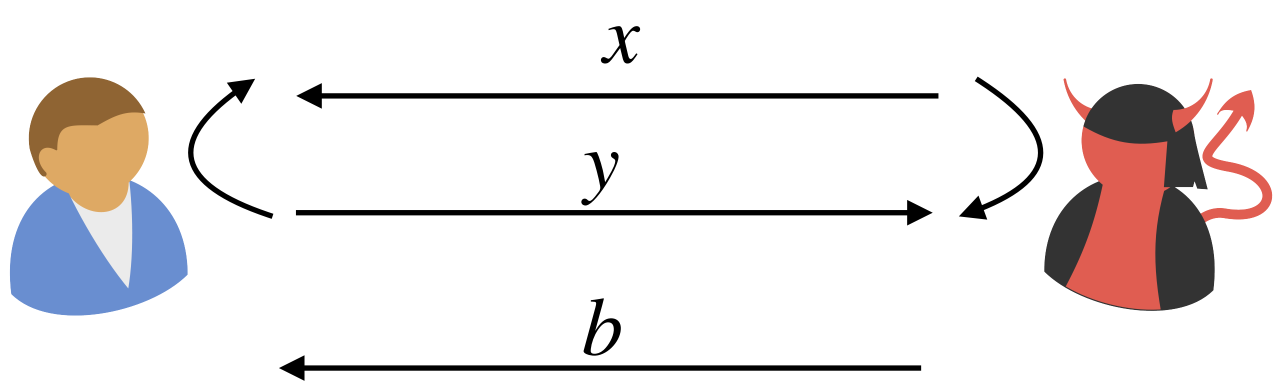
Intuition: to forge a signature $\mathscr{A}$ would need to *predict* the output of a PRF. This should be impossible!

Recap: PRF Security

Recap: PRF Security

Game$_0$

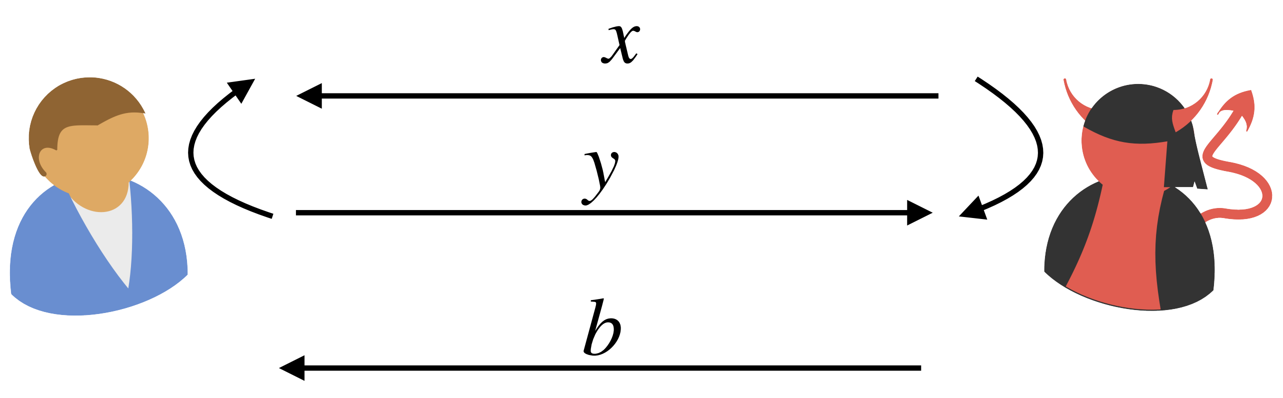
$k \xleftarrow{\$} \{0,1\}^\lambda \qquad y = F_k(x)$



Recap: PRF Security

Game_0

$k \xleftarrow{\$} \{0,1\}^\lambda \quad y = F_k(x)$



Game_1

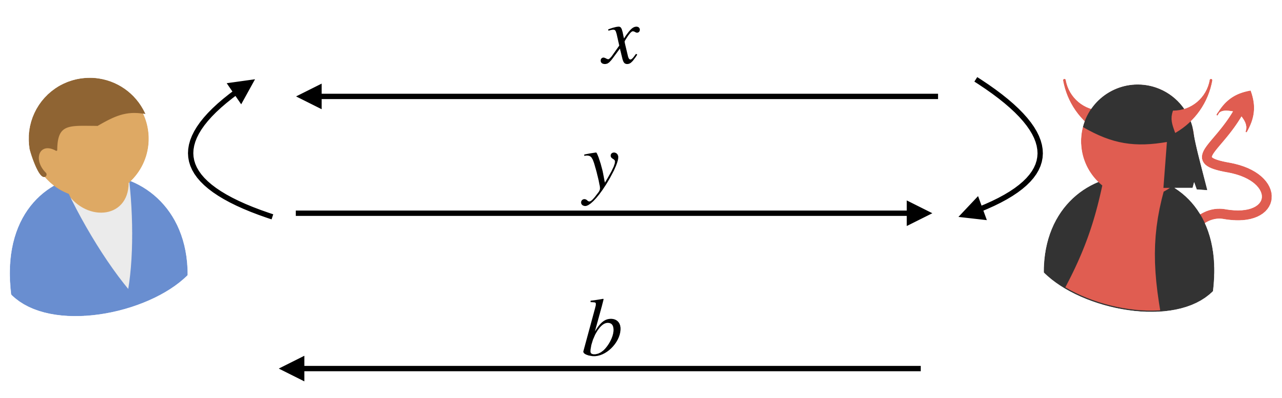
$T := \{\}$

if $x \notin T$

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

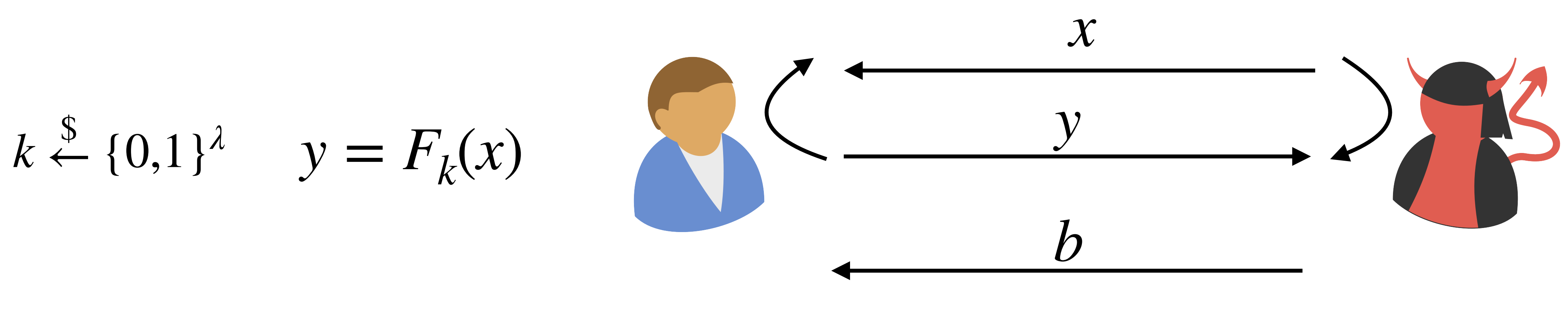
$T[x] = r$

$y := T[x]$

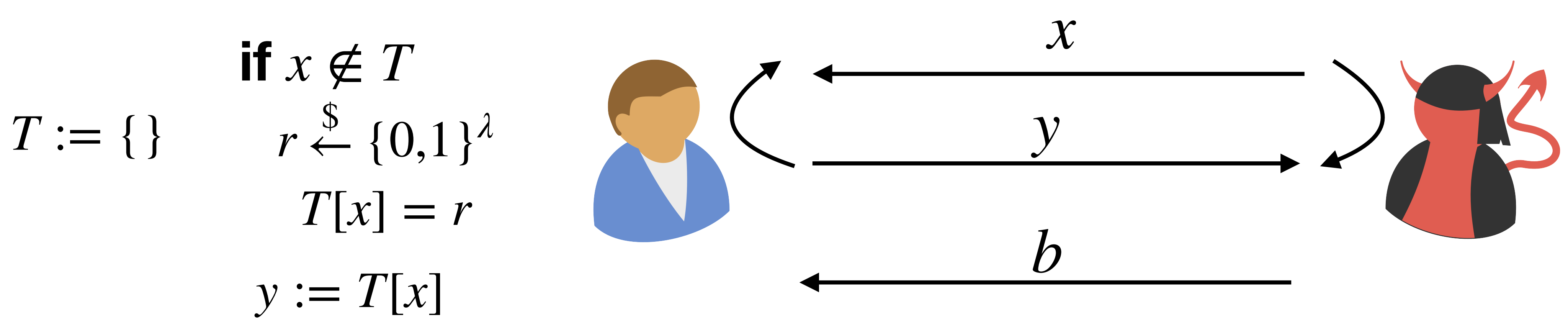


Recap: PRF Security

Game$_0$



Game$_1$

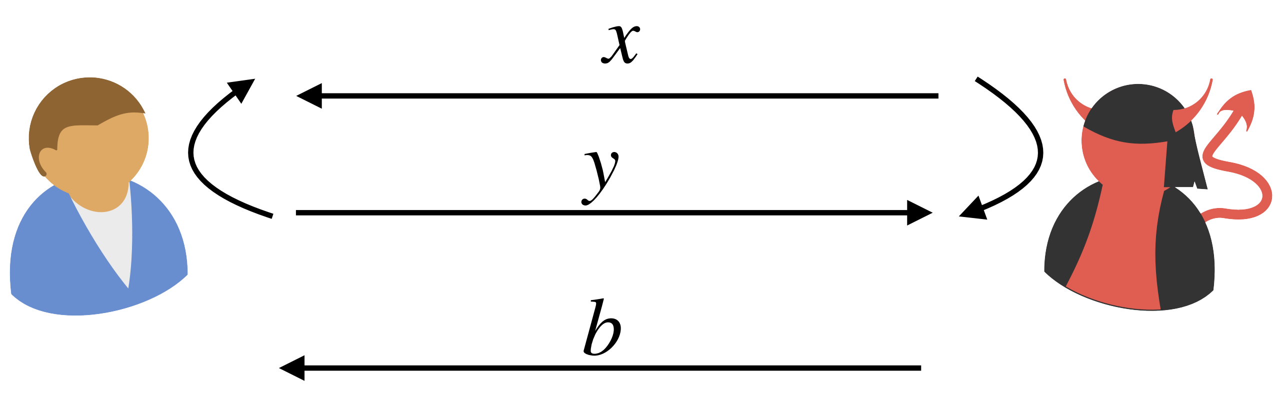


Let W_b be the event that $\mathscr{A}$ outputs 0 in Game$_b$

Recap: PRF Security

Game_0

$k \xleftarrow{\$} \{0,1\}^\lambda$ $\quad y = F_k(x)$



Game_1

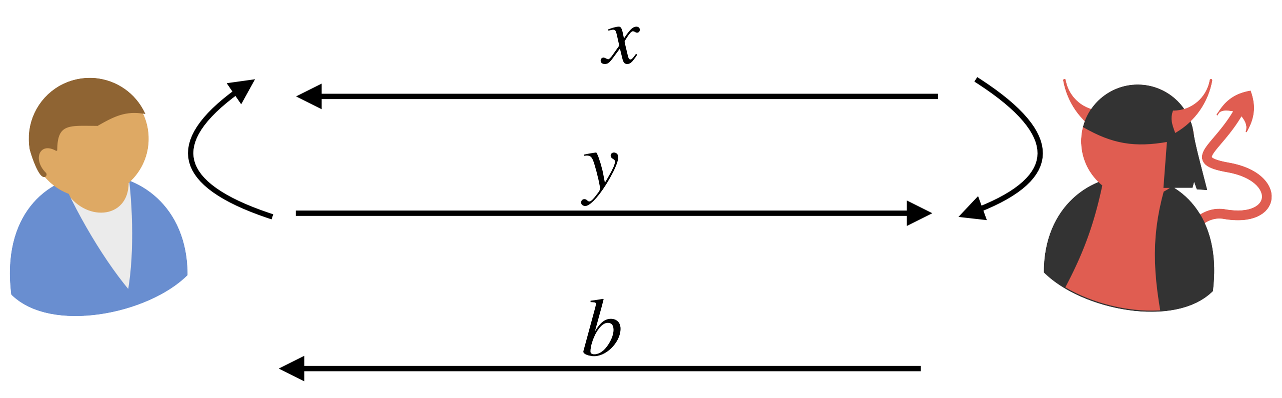
$T := \{\}$

if $x \notin T$

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

$T[x] = r$

$y := T[x]$

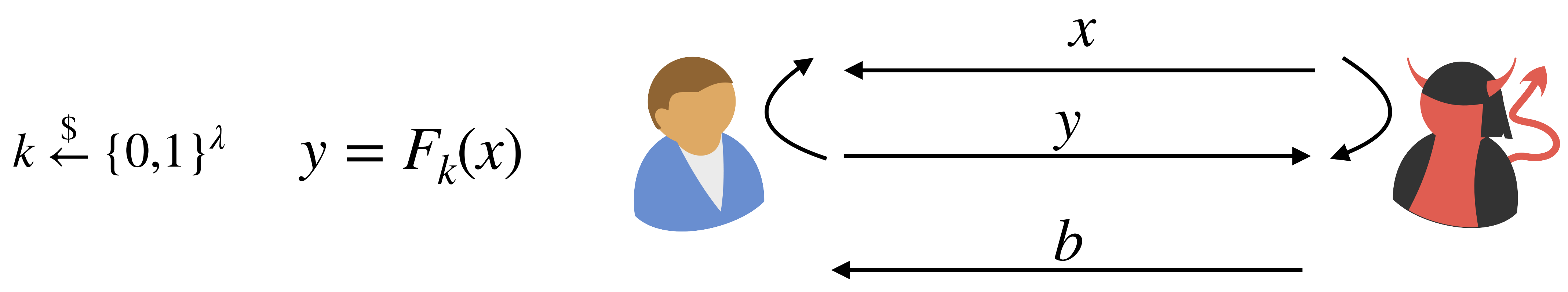


Let W_b be the event that $\mathscr{A}$ outputs 0 in Game_b

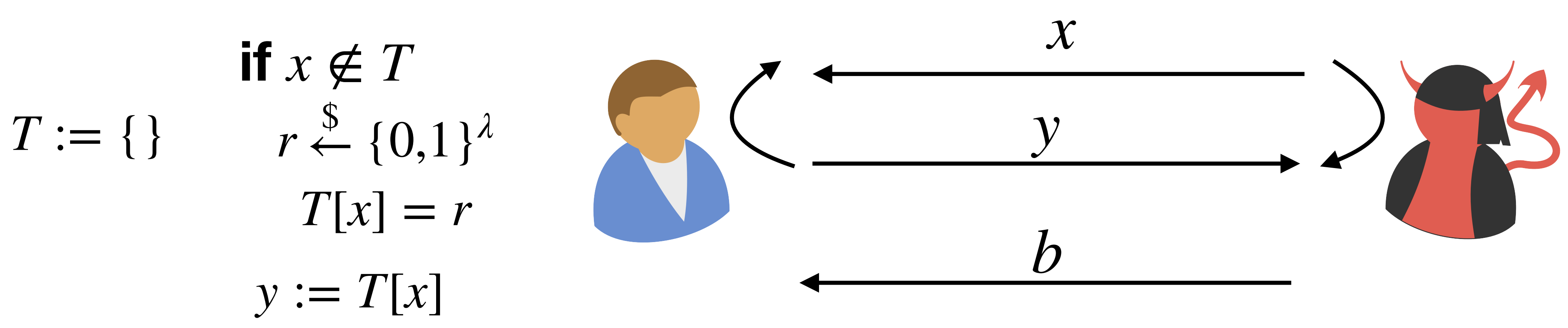
$$\left|\Pr[W_0] - \Pr[W_1]\right| \leq \text{negl}(\lambda)$$

Recap: PRF Security

Game_0



Game_1



Let W_b be the event that $\mathscr{A}$ outputs 0 in Game_b

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

This is a *decision* problem!
Need to be careful in the proof as we relay on the *decisional* security to prove that a *search* problem is hard

Proof of Security

Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

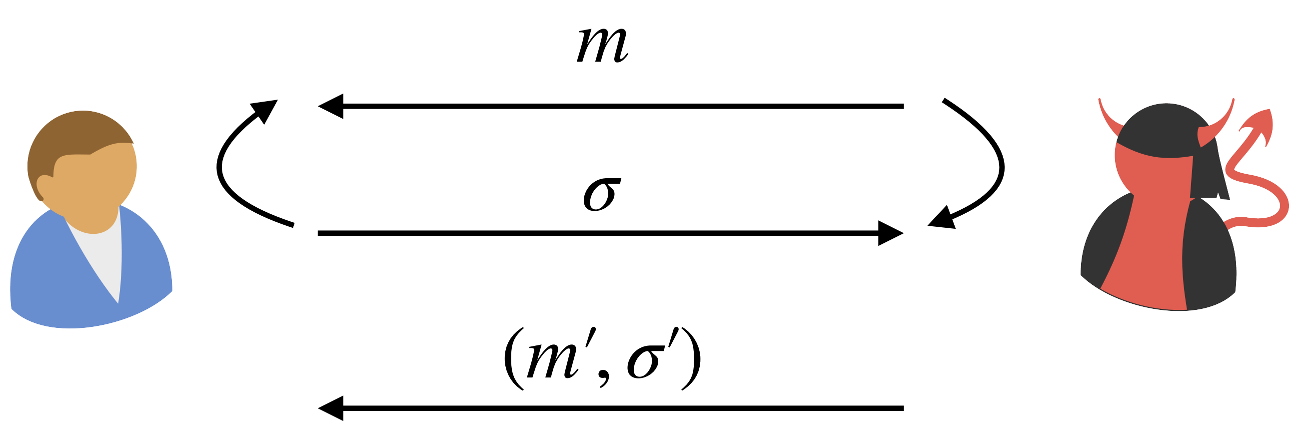
Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

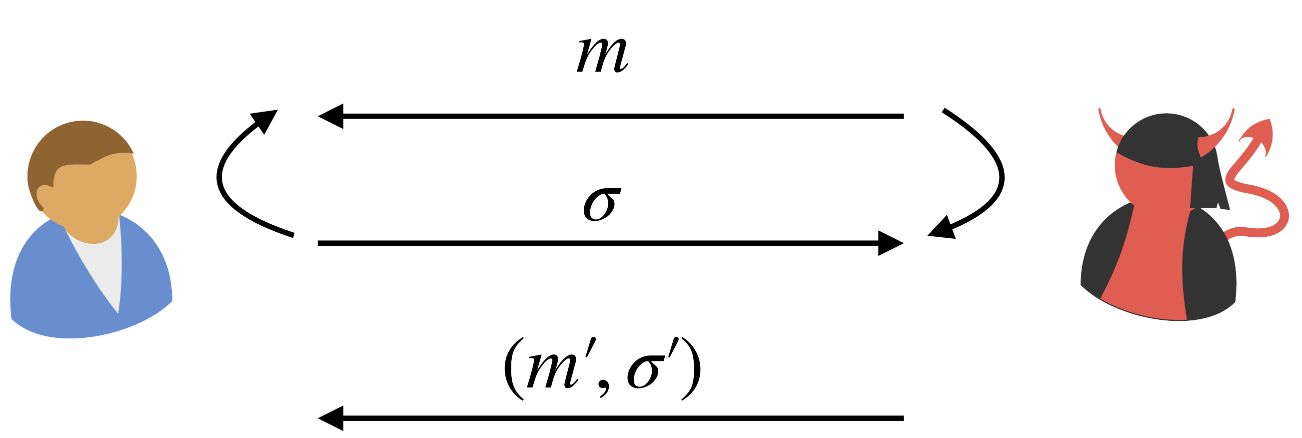
$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



Claim: If F is a secure family of PRFs, then PRF-MAC is a secure MAC

Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

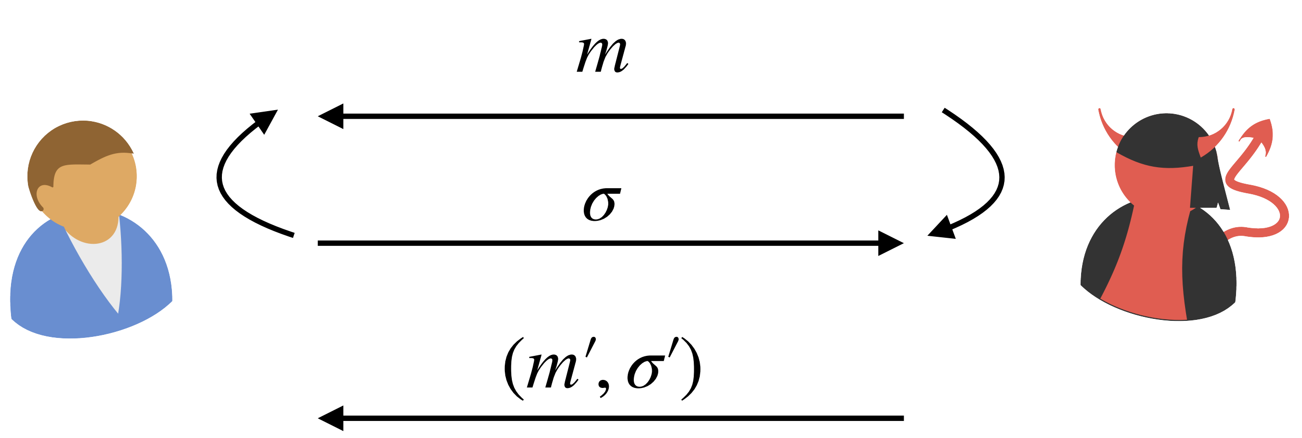
Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



Claim: If F is a secure family of PRFs, then **Pr**$[\mathscr{A}$ wins in $H_0] \leq$ **negl**(λ)

Proof of Security

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

$$\text{Tag}(k, m) : \sigma := F_k(m)$$

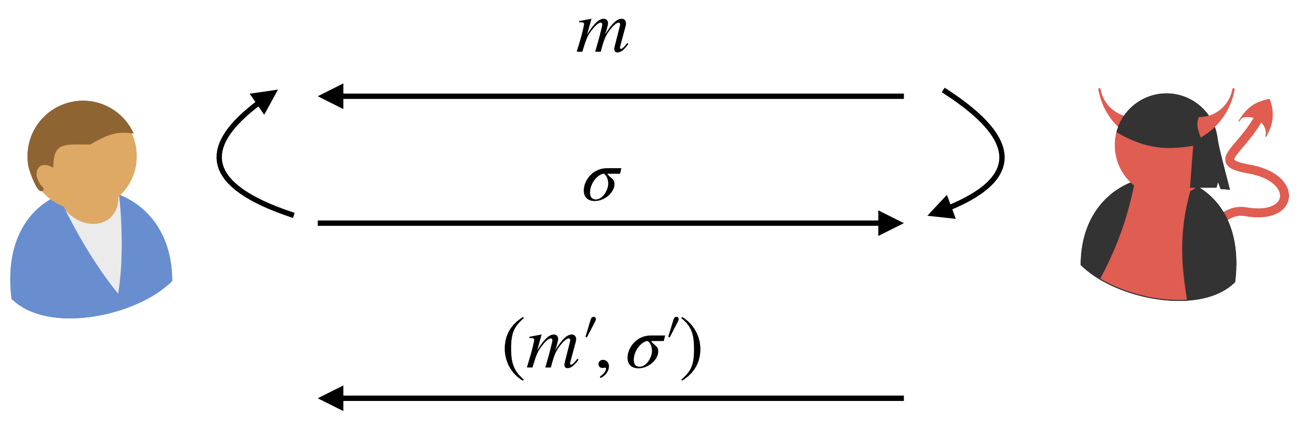
$$\text{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



Proof of Security

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

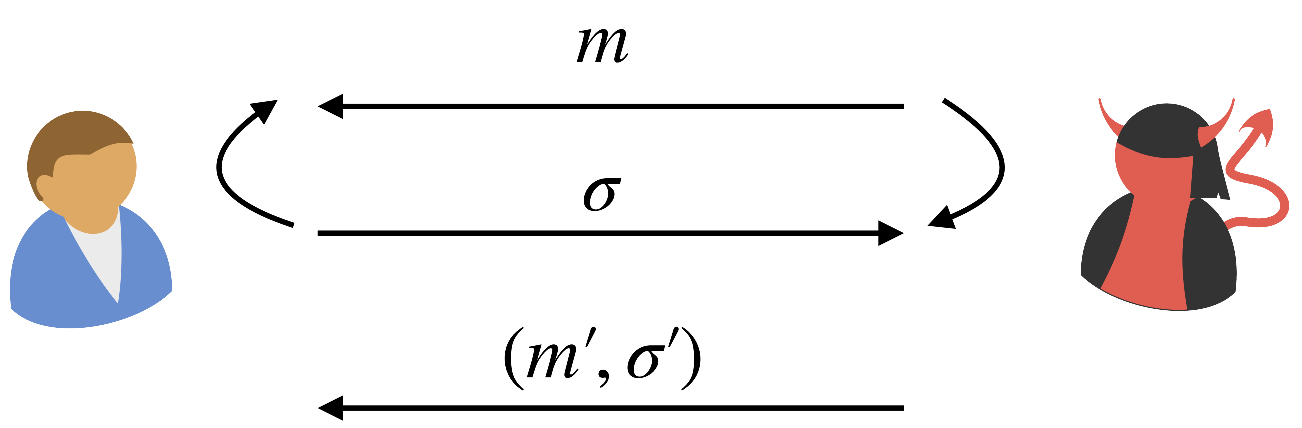
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

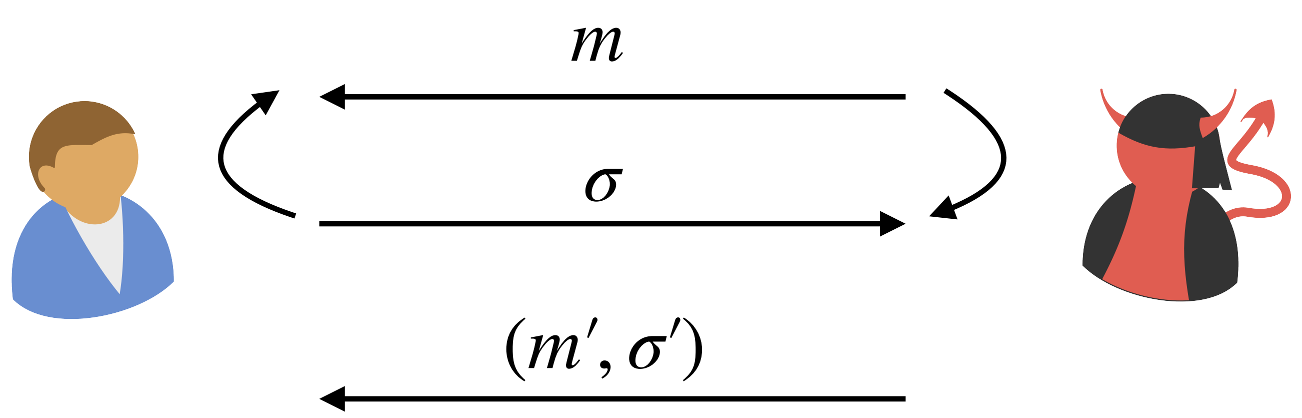
$\sigma := F_k(m)$



Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Proof of Security

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

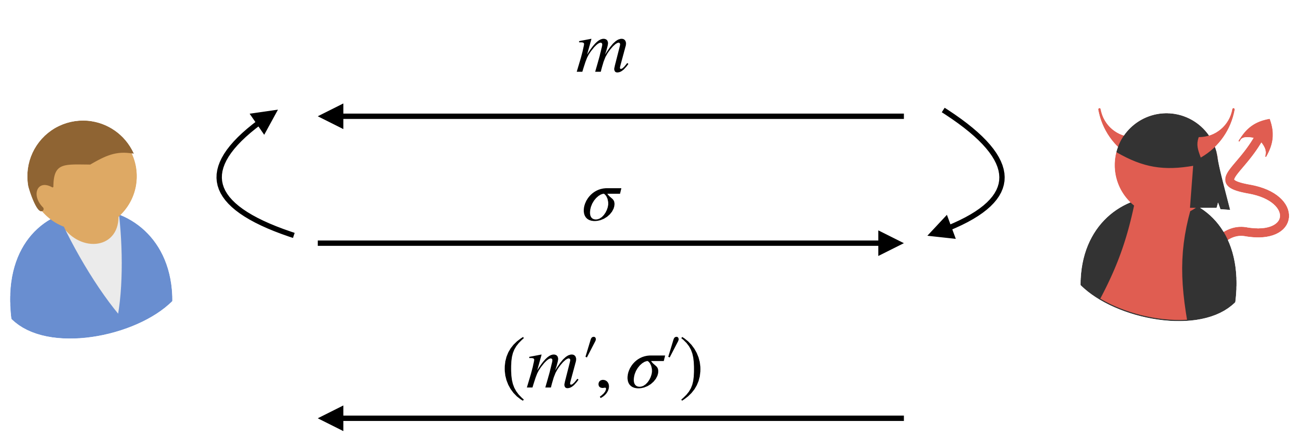
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



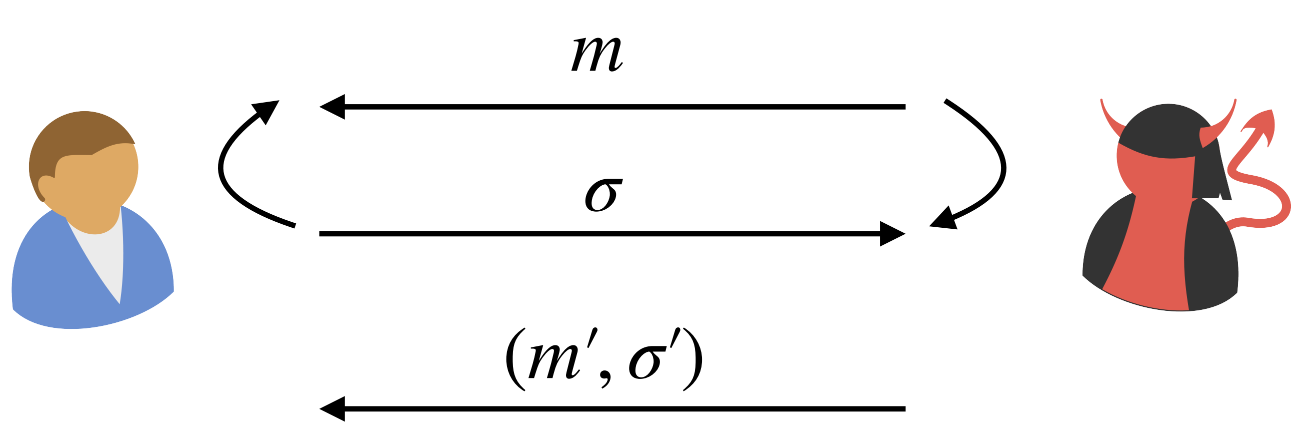
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

$$\Big| \text{Pr}[\mathscr{A} \text{ wins in } H_0] - \text{Pr}[\mathscr{A} \text{ wins in } H_1] \Big| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Proof of Security

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

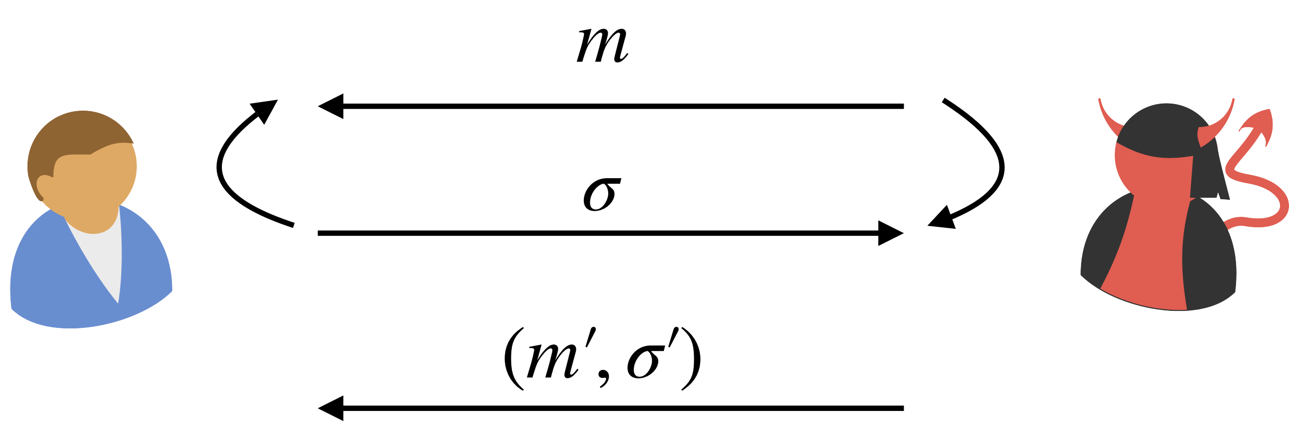
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



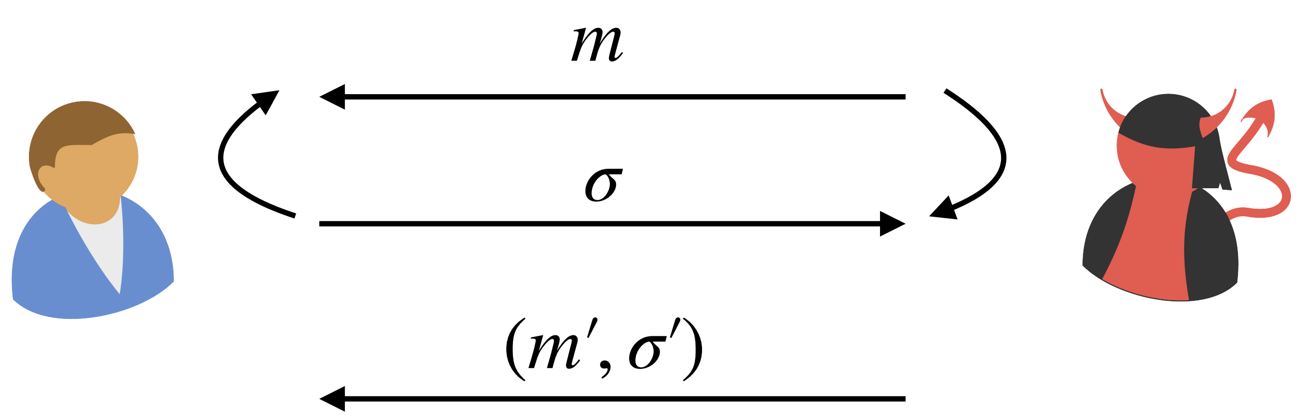
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

$$\left| \text{Pr}[\mathscr{A} \text{ wins in } H_0] - \text{Pr}[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Why? PRF Security!

Proof of Security

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

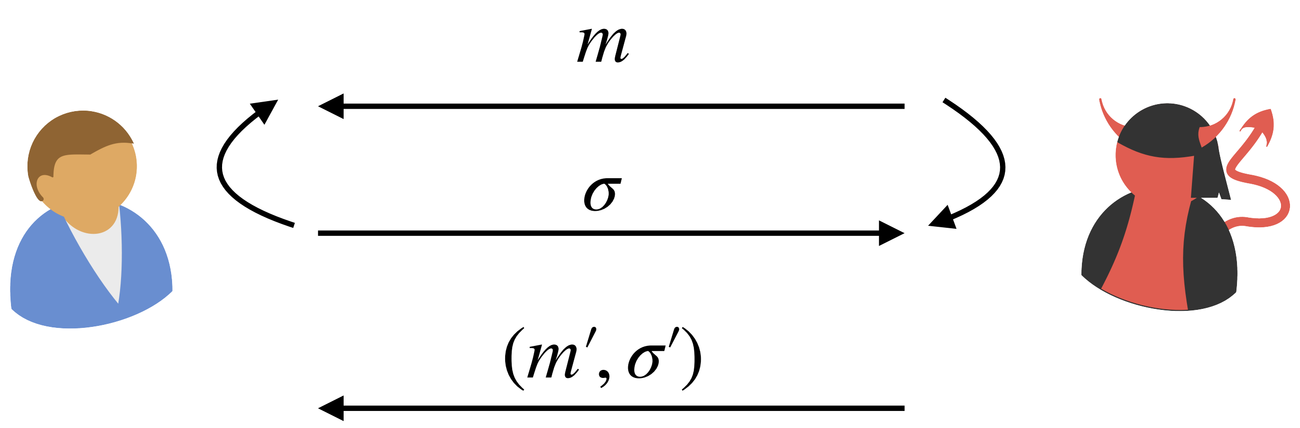
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



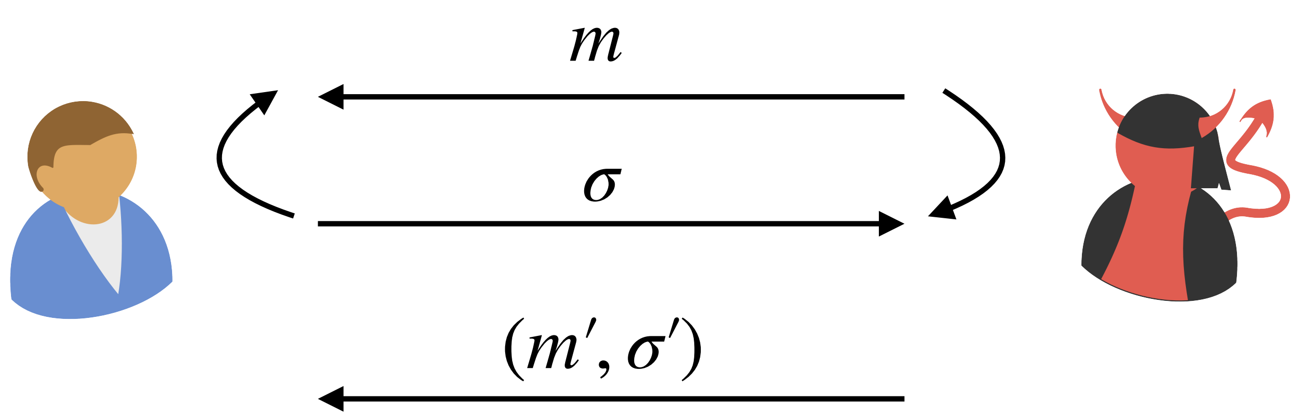
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Proof of Security

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

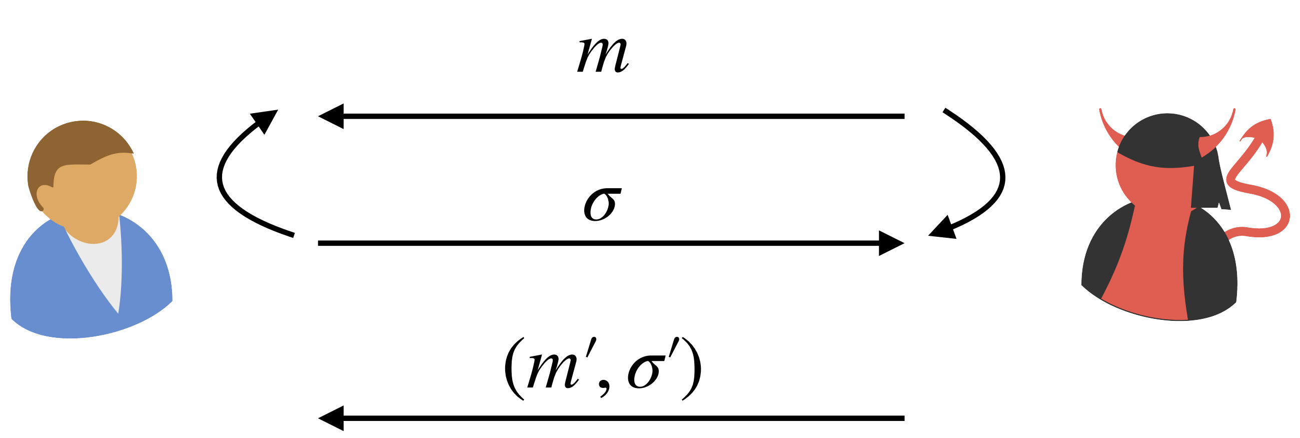
$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



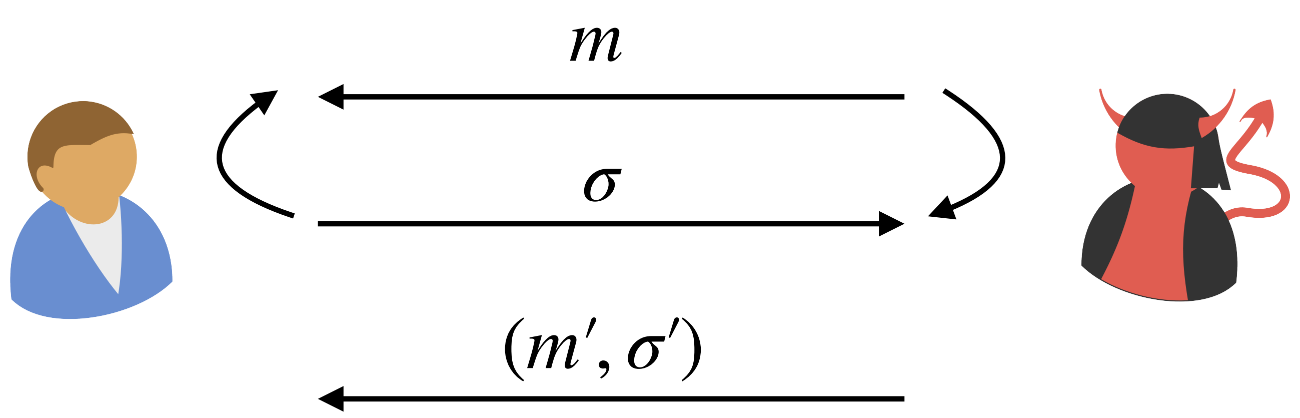
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

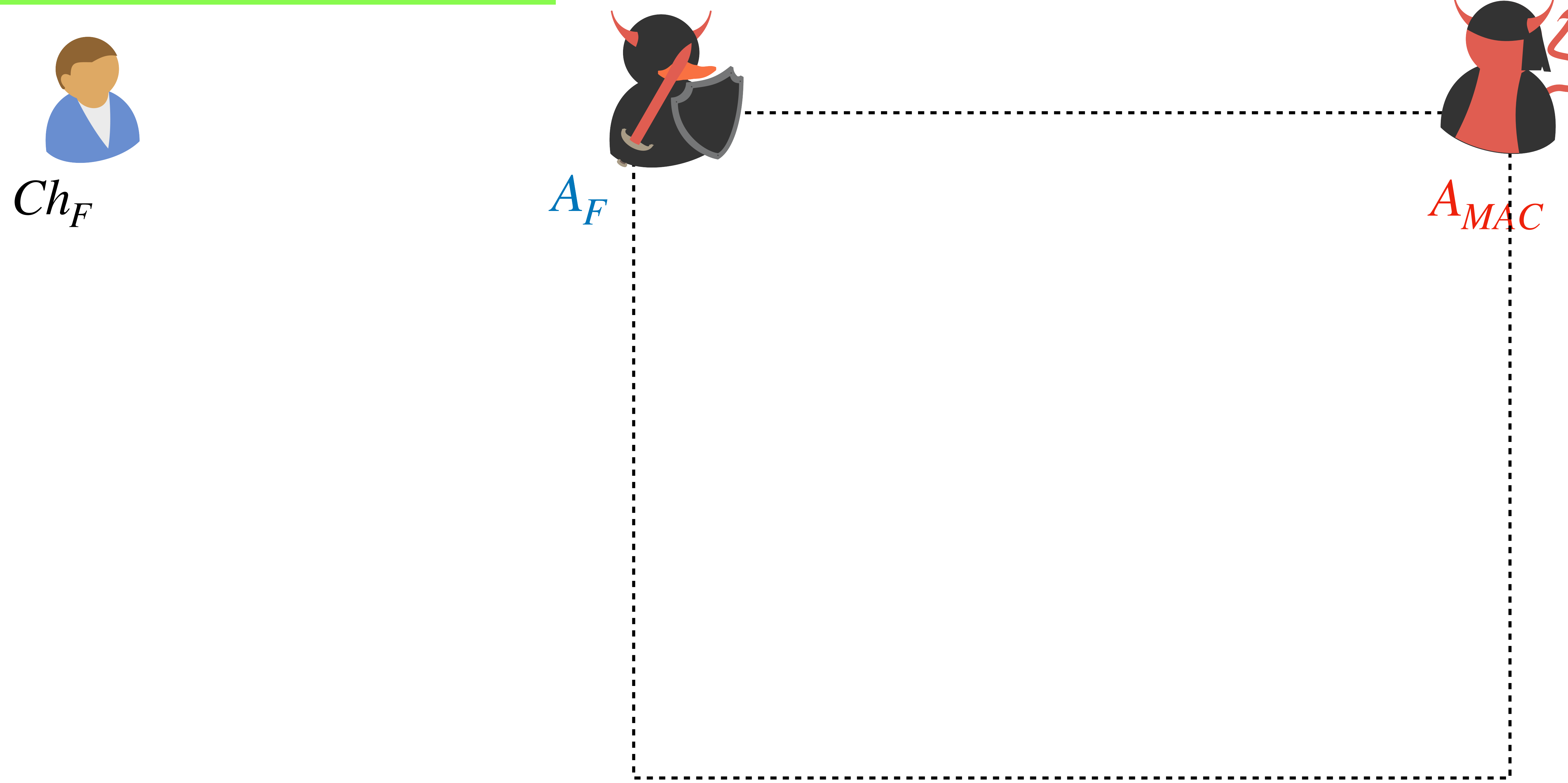
$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \mathsf{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

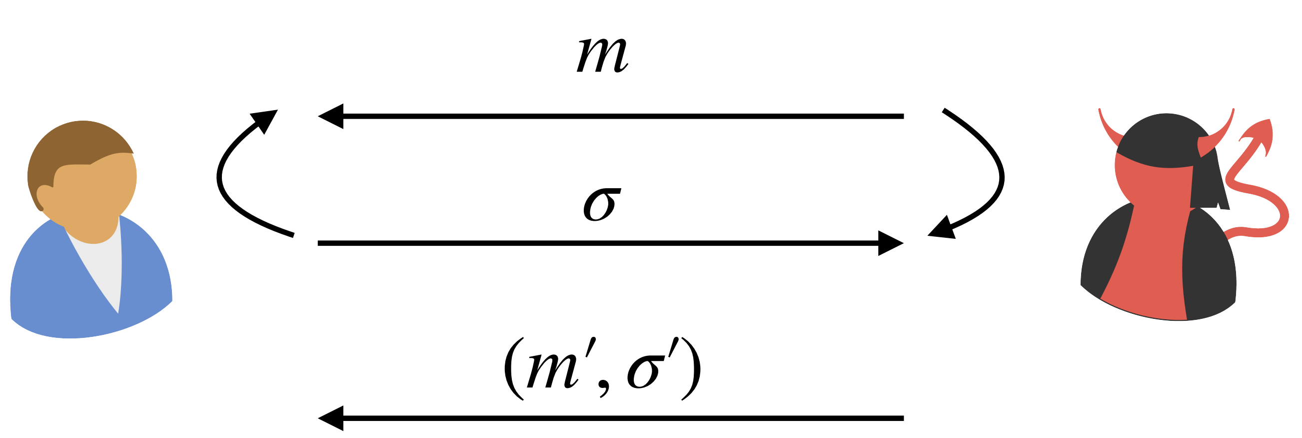
Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



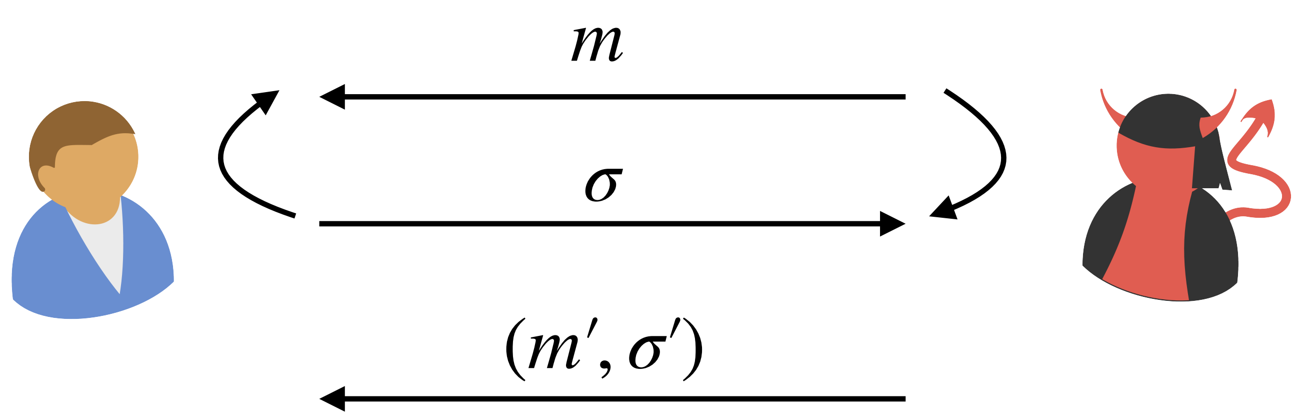
Claim:

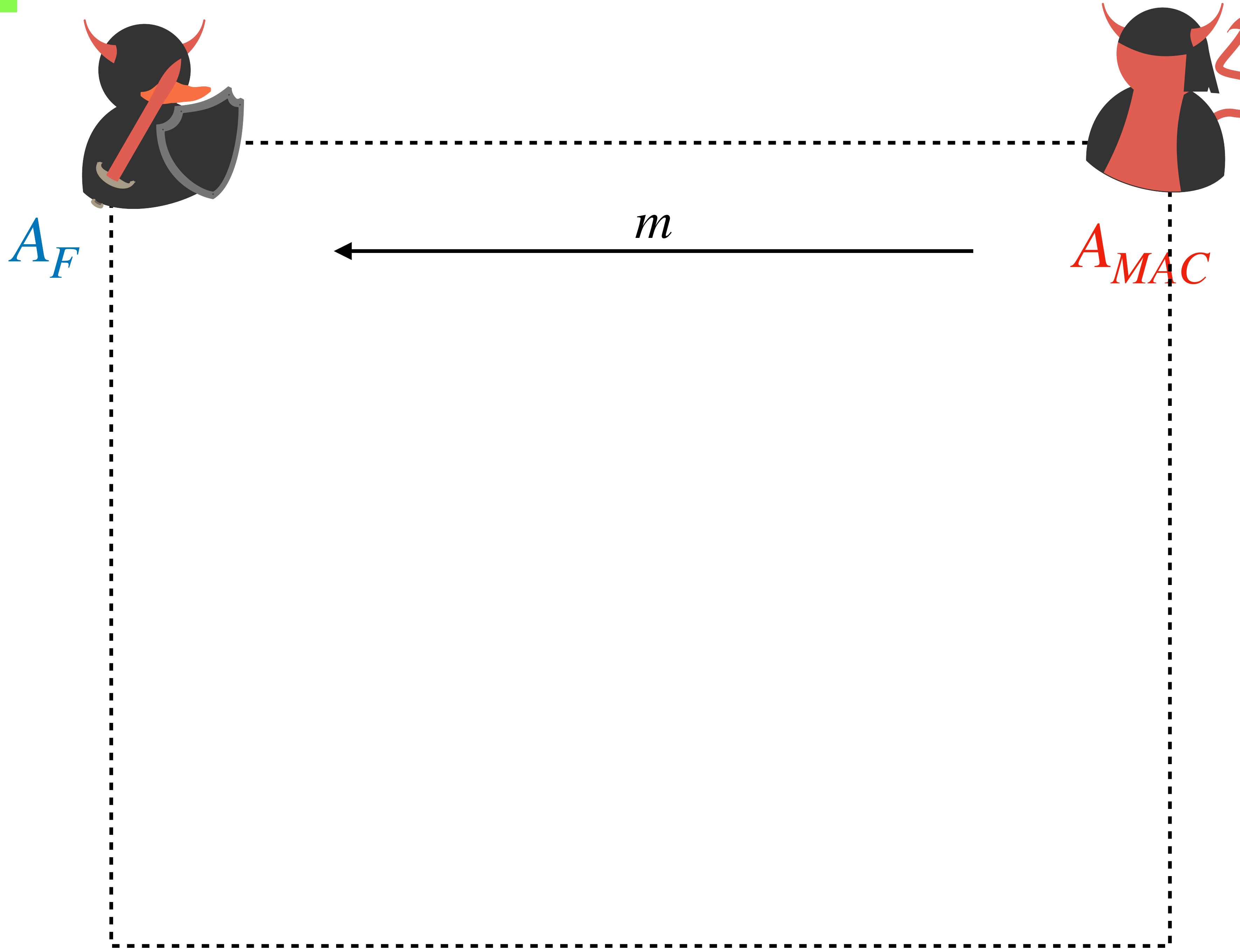
$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$





Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

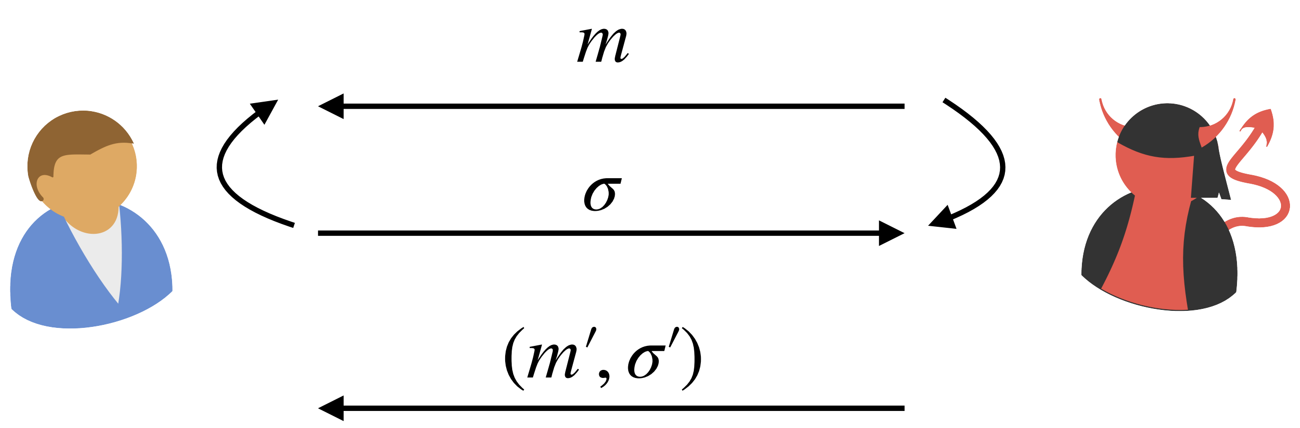
Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



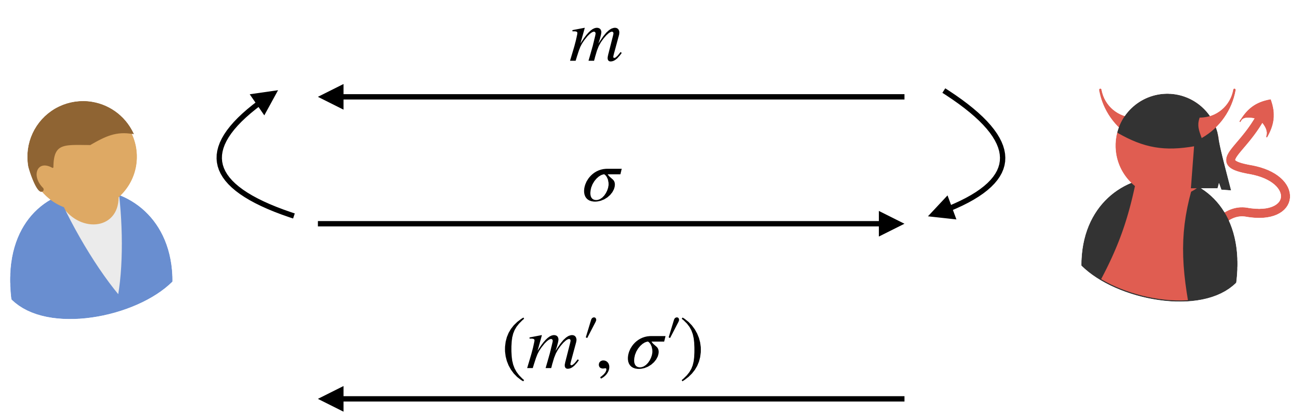
Claim:

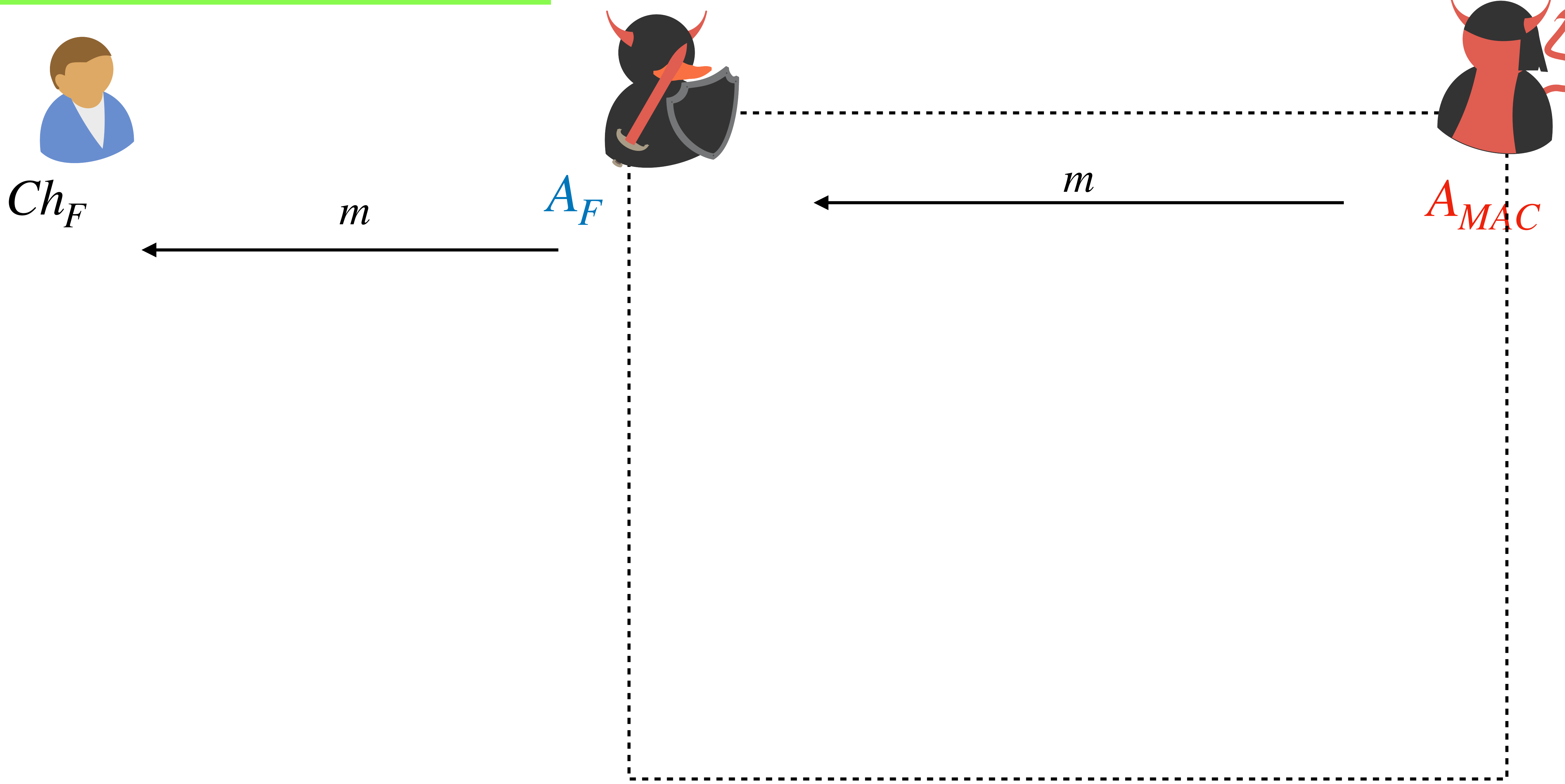
$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \mathsf{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$





Proof of Security

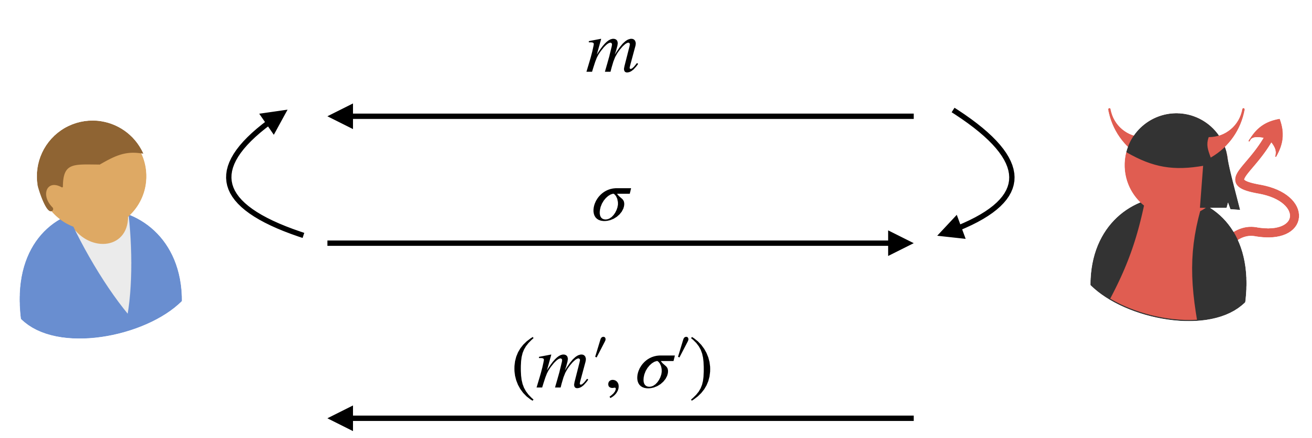
KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$
$\sigma := F_k(m)$

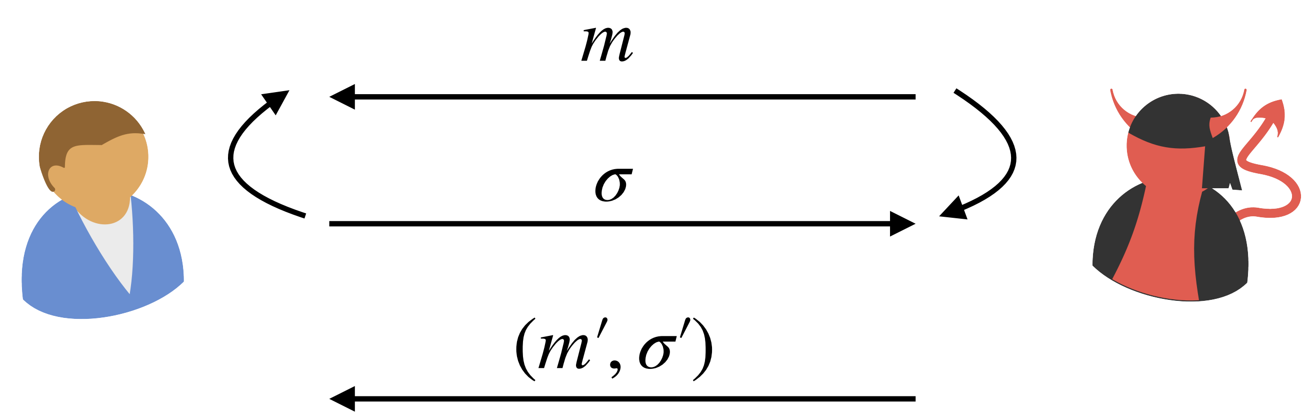


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

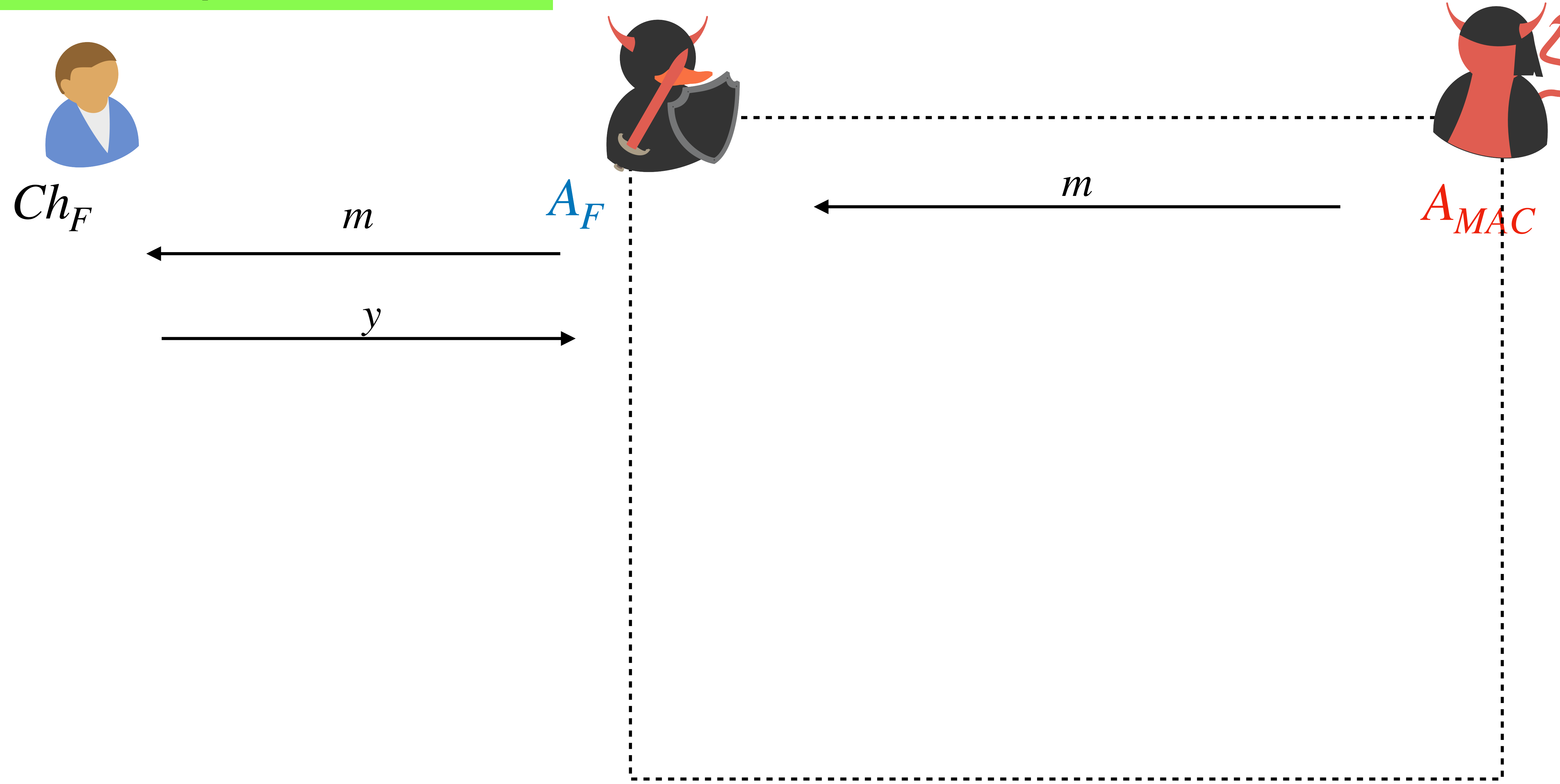
Claim:
$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \mathsf{negl}(\lambda)$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



Proof of Security

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

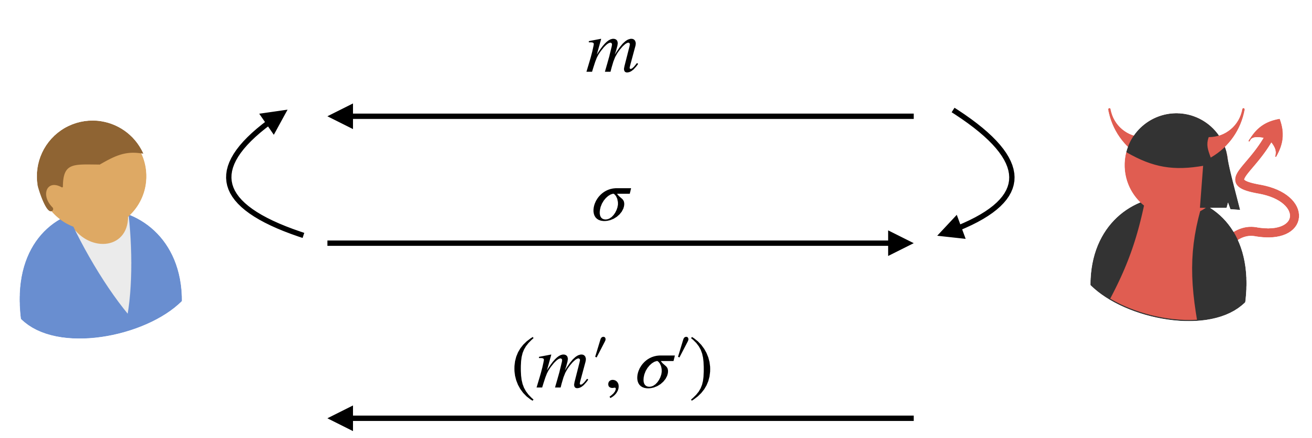
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$
$\sigma := F_k(m)$

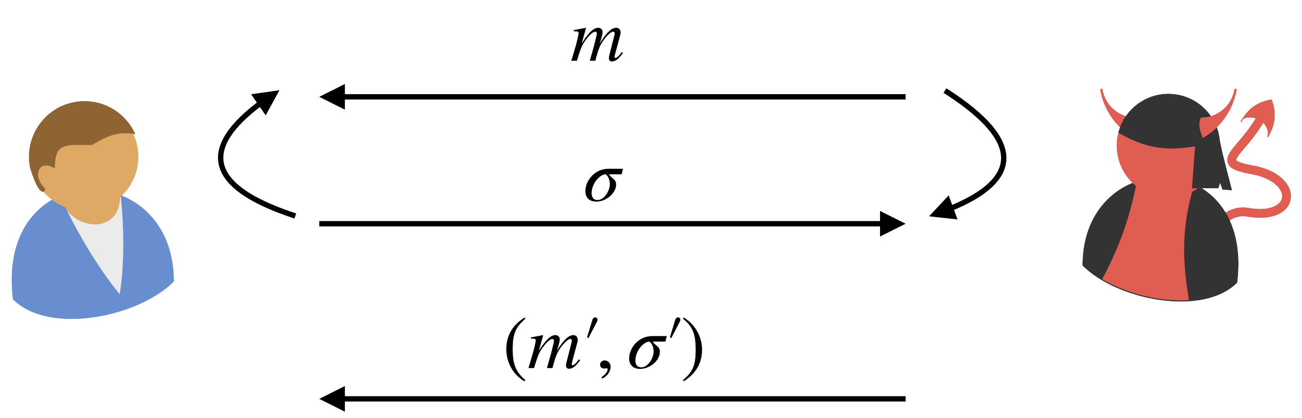


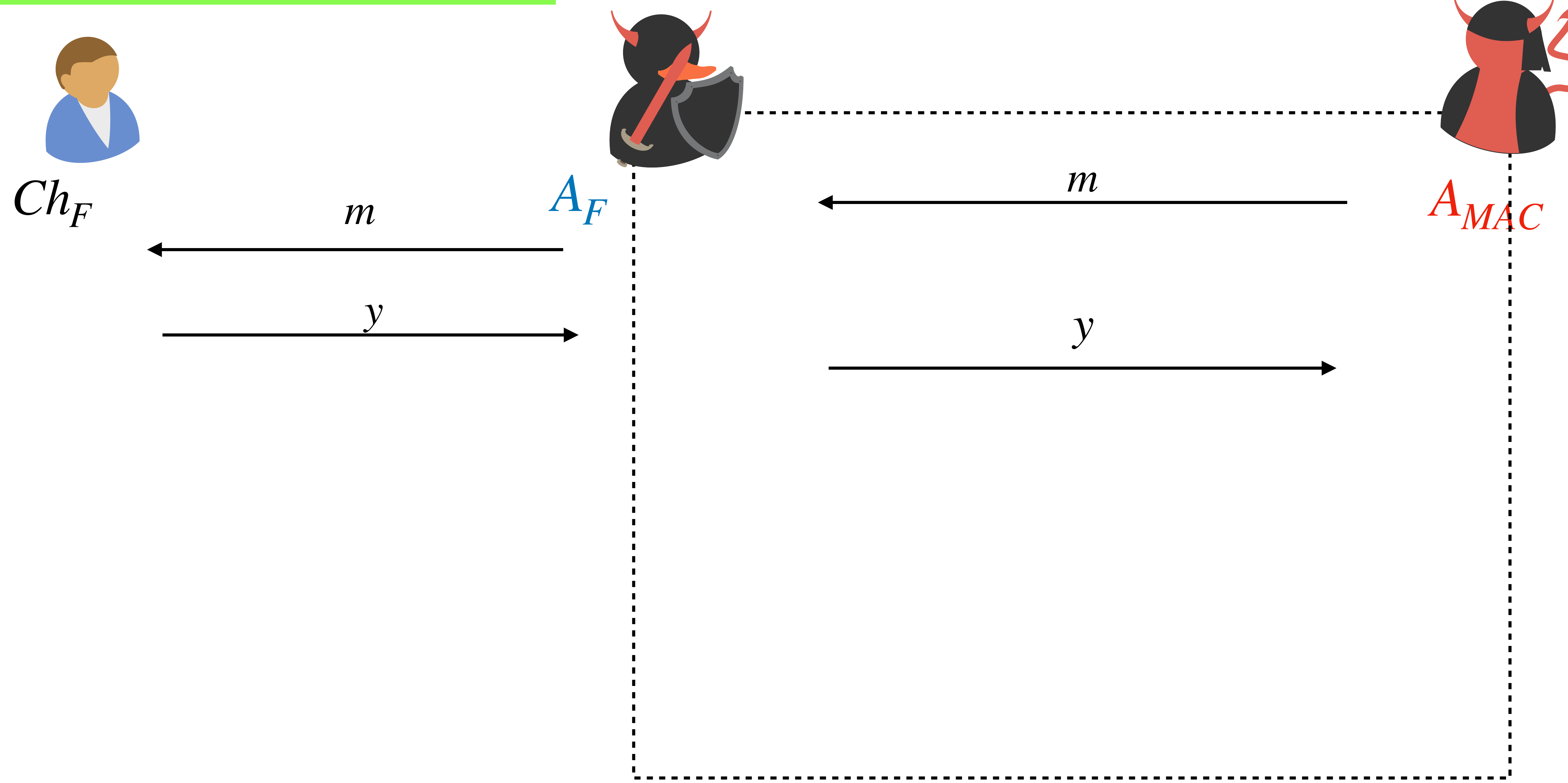
Claim:
$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \text{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$





Proof of Security

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

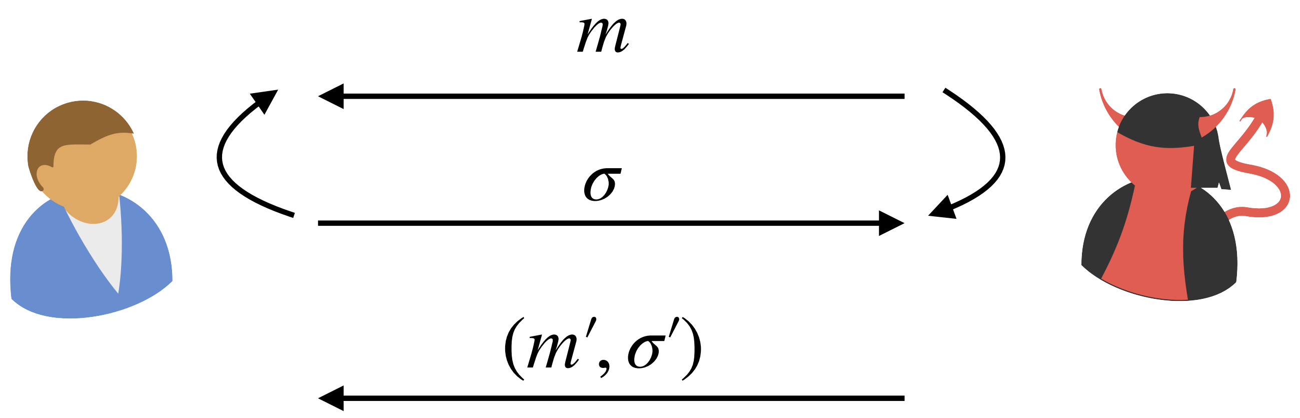
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



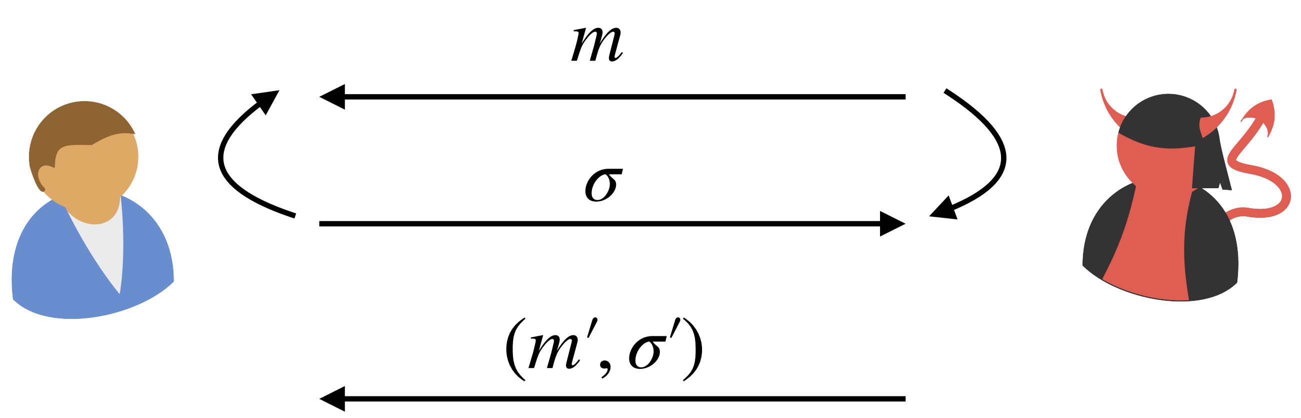
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

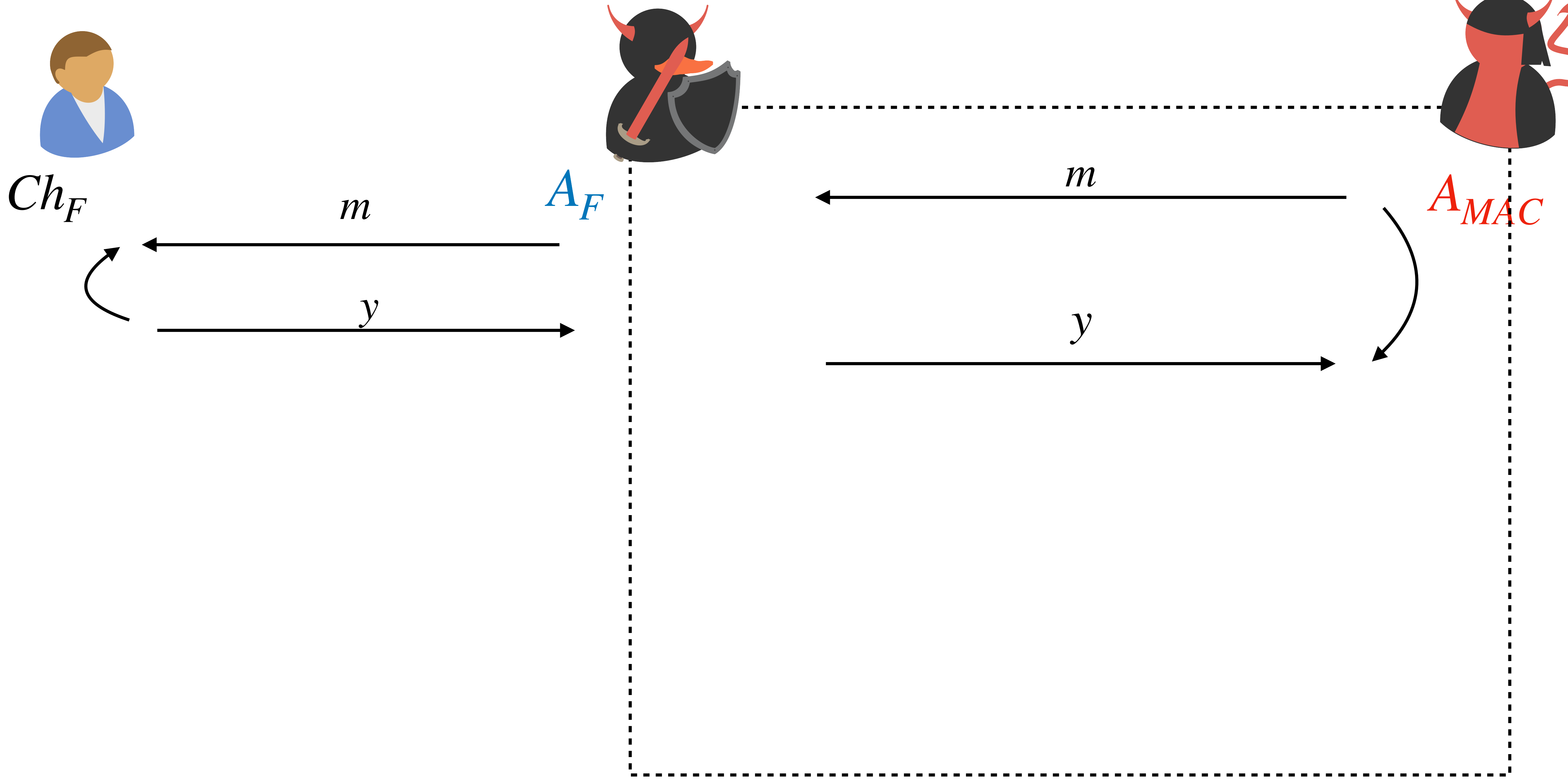
$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \text{negl}(\lambda)$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



Proof of Security

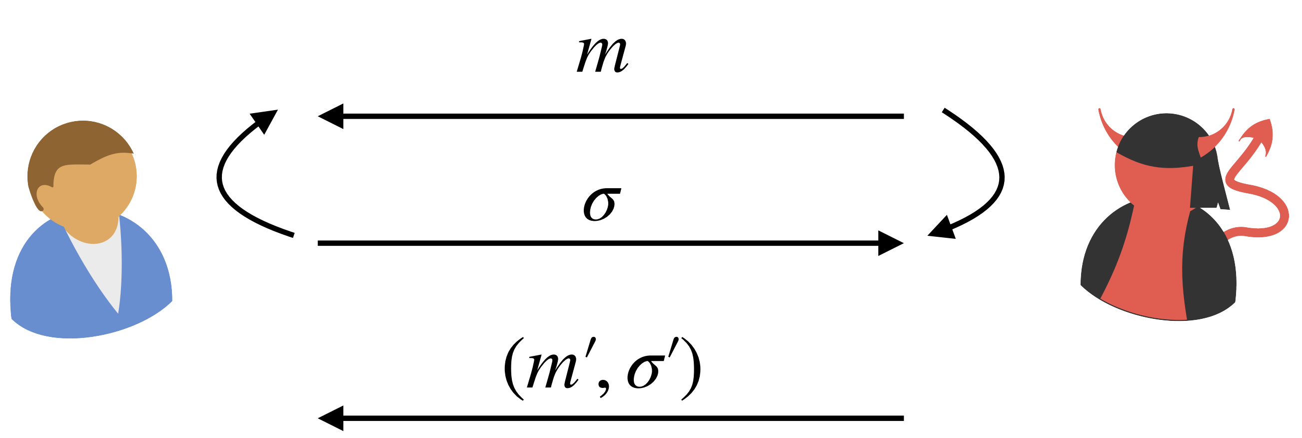
$\text{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\text{Tag}(k, m) : \sigma := F_k(m)$

$\text{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0 $\quad k \xleftarrow{\$} \{0,1\}^\lambda$, $\sigma := F_k(m)$

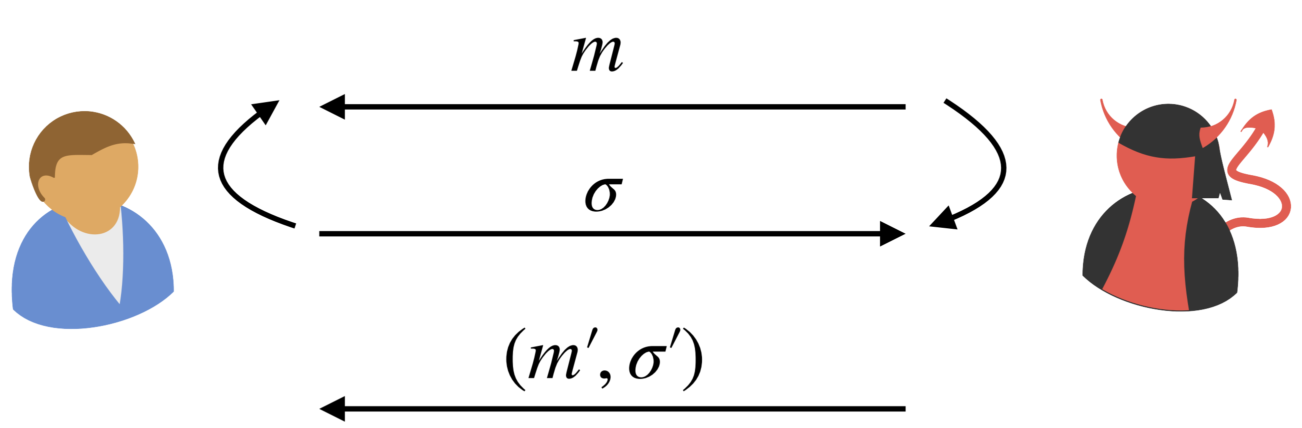


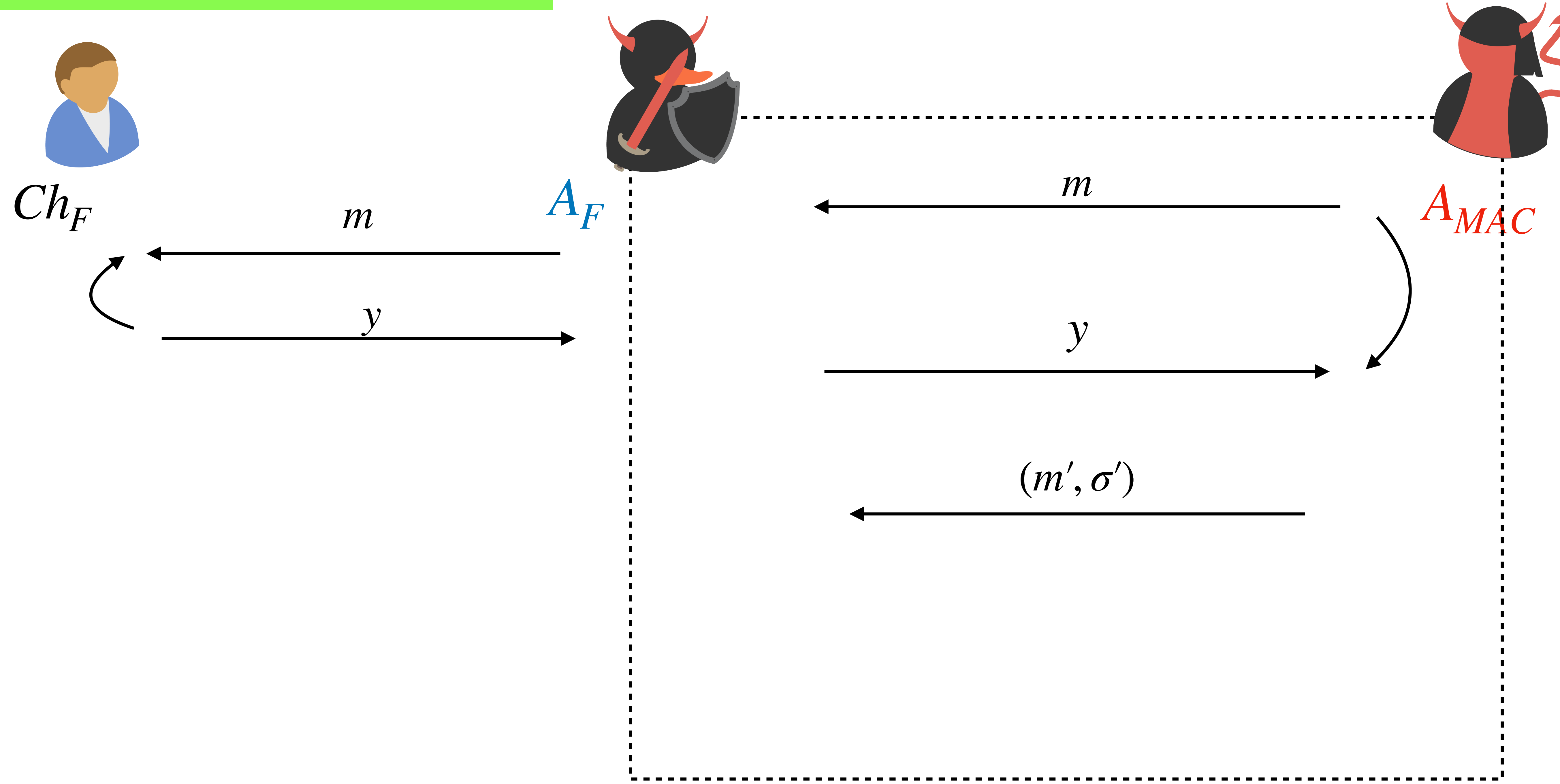
Claim:

$$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \text{negl}(\lambda)$$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1 $\quad \sigma := T[m]$





Proof of Security

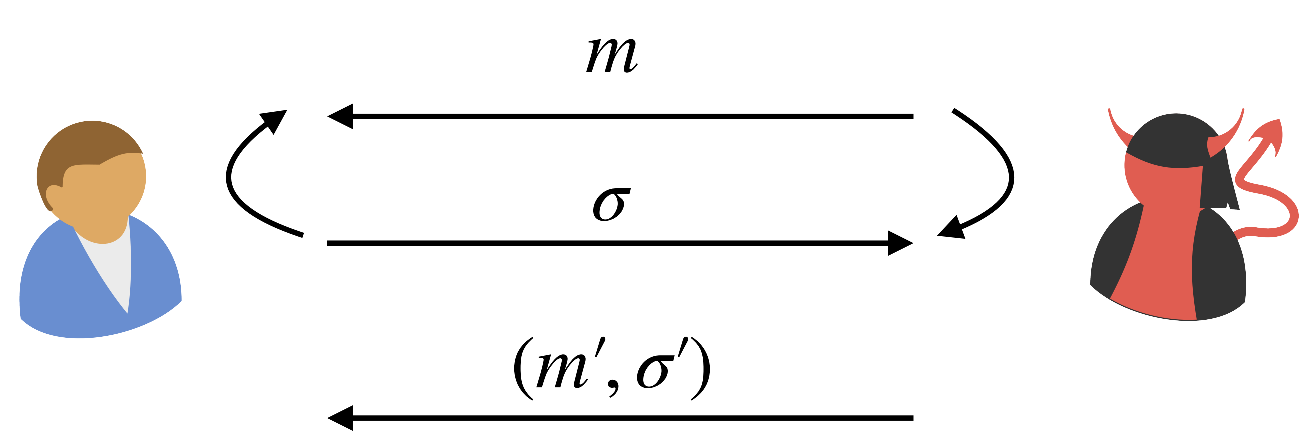
KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

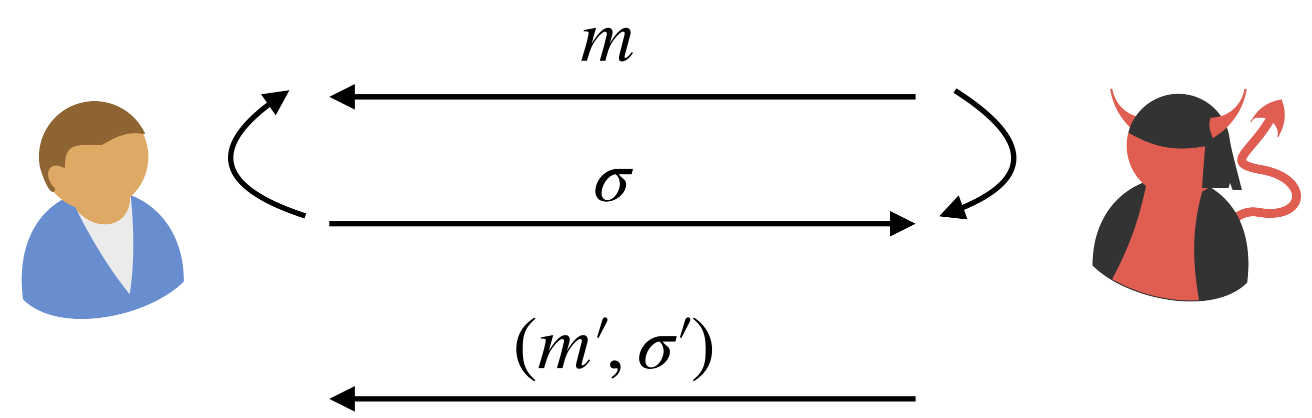
H_0 $k \xleftarrow{\$} \{0,1\}^\lambda$ $\sigma := F_k(m)$

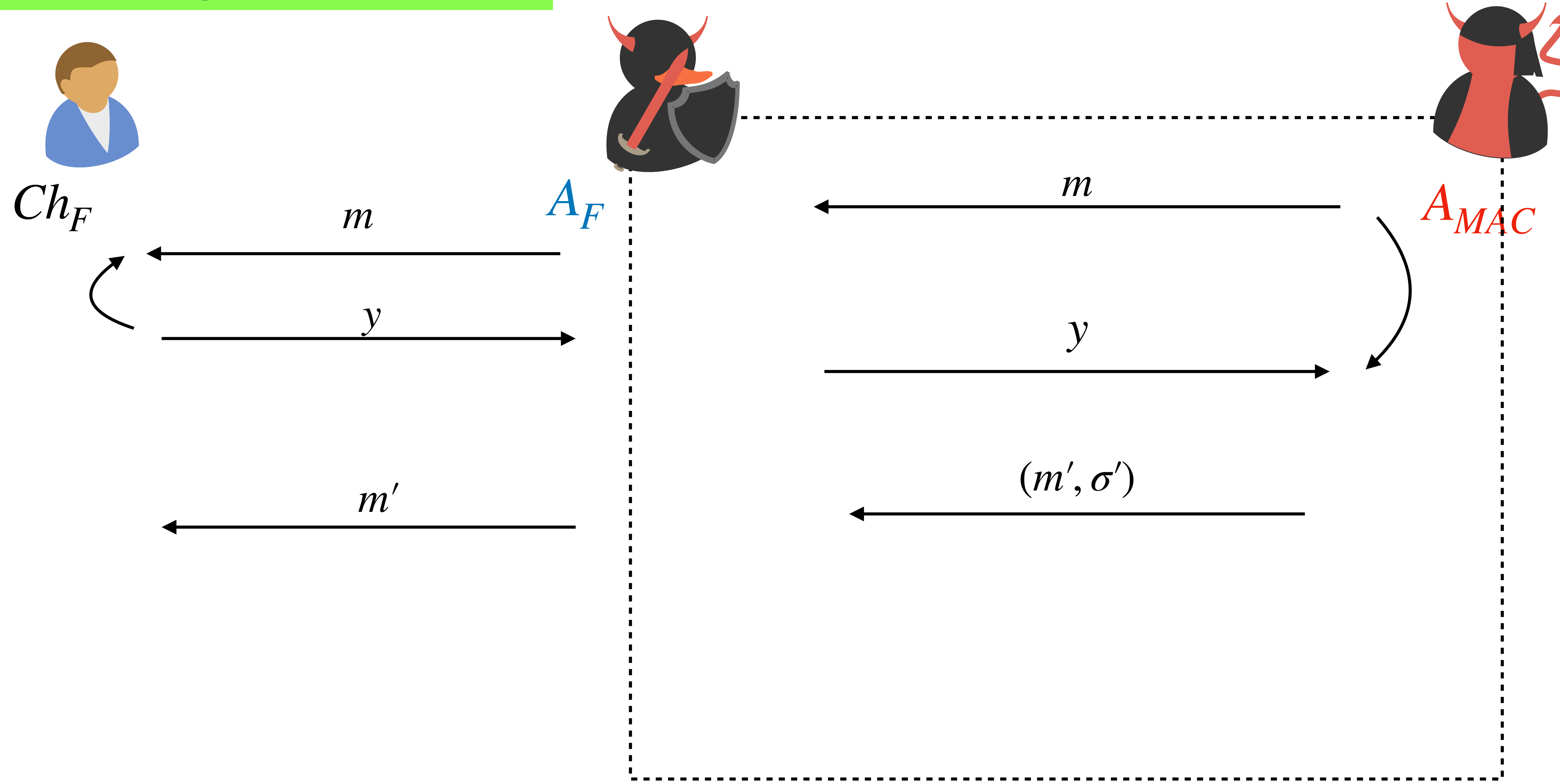


Claim:
$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \mathsf{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1 $\sigma := T[m]$





Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

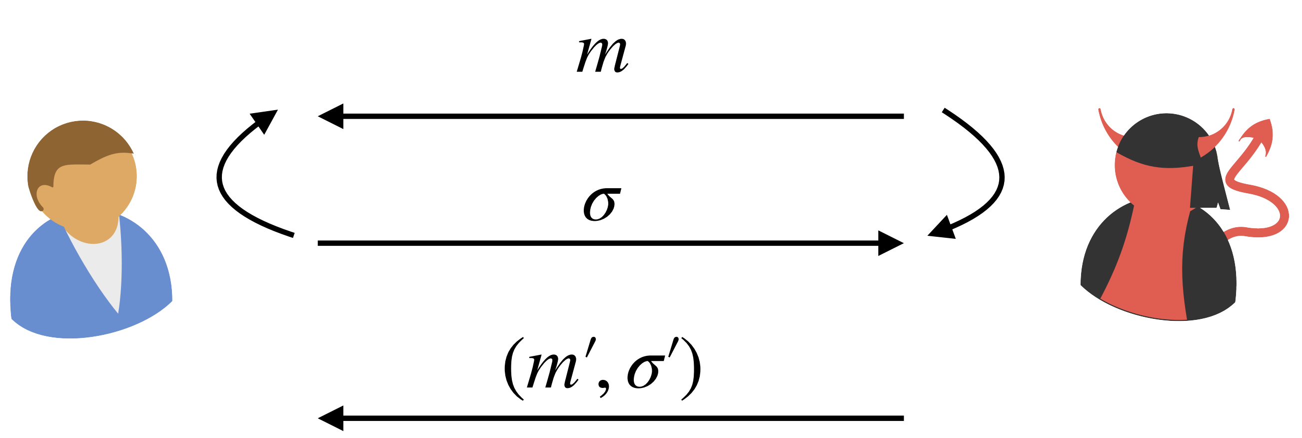
Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



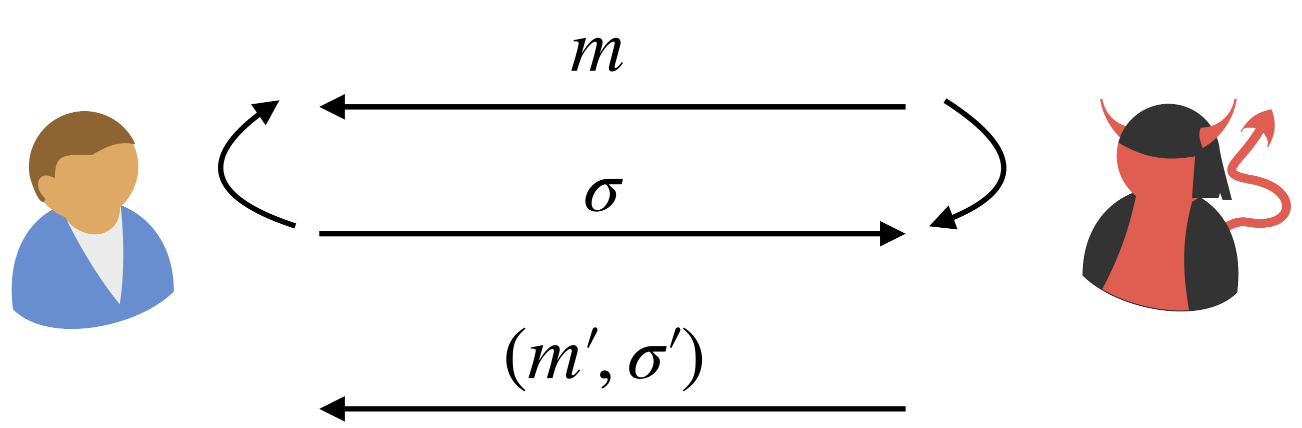
Claim:

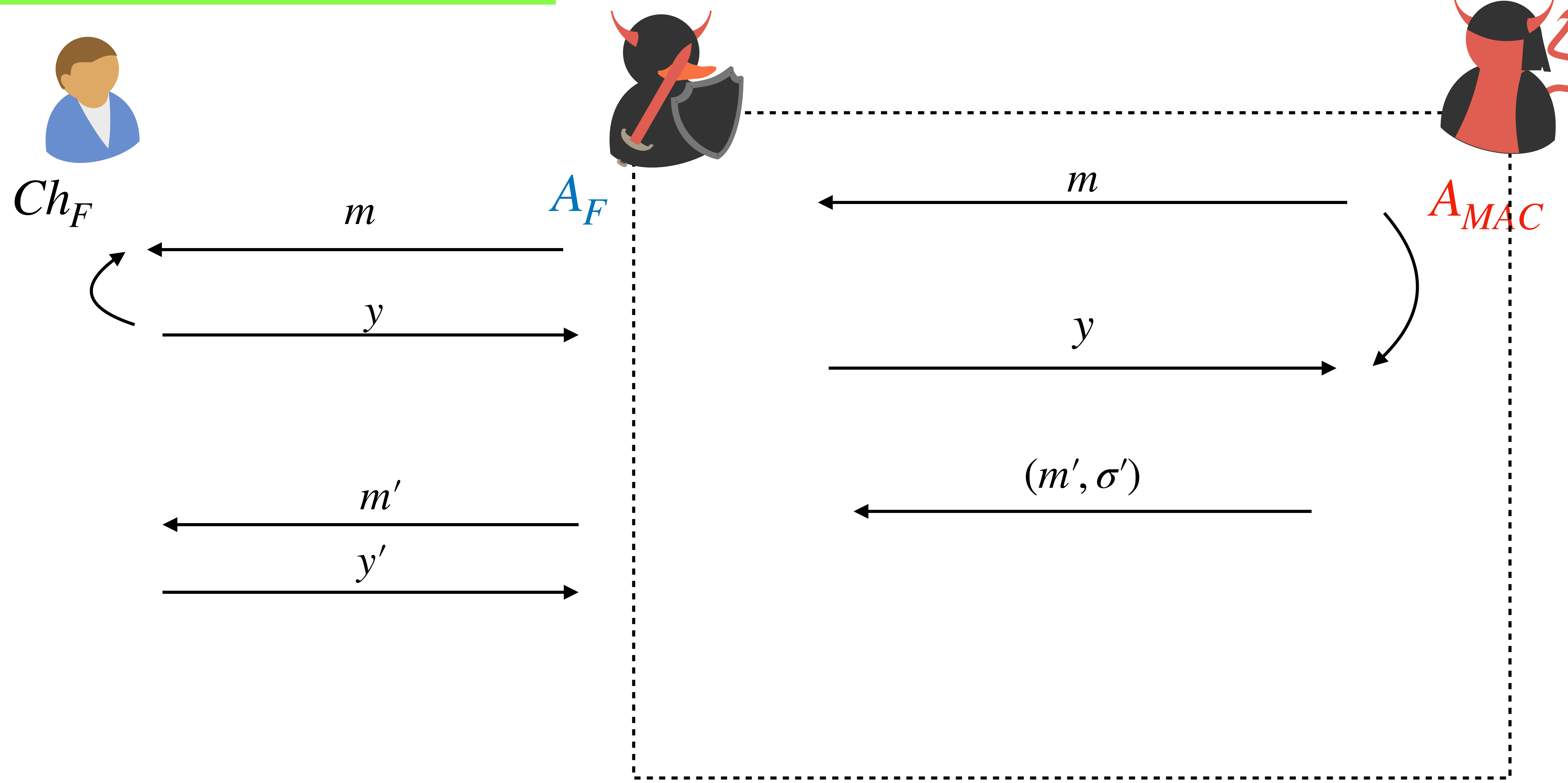
$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \text{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$



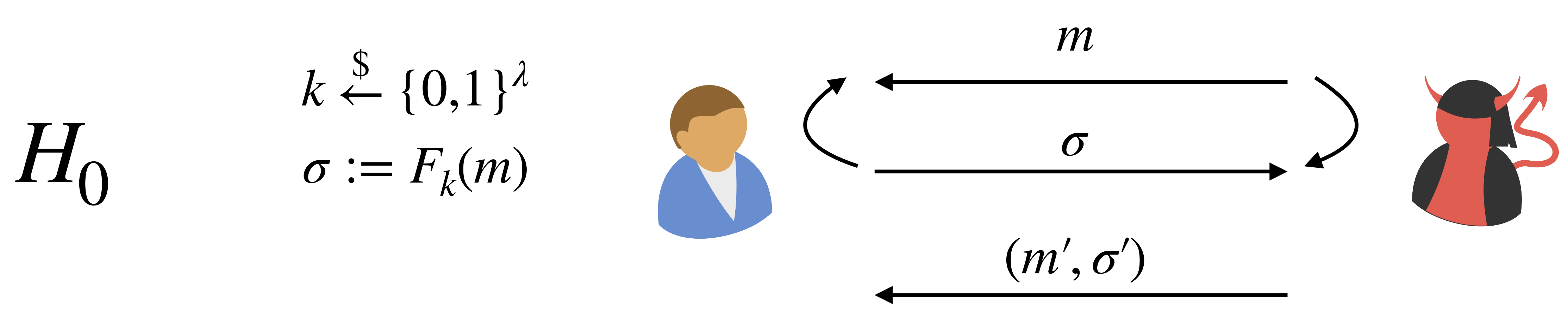


Proof of Security

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

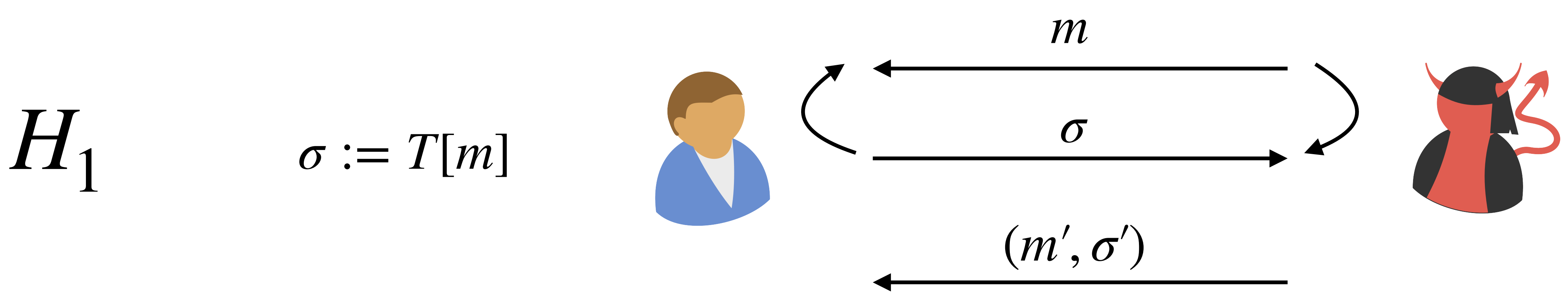
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

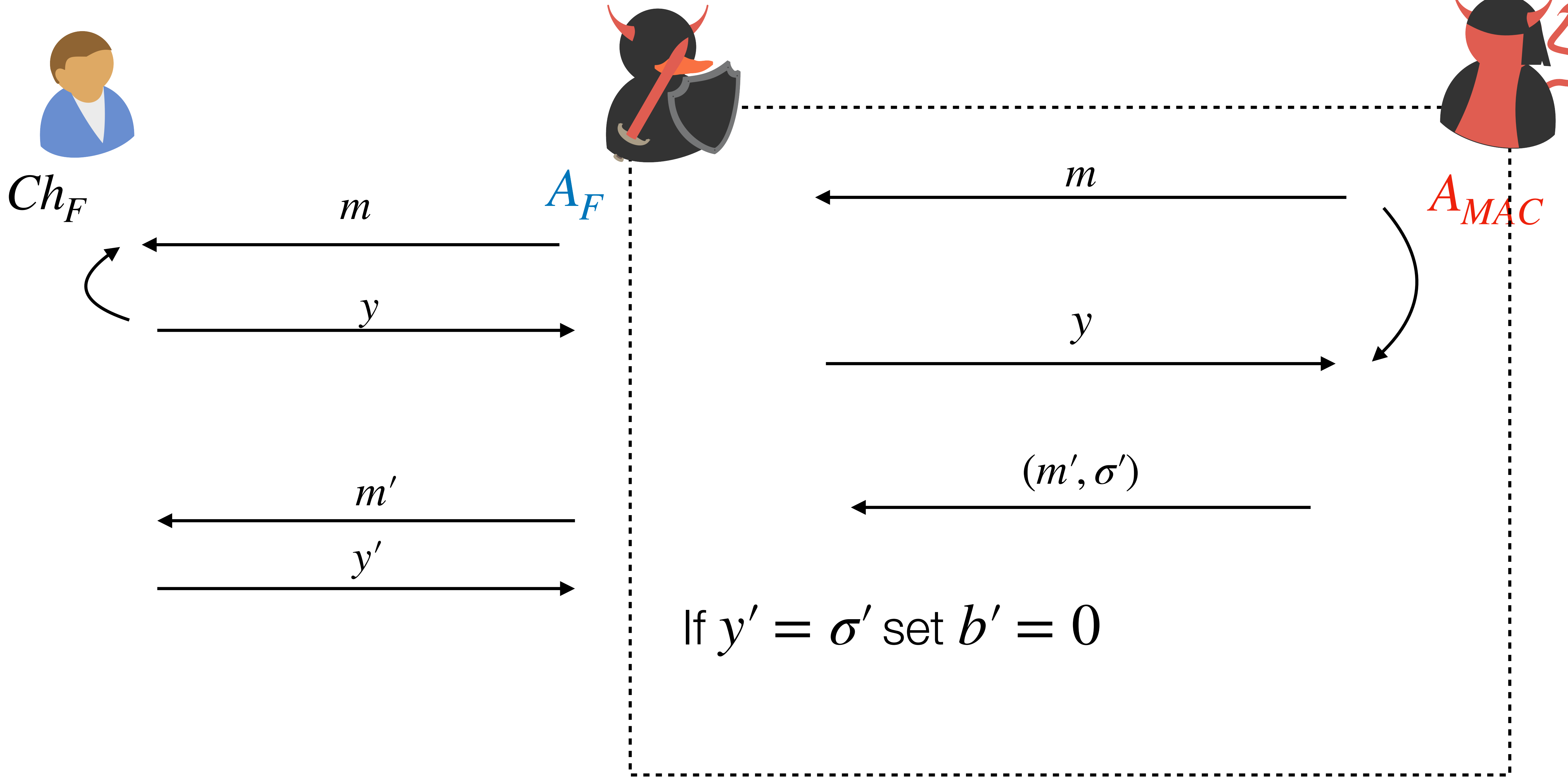


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:
$$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \mathsf{negl}(\lambda)$$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



Proof of Security

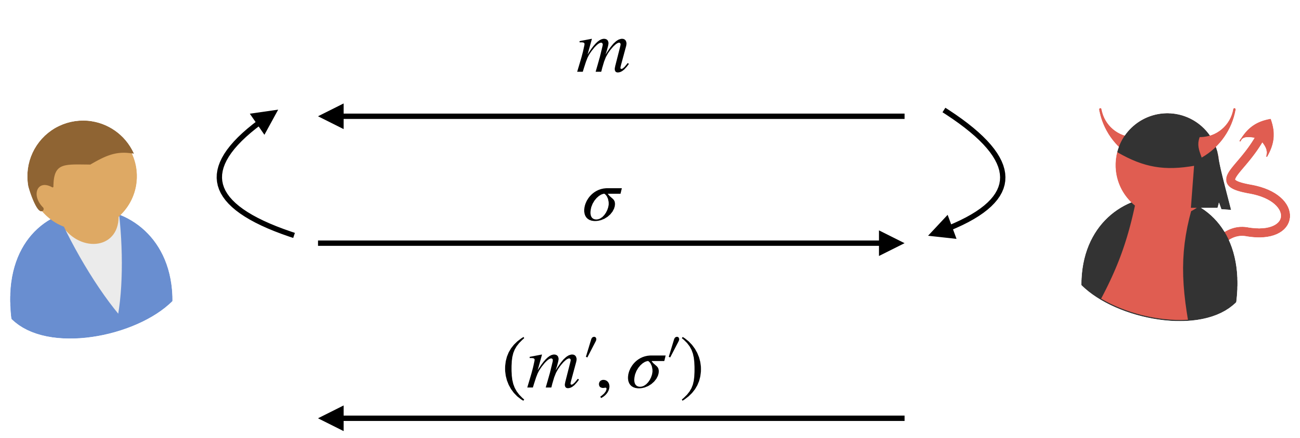
KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0 $\quad k \xleftarrow{\$} \{0,1\}^\lambda$, $\sigma := F_k(m)$

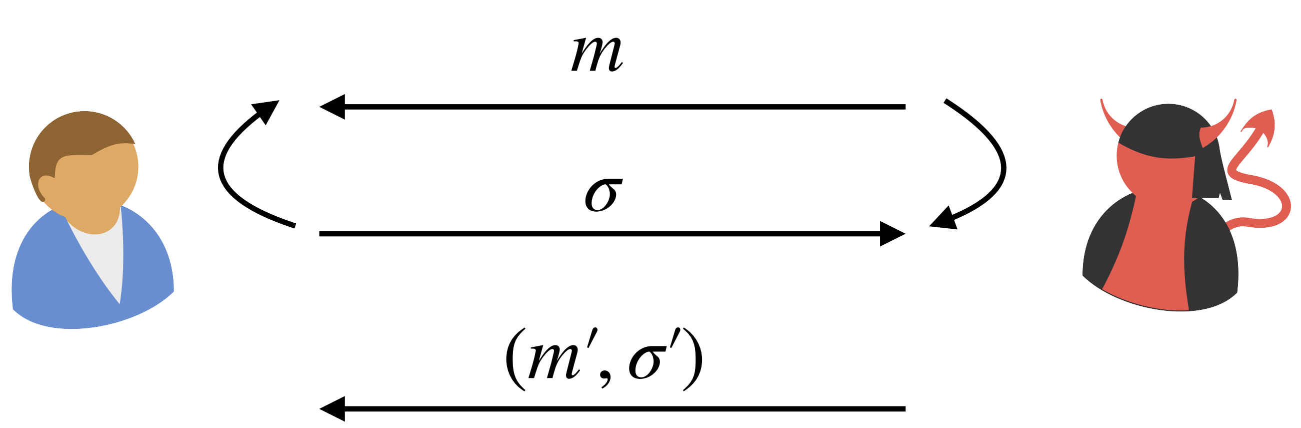


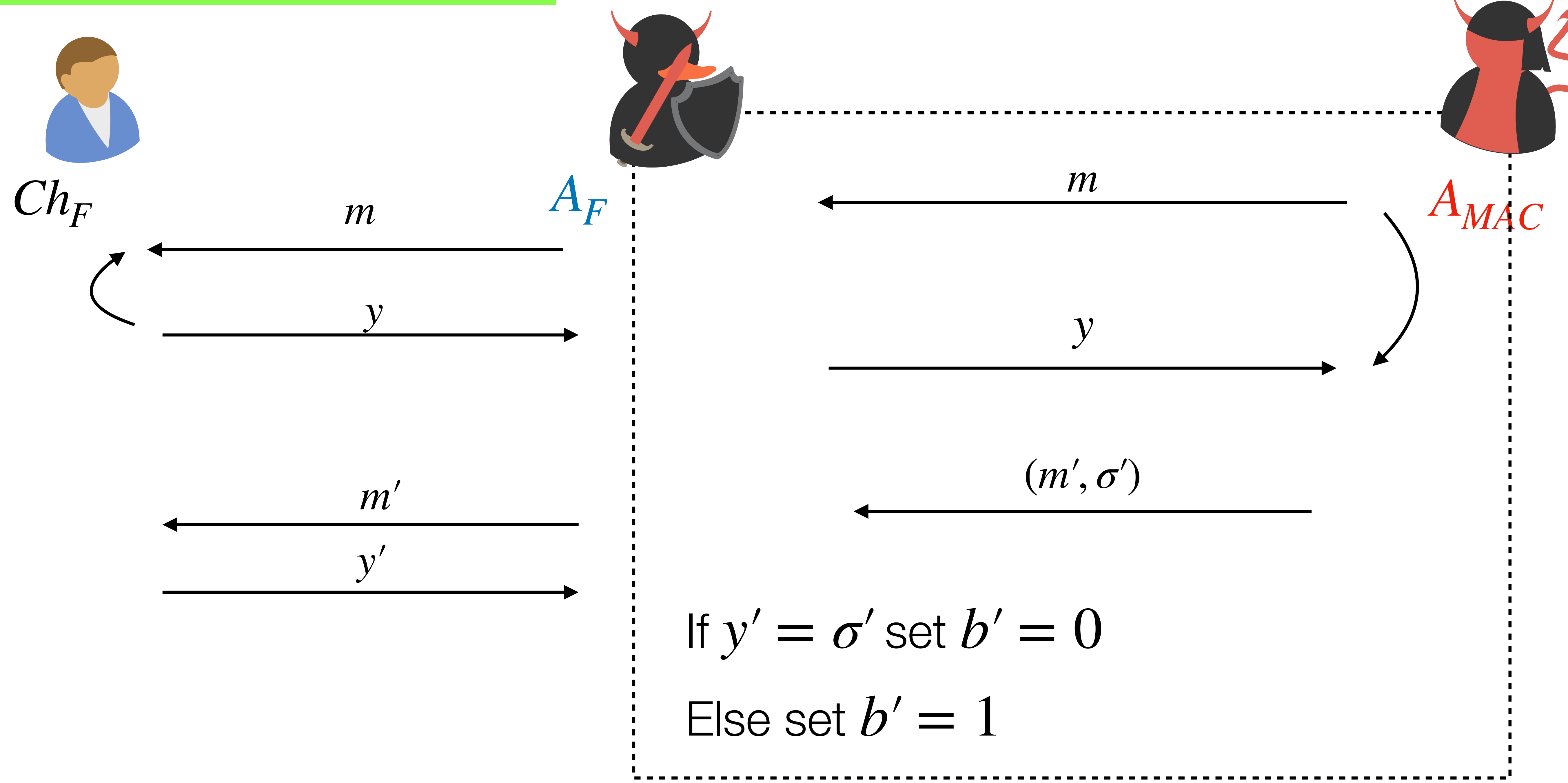
Claim:

$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1 $\quad \sigma := T[m]$





Proof of Security

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

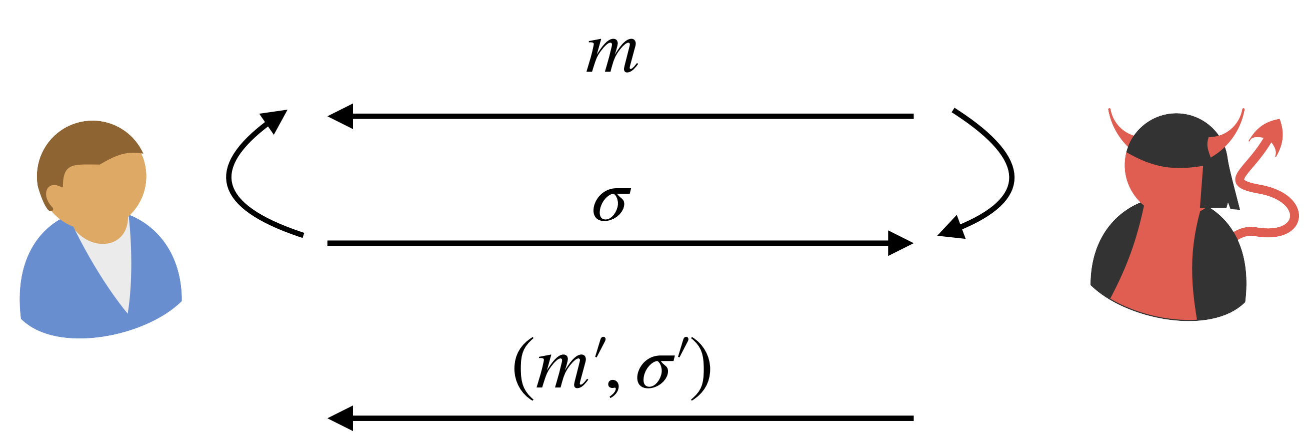
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$
$\sigma := F_k(m)$

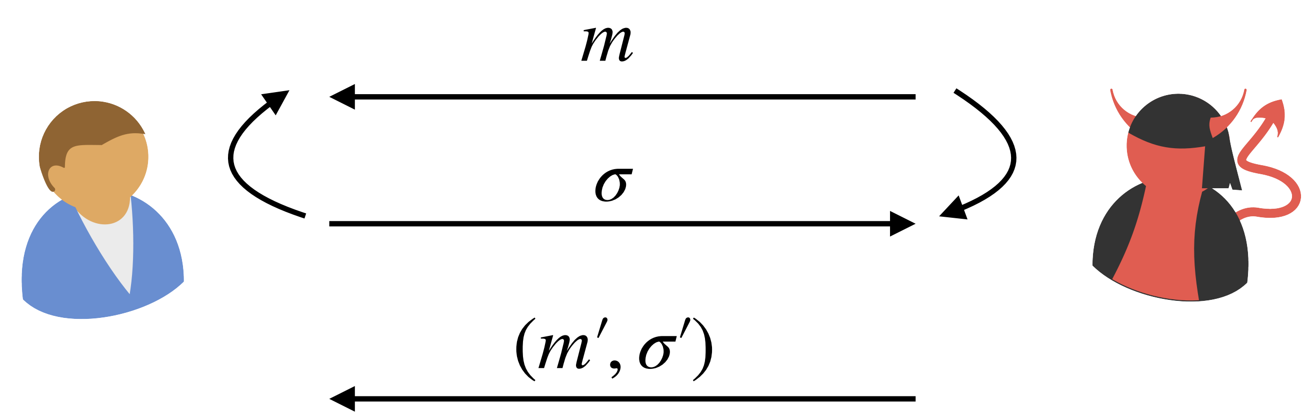


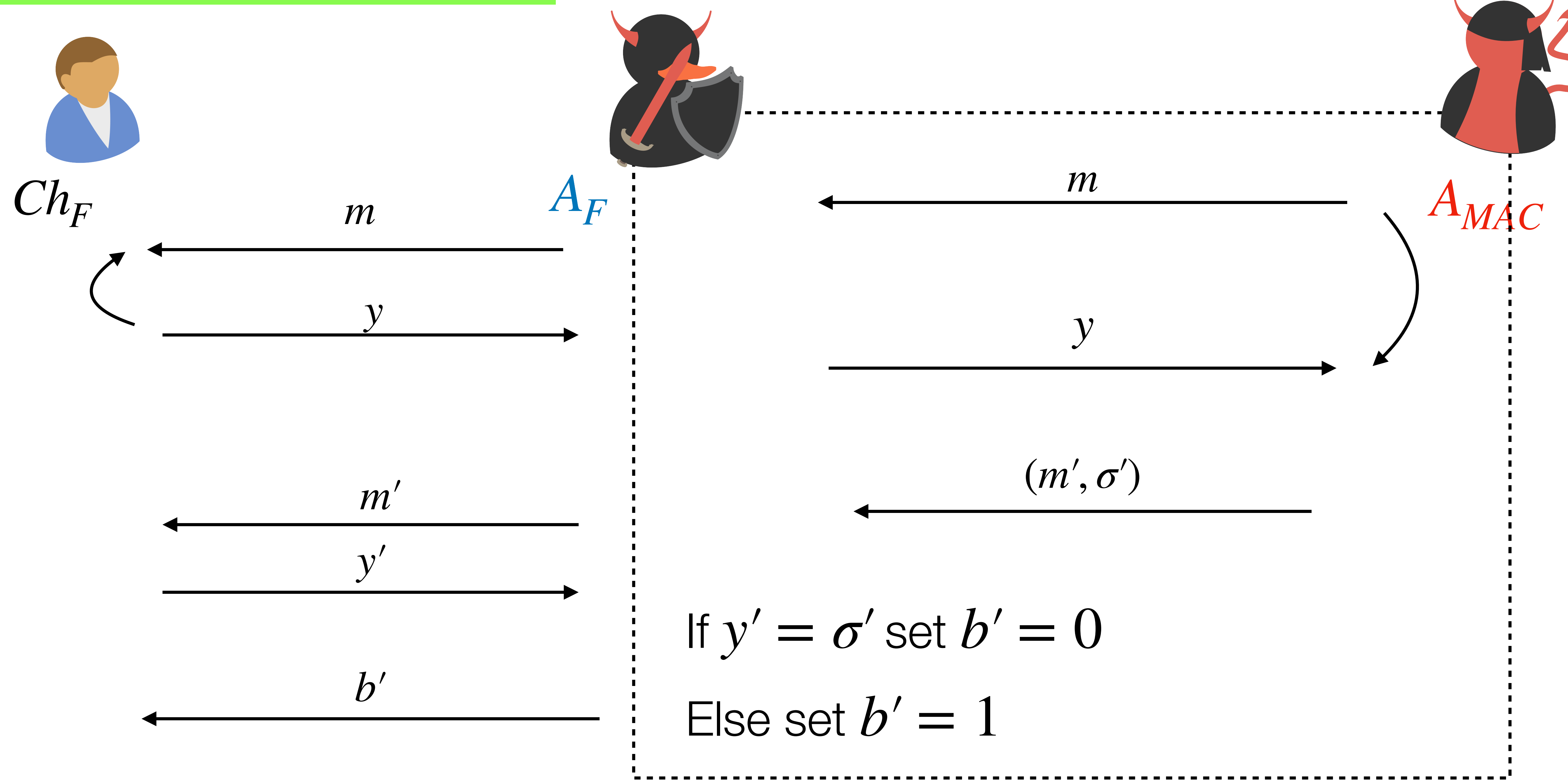
Claim:
$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$





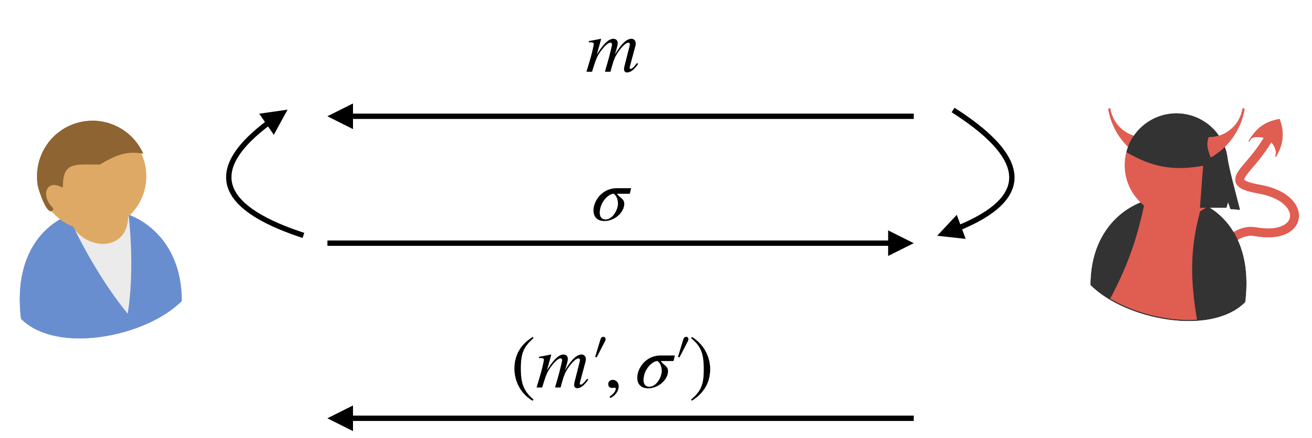
Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0 $k \xleftarrow{\$} \{0,1\}^\lambda$, $\sigma := F_k(m)$

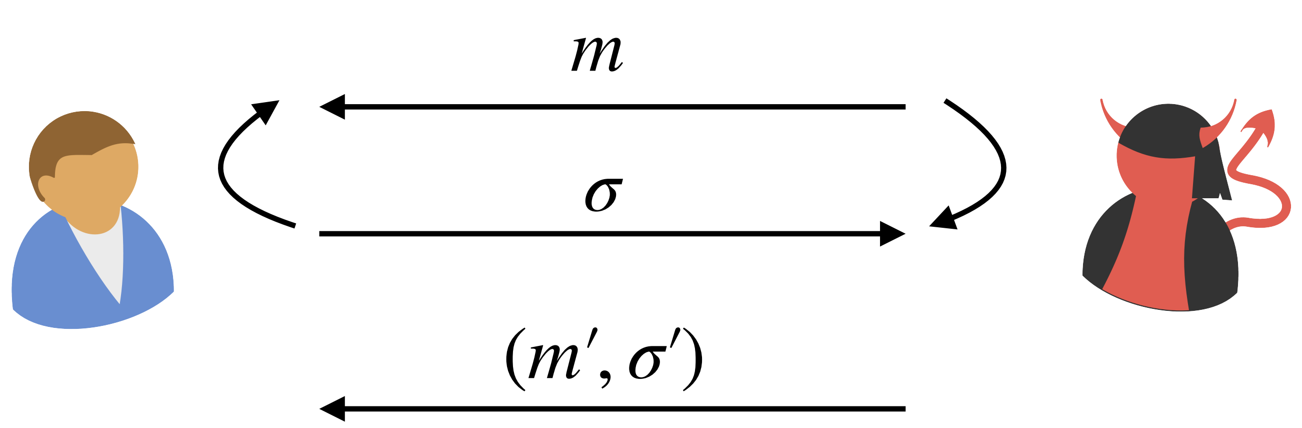


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

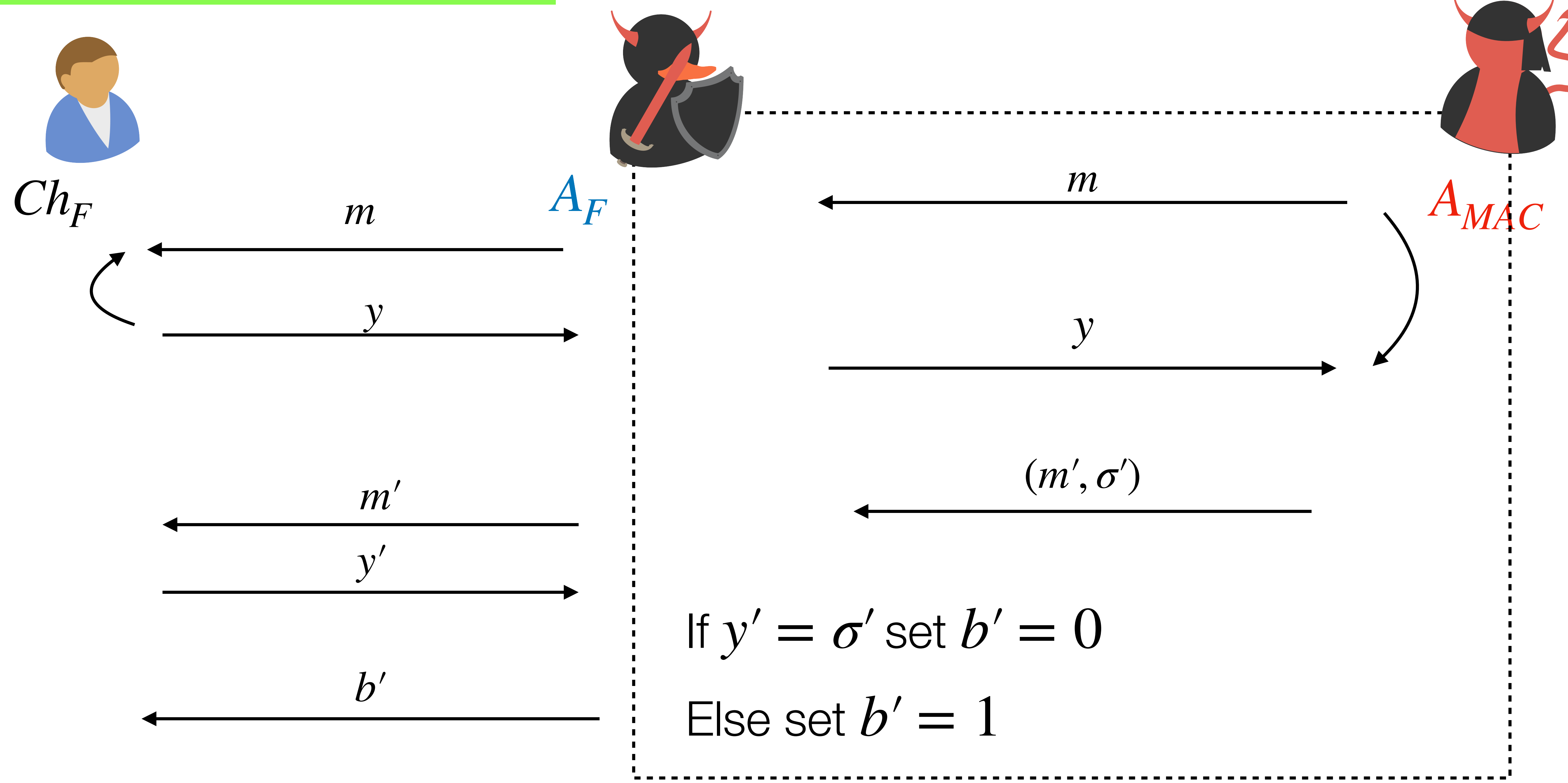
Claim:

$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$

H_1 $\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



$\Pr[W_0] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in Game}_0]$

Proof of Security

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

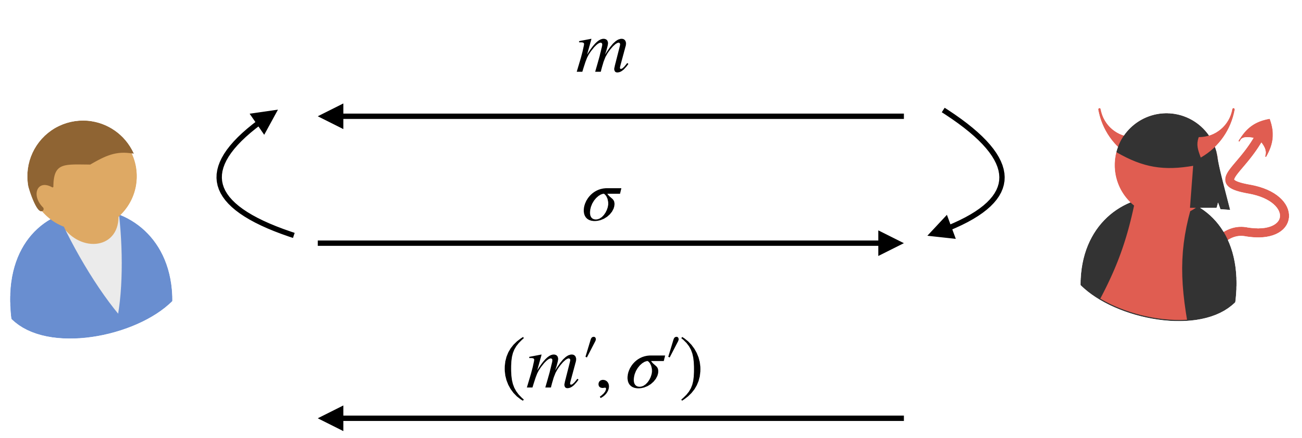
$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



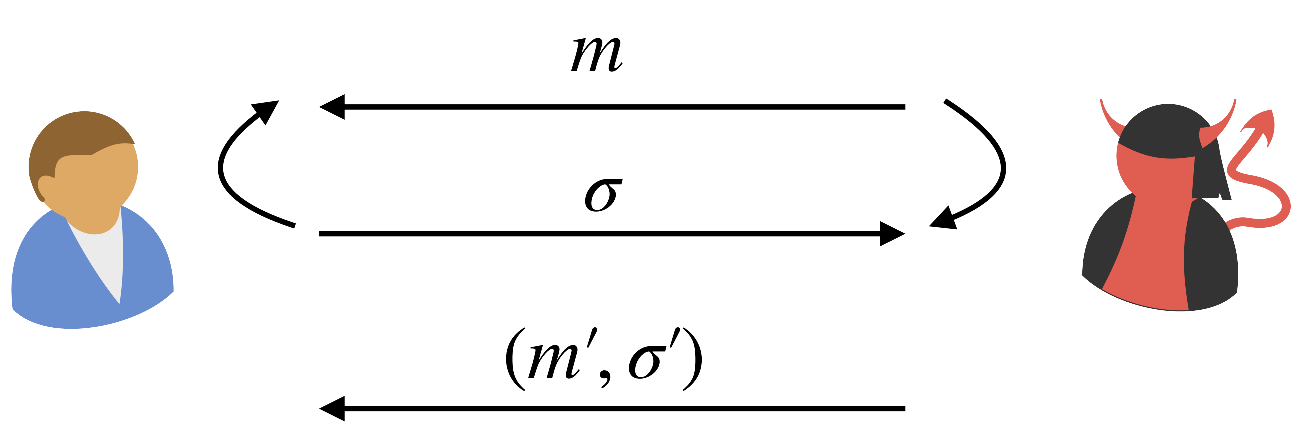
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

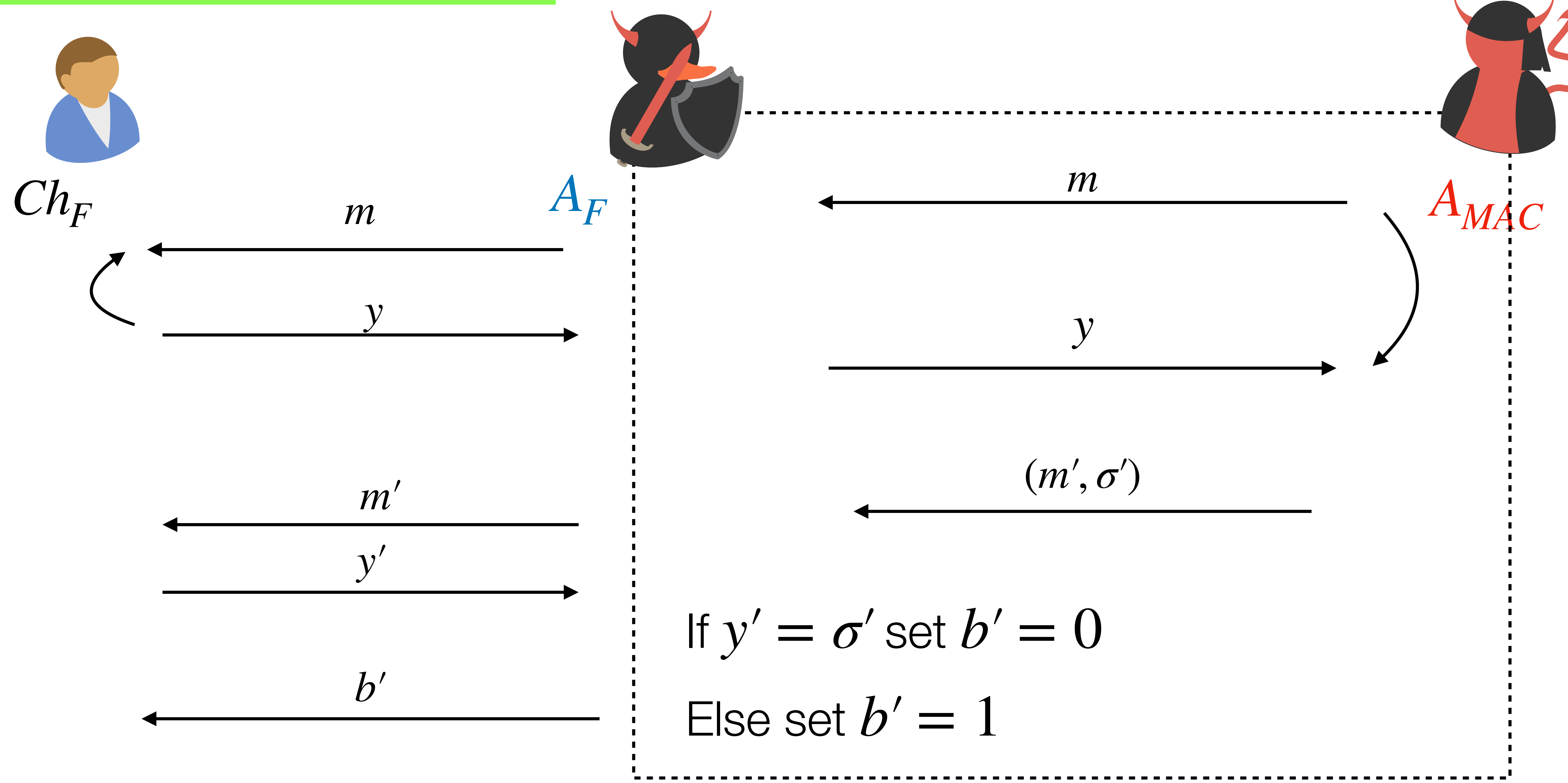
$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \mathsf{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



$\Pr[W_0] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in } \mathsf{Game}_0]$

$= \Pr[\mathscr{A}_{MAC} \text{ wins } H_0]$

Proof of Security

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

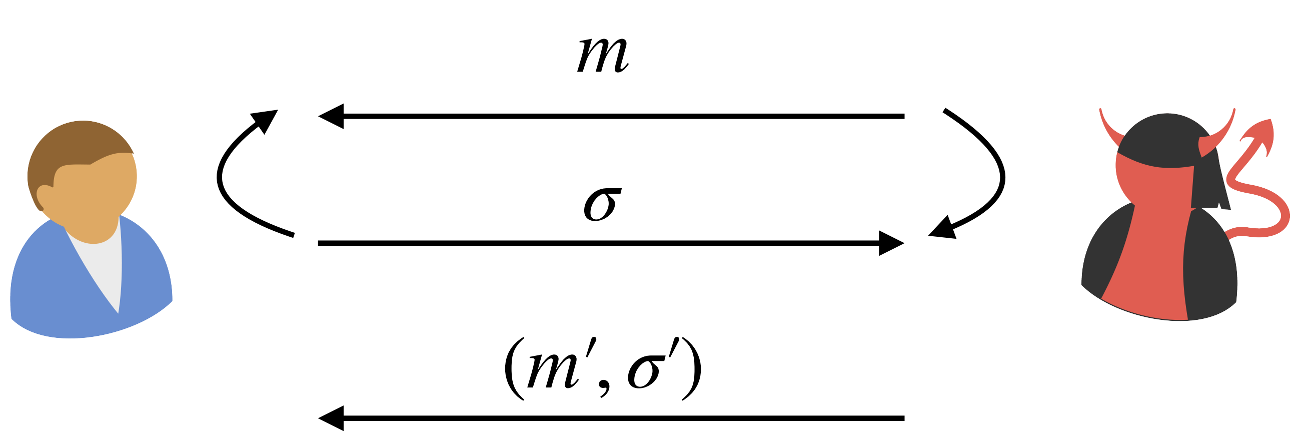
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$



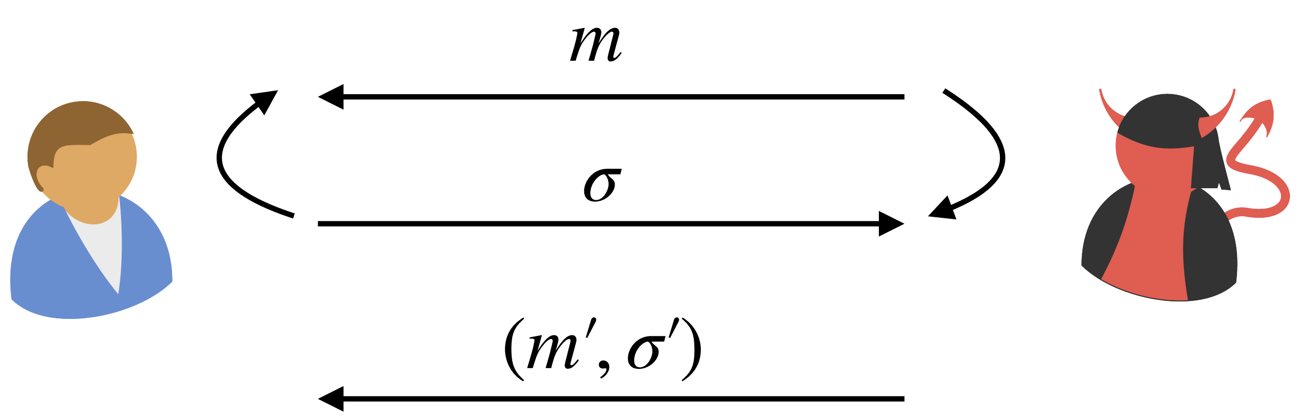
Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

Claim:

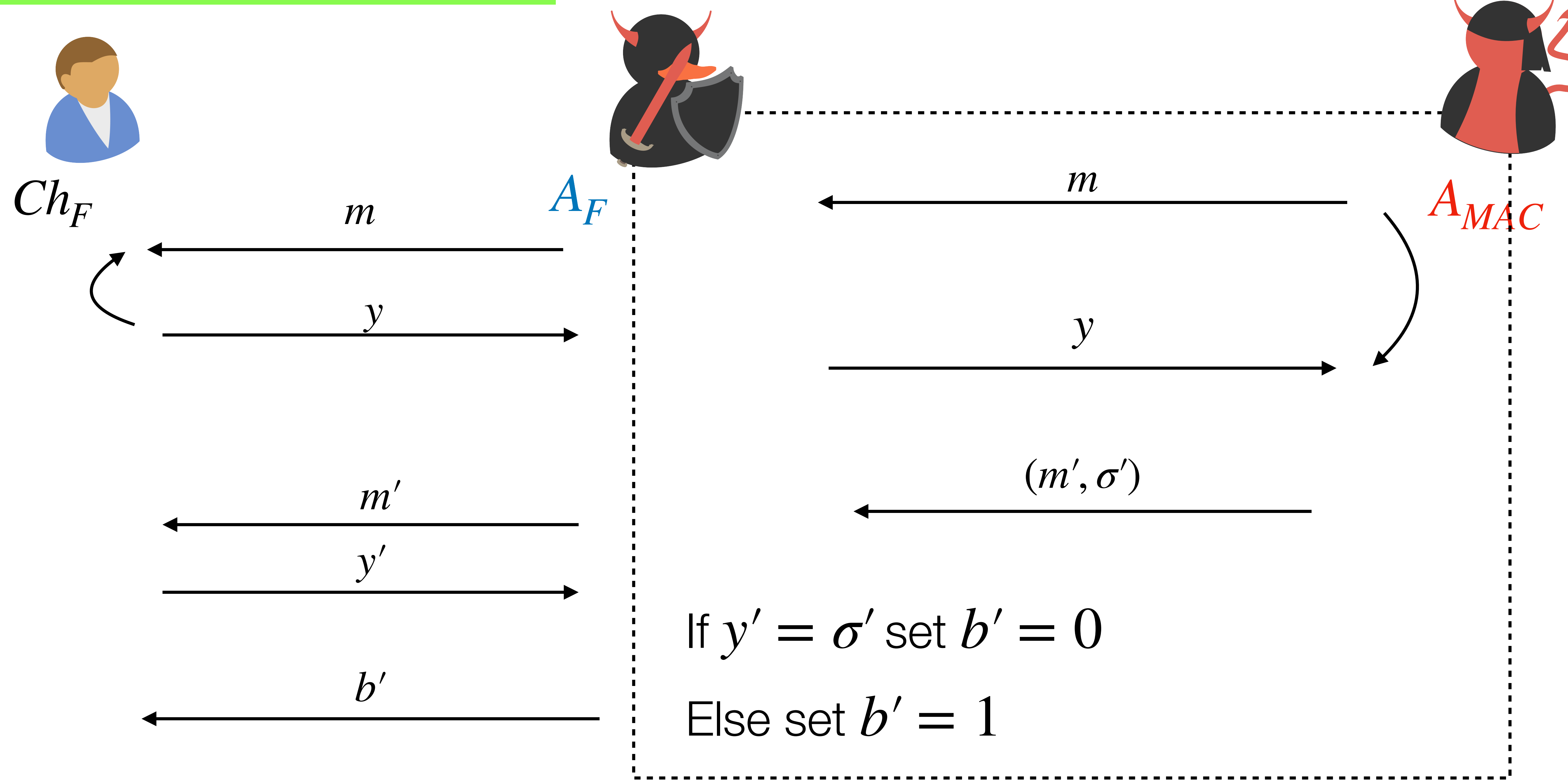
$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



$\Pr[W_0] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in Game}_0]$

$= \Pr[\mathscr{A}_{MAC} \text{ wins } H_0]$

Proof of Security

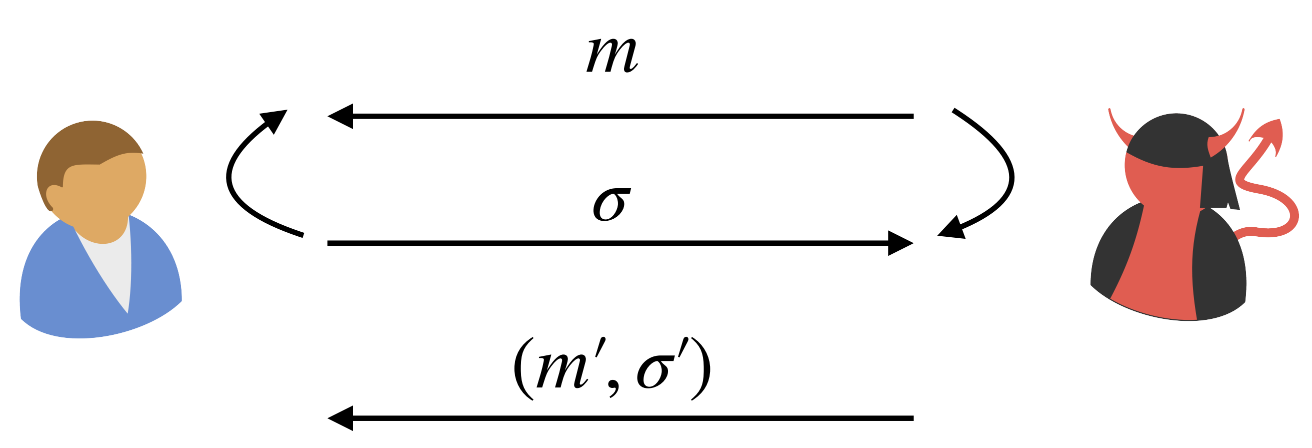
$\mathsf{KeyGen}(1^\lambda) : k \xleftarrow{\$} \{0,1\}^\lambda$

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$
$\sigma := F_k(m)$

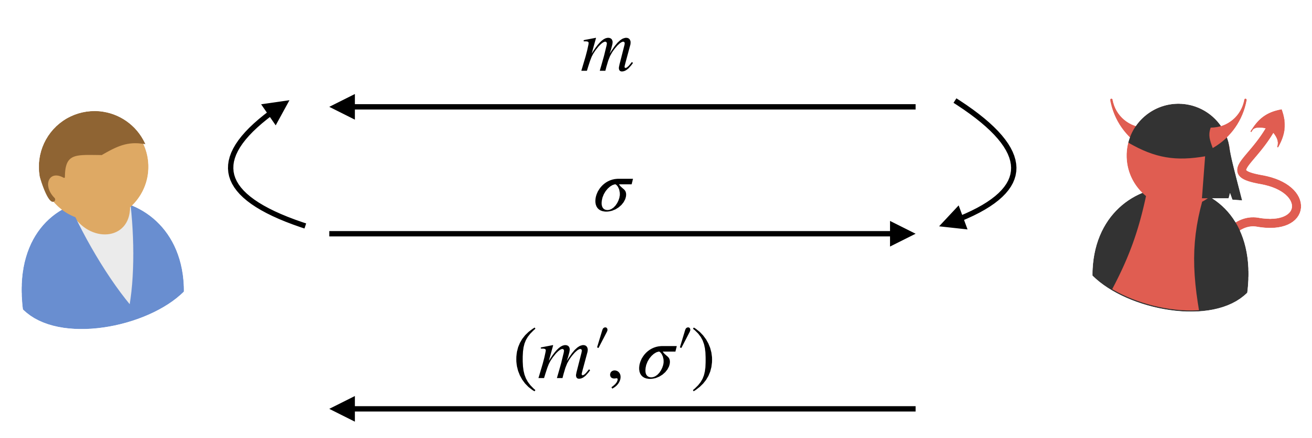


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

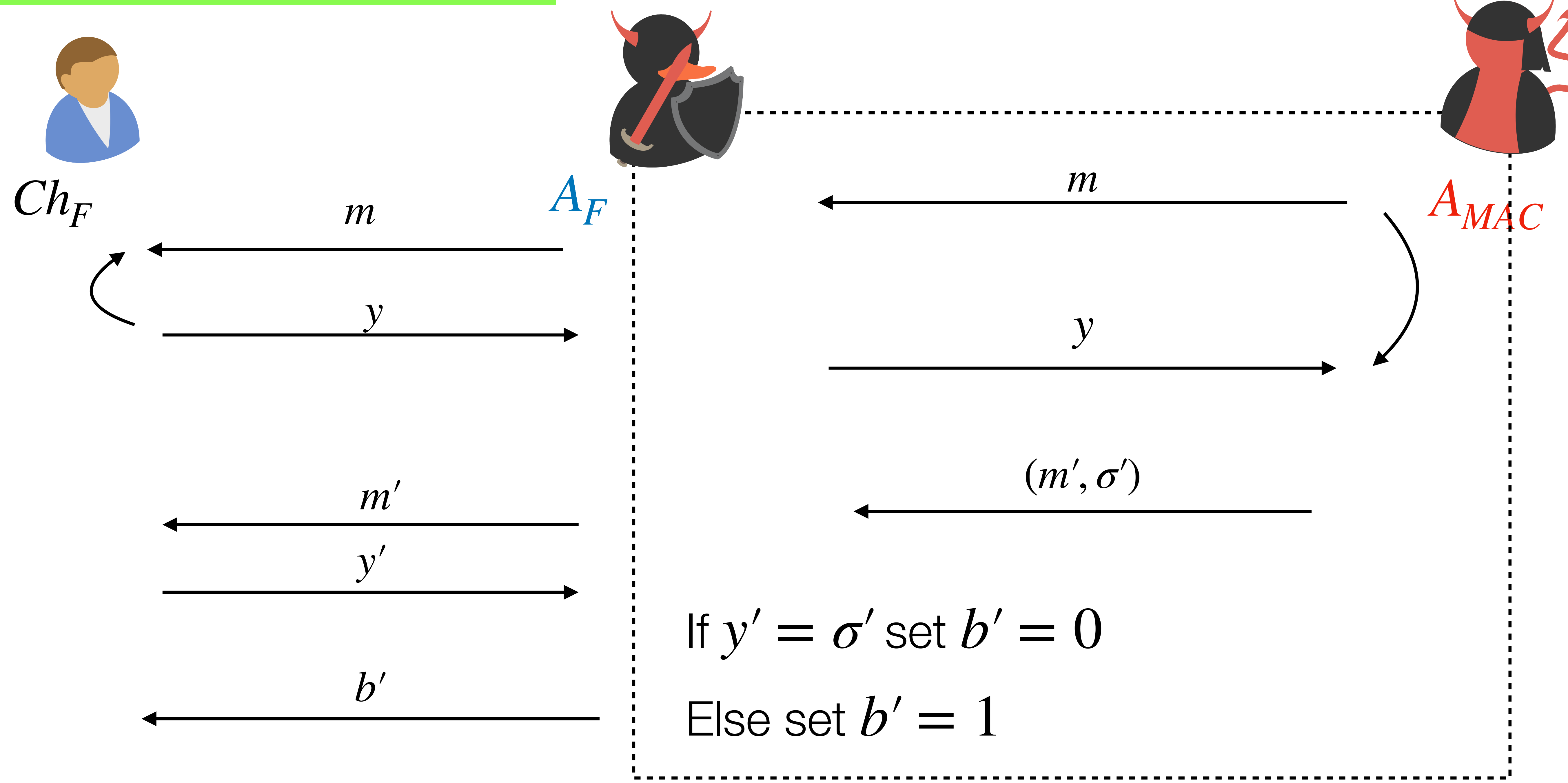
Claim:
$\Big| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \Big| \leq \mathsf{negl}(\lambda)$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'



$\Pr[W_0] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in } \mathsf{Game}_0]$

$= \Pr[\mathscr{A}_{MAC} \text{ wins } H_0]$

$\Pr[W_1] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in } \mathsf{Game}_1]$

Proof of Security

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

Tag(k, m) : $\sigma := F_k(m)$

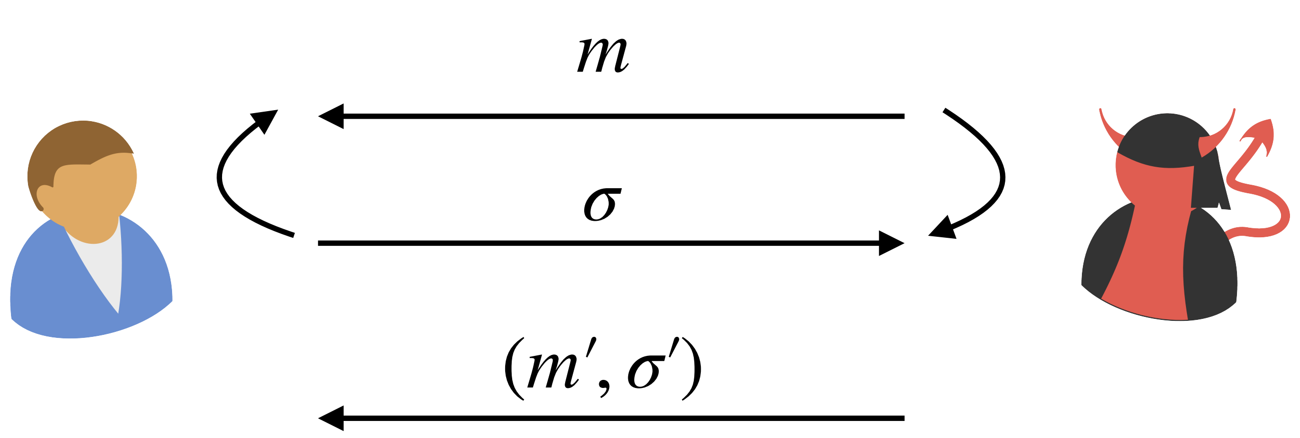
Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

H_0

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

$\sigma := F_k(m)$



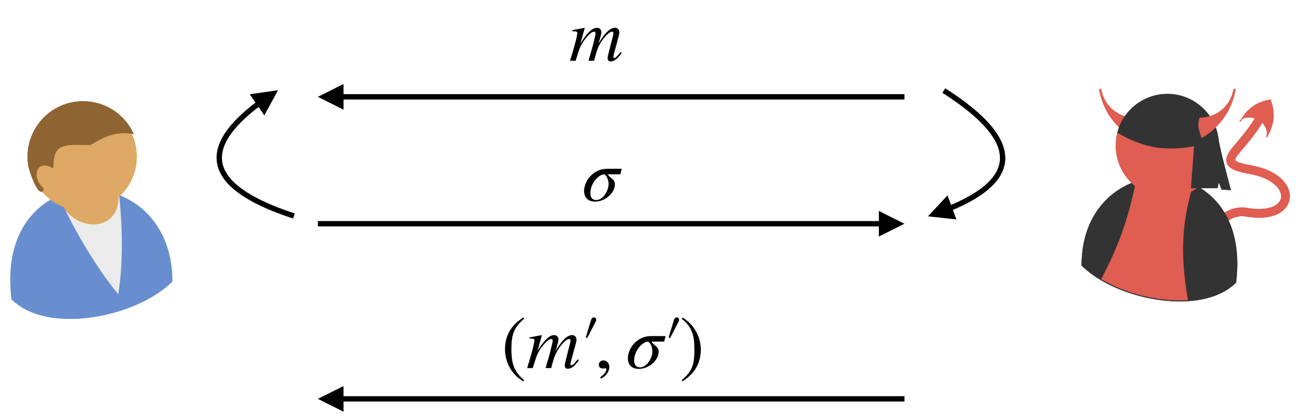
Claim:

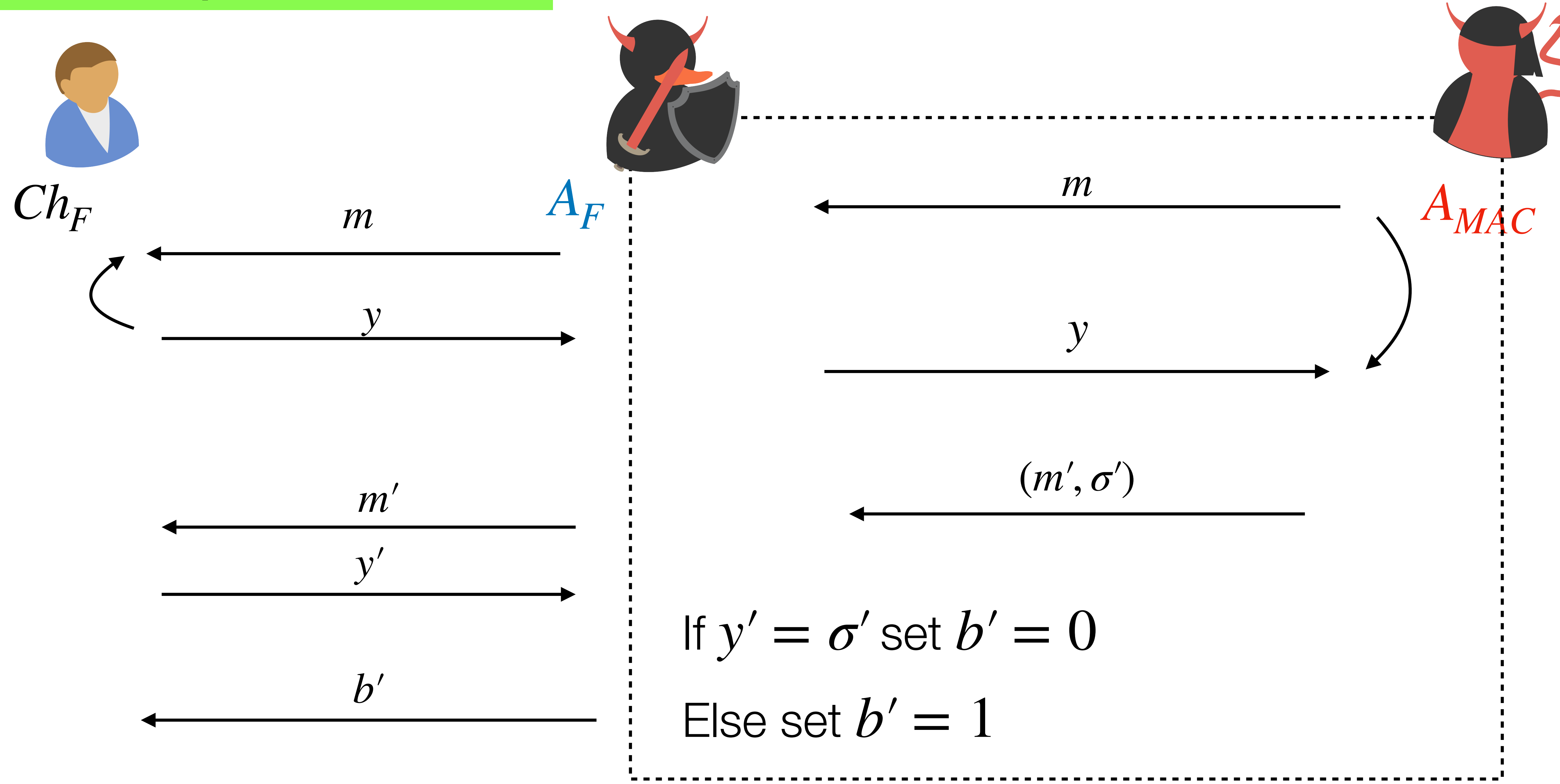
$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$

Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

H_1

$\sigma := T[m]$





$\Pr[W_0] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in Game}_0]$

$= \Pr[\mathscr{A}_{MAC} \text{ wins } H_0]$

$\Pr[W_1] = \Pr[\mathscr{A}_F \text{ outputs } 0 \text{ in Game}_1]$

$= \Pr[\mathscr{A}_{MAC} \text{ wins } H_1]$

Proof of Security

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

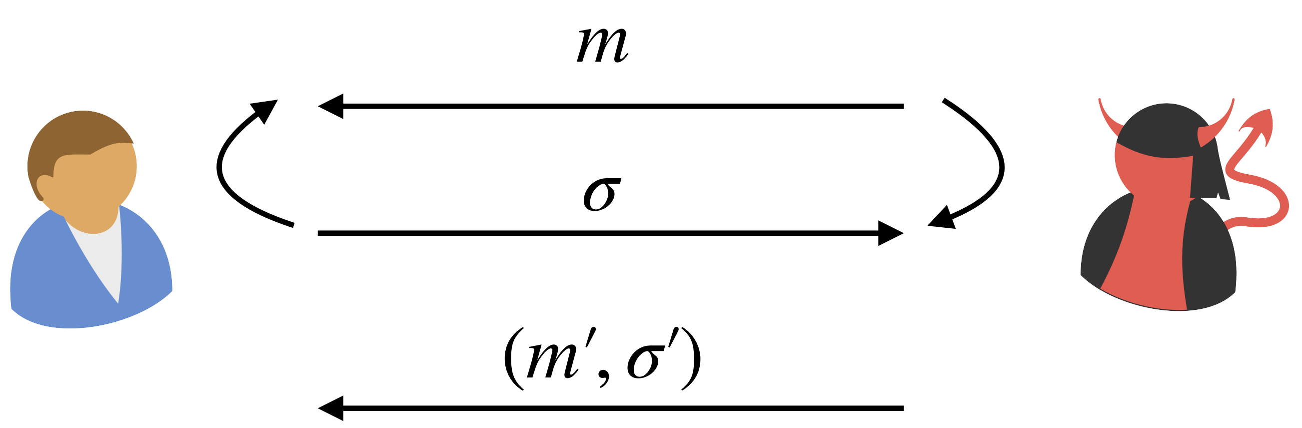
$\text{Tag}(k, m) : \sigma := F_k(m)$

$\text{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$

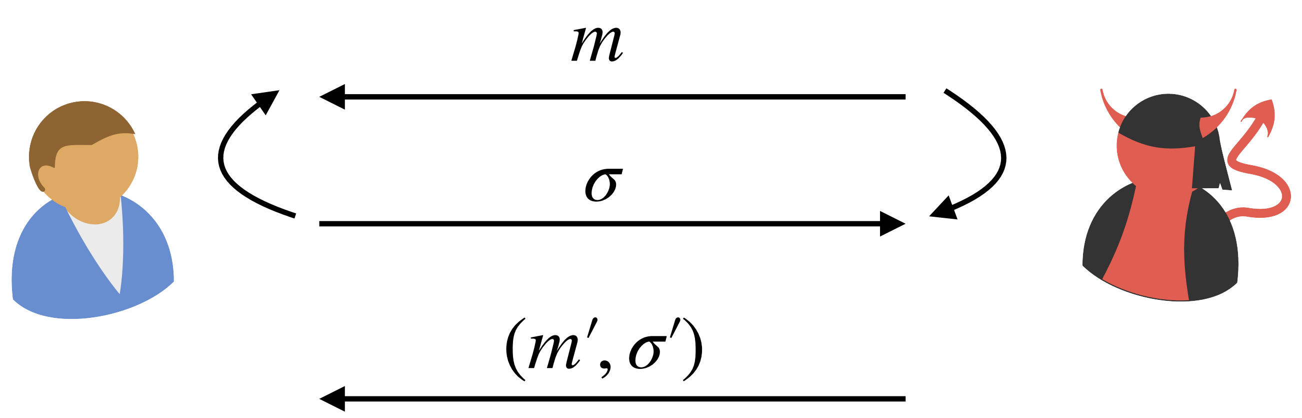


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Proof of Security

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

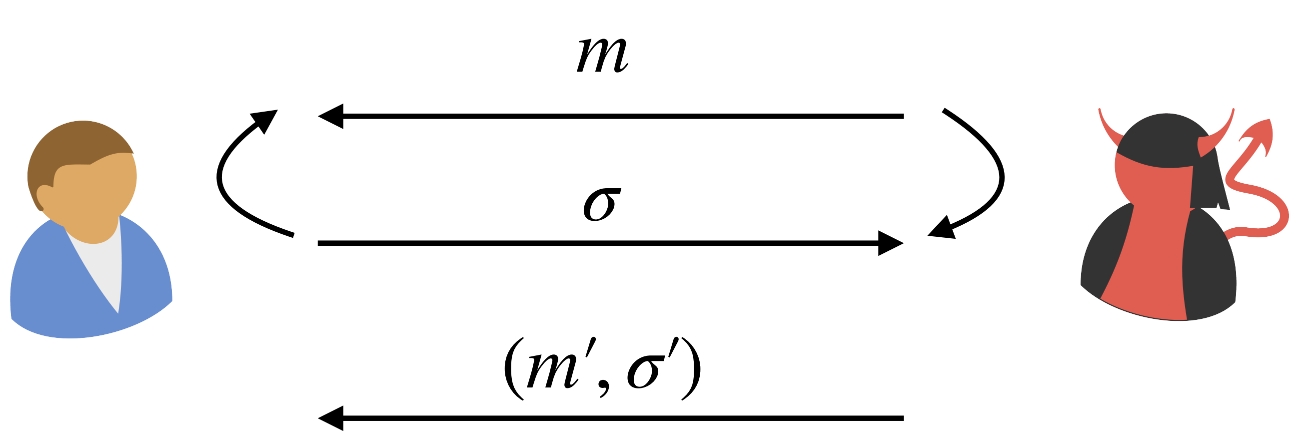
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$

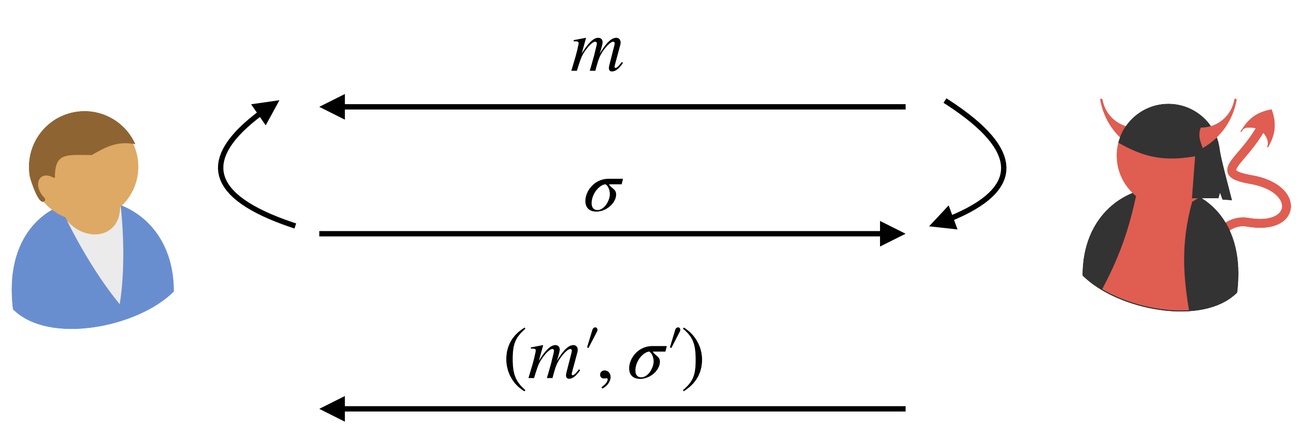


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\left| \text{Pr}[\mathscr{A} \text{ wins in } H_0] - \text{Pr}[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Claim: $\text{Pr}[\mathscr{A} \text{ wins in } H_1] \leq \text{negl}(\lambda)$

Proof of Security

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

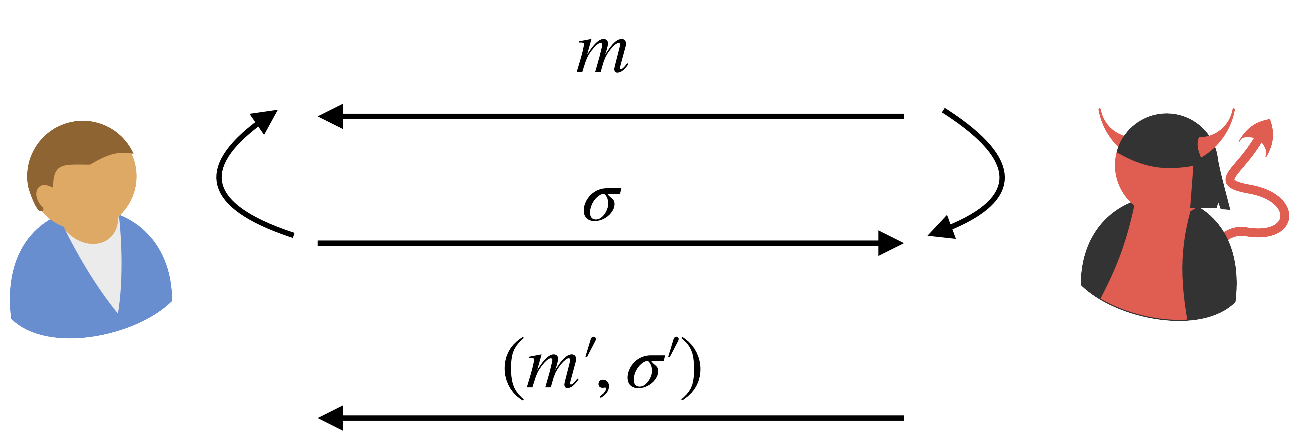
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$

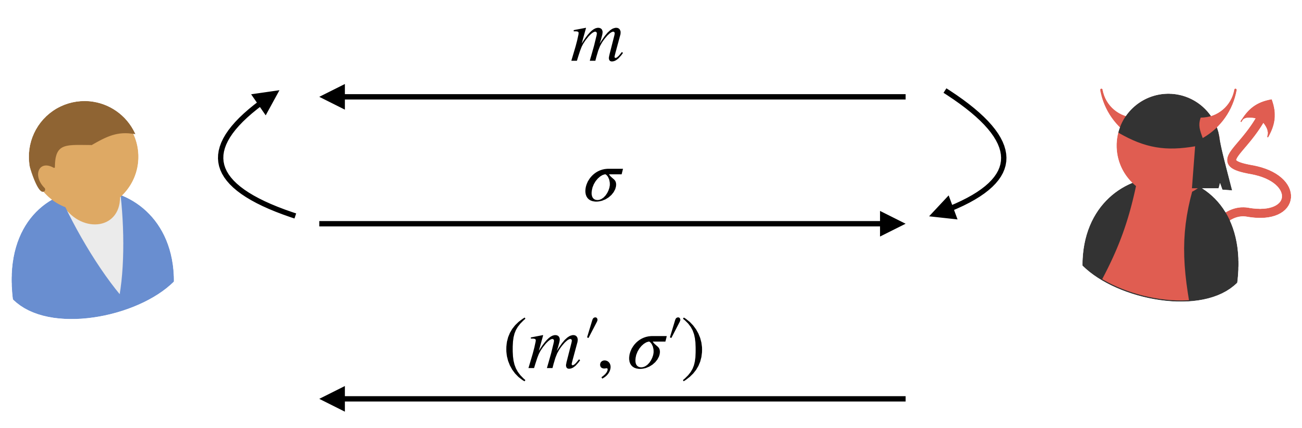


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\left| \text{Pr}[\mathscr{A} \text{ wins in } H_0] - \text{Pr}[\mathscr{A} \text{ wins in } H_1] \right| \le \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Claim: $\text{Pr}[\mathscr{A} \text{ wins in } H_1] \le \text{negl}(\lambda)$

Proof:

Proof of Security

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

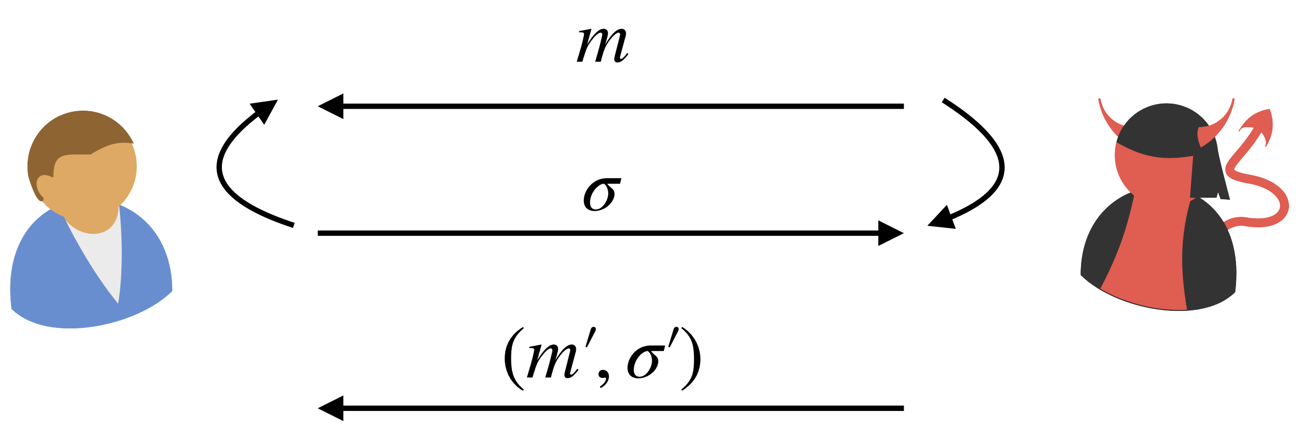
Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \overset{?}{=} \sigma$

H_0

$k \overset{\$}{\leftarrow} \{0,1\}^\lambda$

$\sigma := F_k(m)$

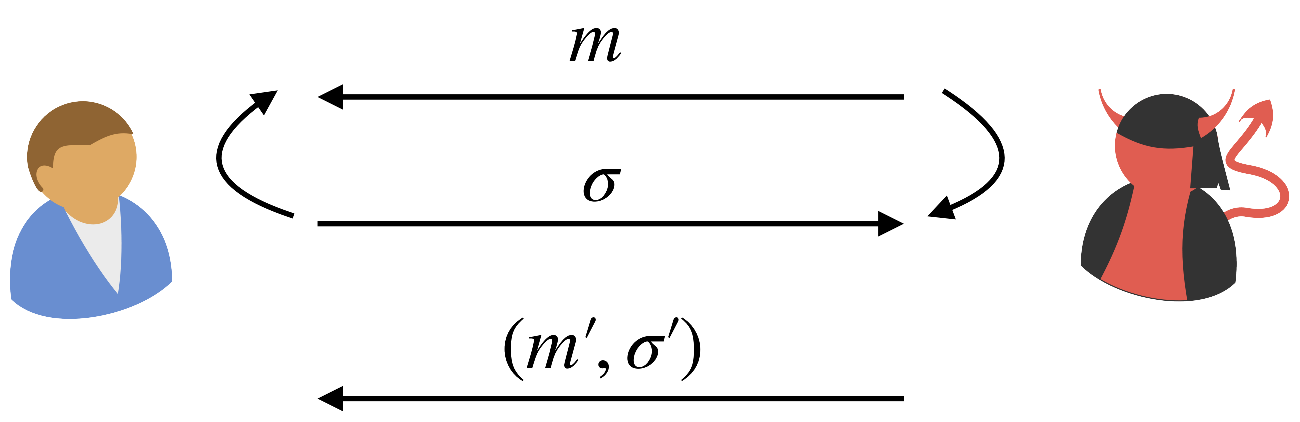


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\Big| \text{Pr}[\mathscr{A} \text{ wins in } H_0] - \text{Pr}[\mathscr{A} \text{ wins in } H_1] \Big| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Claim: $\text{Pr}[\mathscr{A} \text{ wins in } H_1] \leq \text{negl}(\lambda)$

Proof:

- Remember: T is just a table evaluating a random function!

Proof of Security

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

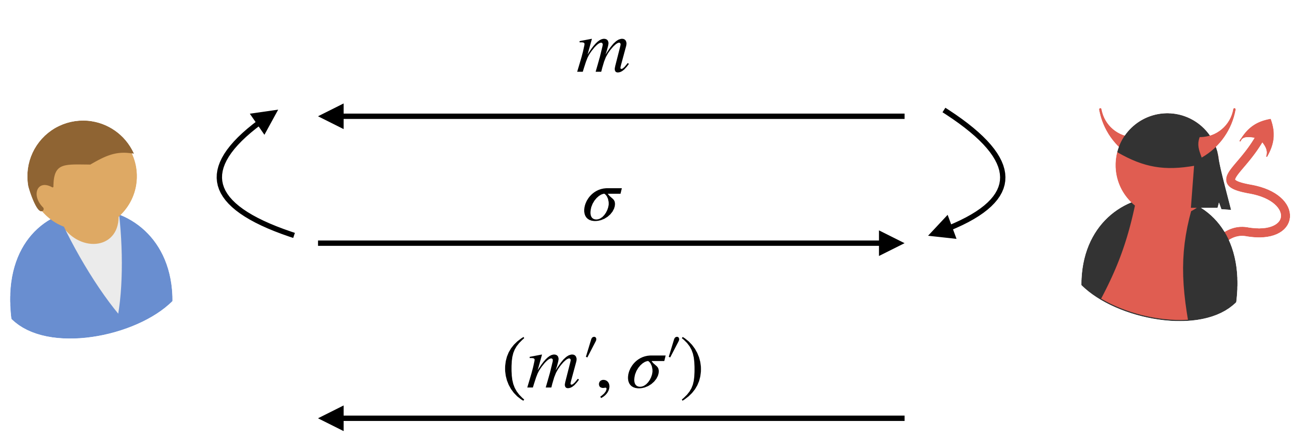
$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$

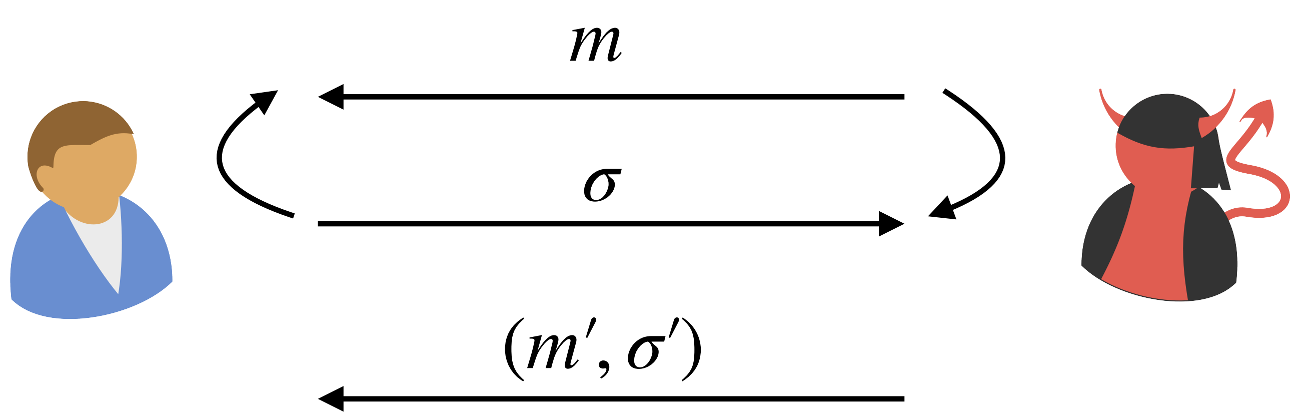


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \mathsf{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

Claim: $\Pr[\mathscr{A} \text{ wins in } H_1] \leq \mathsf{negl}(\lambda)$

Proof:

- Remember: T is just a table evaluating a random function!
- So, $\Pr[T[m'] = \sigma'] = \dfrac{1}{2^\lambda} = \mathsf{negl}(\lambda)$

Proof of Security

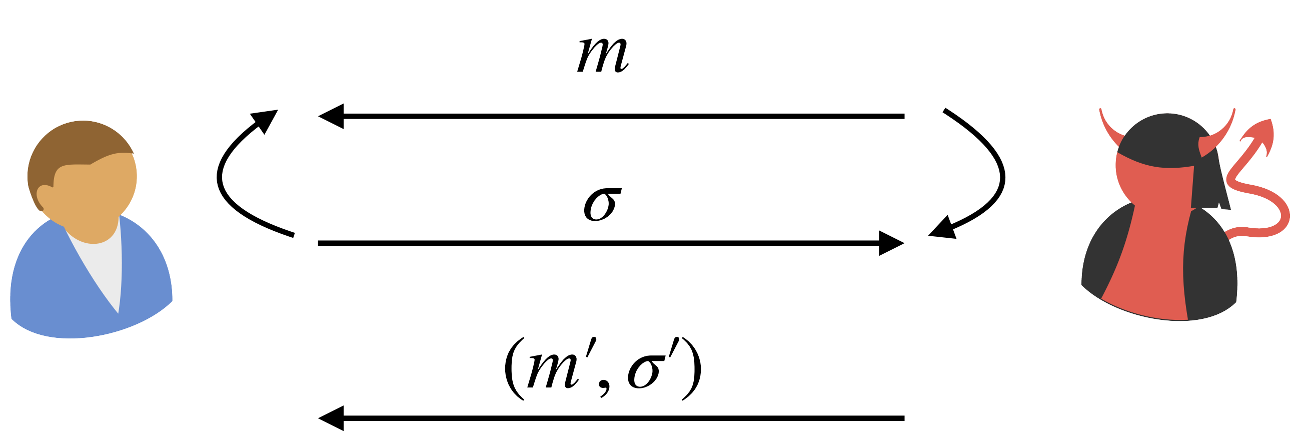
KeyGen(1^λ) : $k \xleftarrow{\$} \{0,1\}^\lambda$

Tag(k, m) : $\sigma := F_k(m)$

Ver(k, m, σ) : $F_k(m) \stackrel{?}{=} \sigma$

H_0

$k \xleftarrow{\$} \{0,1\}^\lambda$
$\sigma := F_k(m)$

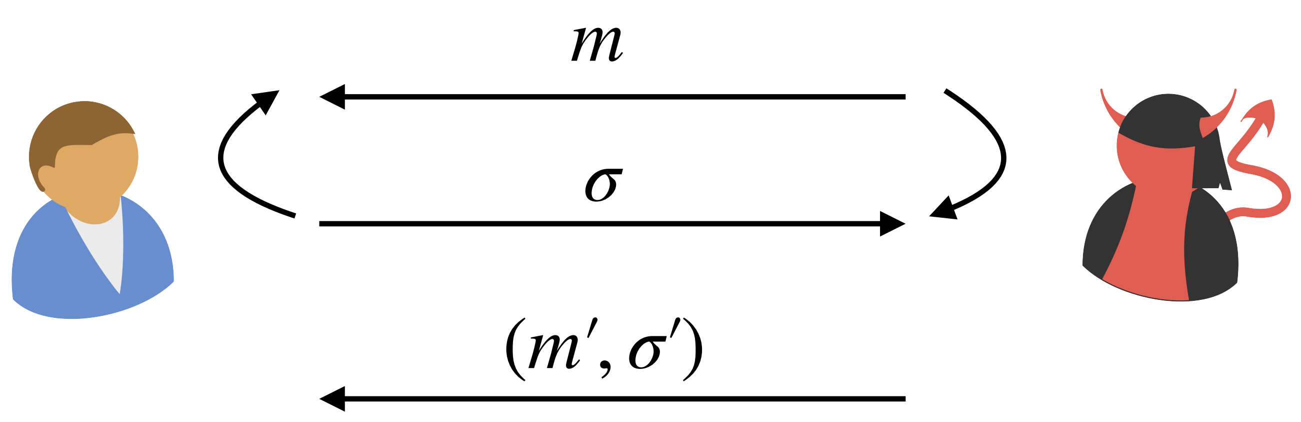


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\left| \text{Pr}[\mathscr{A} \text{ wins in } H_0] - \text{Pr}[\mathscr{A} \text{ wins in } H_1] \right| \leq \text{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

$$\text{Pr}[\mathscr{A} \text{ wins in } H_1] \leq \text{negl}(\lambda)$$

Proof of Security

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

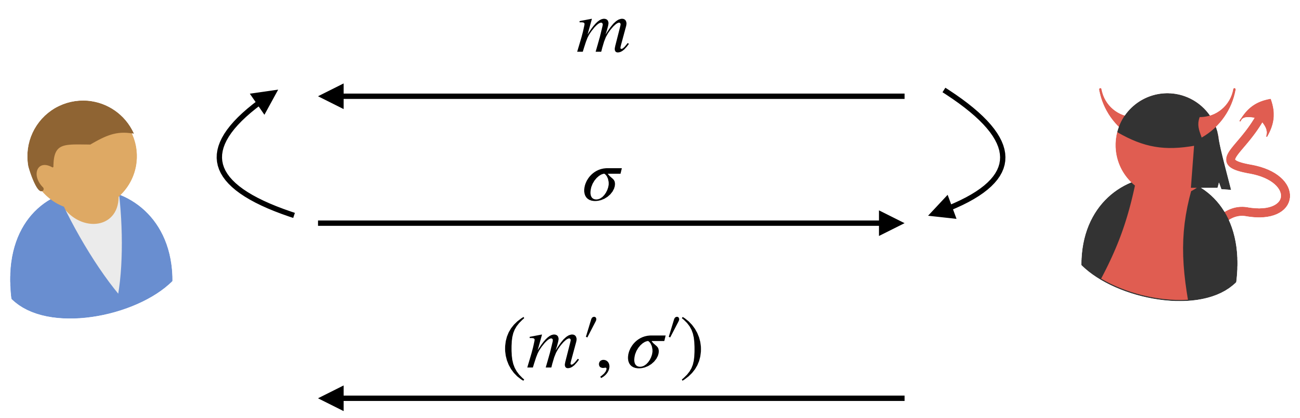
$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

H_0

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

$\sigma := F_k(m)$

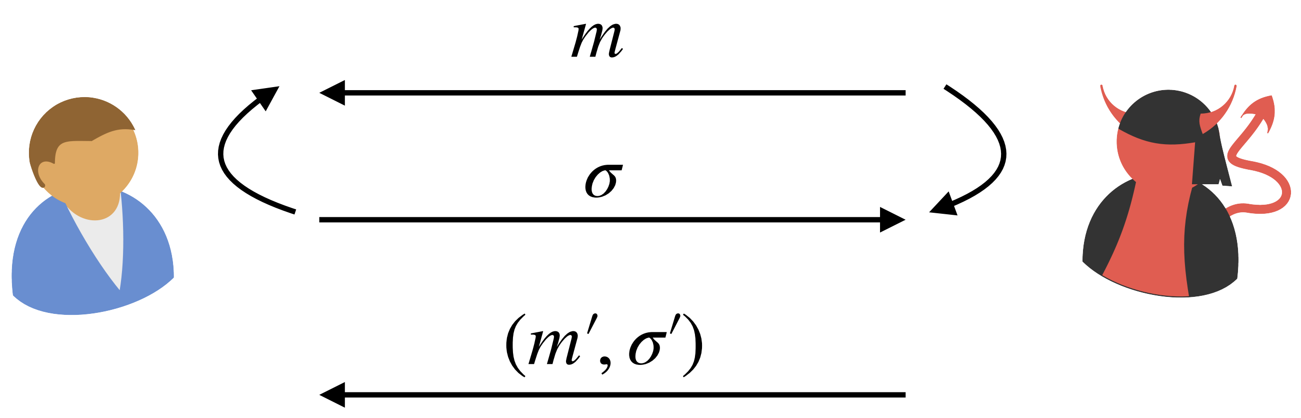


Wins if $F_k(m') = \sigma'$ and $\mathscr{A}$ never queried m'

$$\left| \Pr[\mathscr{A} \text{ wins in } H_0] - \Pr[\mathscr{A} \text{ wins in } H_1] \right| \leq \mathsf{negl}(\lambda)$$

H_1

$\sigma := T[m]$



Wins if $T[m'] = \sigma'$ and $\mathscr{A}$ never queried m'

$$\Pr[\mathscr{A} \text{ wins in } H_1] \leq \mathsf{negl}(\lambda)$$

(Triangle inequality)

$$\Pr[\mathscr{A} \text{ wins in } H_0] \leq \mathsf{negl}(\lambda)$$

Hash Functions

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

Hash Functions

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

- The PRF MAC construction is secure, but would be burdensome to use for long (or variable length) messages.

Hash Functions

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

- The PRF MAC construction is secure, but would be burdensome to use for long (or variable length) messages.

- What we want is a way to *compress* the message m, and then only sign the *compressed* version.

Hash Functions

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

- The PRF MAC construction is secure, but would be burdensome to use for long (or variable length) messages.
- What we want is a way to *compress* the message m, and then only sign the *compressed* version.
- Critical for security! If $\mathscr{A}$ could find some other message m' that compresses to the same value, they could forge a signature (just use the signature for m)!

Hash Functions

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

- The PRF MAC construction is secure, but would be burdensome to use for long (or variable length) messages.
- What we want is a way to *compress* the message m, and then only sign the *compressed* version.
- Critical for security! If $\mathscr{A}$ could find some other message m' that compresses to the same value, they could forge a signature (just use the signature for m)!
- A function h such that h compresses the input, and it is hard to find x, x' such that $x \neq x'$ but $h(x) = h(x')$

Hash Functions

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

$\mathsf{Tag}(k, m) : \sigma := F_k(m)$

$\mathsf{Ver}(k, m, \sigma) : F_k(m) \stackrel{?}{=} \sigma$

- The PRF MAC construction is secure, but would be burdensome to use for long (or variable length) messages.
- What we want is a way to *compress* the message m, and then only sign the *compressed* version.
- Critical for security! If $\mathscr{A}$ could find some other message m' that compresses to the same value, they could forge a signature (just use the signature for m)!
- A function h such that h compresses the input, and it is hard to find x, x' such that $x \neq x'$ but $h(x) = h(x')$
- Like with PRFs, we will consider *families* of hash functions (as otherwise non-uniform adversaries could just have collisions "built in").

Hash Functions

Collision-resistant Hash Function Family

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \rightarrow R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \to R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

- **Easy to Sample**: There exists a PPT Gen such that: $i \leftarrow \mathsf{Gen}(1^\lambda)$, $i \in I$

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \rightarrow R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

- **Easy to Sample**: There exists a PPT Gen such that: $i \leftarrow \text{Gen}(1^\lambda)$, $i \in I$
- **Compressing**: For all $i \in I$, $|R_i| \leq |D_i|$

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \to R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

- **Easy to Sample**: There exists a PPT Gen such that: $i \leftarrow \mathsf{Gen}(1^\lambda)$, $i \in I$
- **Compressing**: For all $i \in I$, $|R_i| \leq |D_i|$
- **Easy to Compute**: There exists a poly-time algorithm Eval such that given $x \in D_i$, $i \in I$, $\mathsf{Eval}(x, i) = h_i(x)$

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \to R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

- **Easy to Sample**: There exists a PPT Gen such that: $i \leftarrow \mathsf{Gen}(1^\lambda), i \in I$
- **Compressing**: For all $i \in I, \ |R_i| \leq |D_i|$
- **Easy to Compute**: There exists a poly-time algorithm Eval such that given $x \in D_i, i \in I$, $\mathsf{Eval}(x, i) = h_i(x)$
- **Collision Resistance**: For al NUPPT $\mathscr{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \to R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

- **Easy to Sample**: There exists a PPT Gen such that: $i \leftarrow \mathsf{Gen}(1^\lambda),\ i \in I$
- **Compressing**: For all $i \in I,\ |R_i| \leq |D_i|$
- **Easy to Compute**: There exists a poly-time algorithm Eval such that given $x \in D_i$, $i \in I$, $\mathsf{Eval}(x, i) = h_i(x)$
- **Collision Resistance**: For al NUPPT $\mathscr{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\begin{matrix} x \neq x' \wedge \\ h_i(x) = h_i(x') \end{matrix} : \begin{matrix} i \leftarrow \mathsf{Gen}(1^\lambda) \\ (x, x') \leftarrow \mathscr{A}(1^\lambda, i) \end{matrix}\right] \leq \mathsf{negl}(\lambda)$$

Hash Functions

Collision-resistant Hash Function Family

A deterministic family of functions $H = \{h_i : D_i \to R_i\}_{i \in I}$ is called a collision-resistant hash function family (CRHF) if it satisfies the following properties:

- **Easy to Sample**: There exists a PPT Gen such that: $i \leftarrow \mathsf{Gen}(1^\lambda)$, $i \in I$
- **Compressing**: For all $i \in I$, $|R_i| \leq |D_i|$
- **Easy to Compute**: There exists a poly-time algorithm Eval such that given $x \in D_i$, $i \in I$, $\mathsf{Eval}(x, i) = h_i(x)$
- **Collision Resistance**: For al NUPPT $\mathscr{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\mathscr{A} \text{ wins CRHFGame}\right] \leq \mathsf{negl}(\lambda)$$

Collision-resistant Hash Functions

$$H = \{h_i : D_i \to R_i\}_{i \in I}$$

$$H = \{h_i : D_i \to R_i\}_{i \in I}$$

Collision-resistant Hash Functions

- One-bit compression implies arbitrary bit compression

$$H = \{h_i : D_i \to R_i\}_{i \in I}$$

Collision-resistant Hash Functions

- One-bit compression implies arbitrary bit compression
- BUT: range cannot be too small

$$H = \{h_i : D_i \to R_i\}_{i \in I}$$

Collision-resistant Hash Functions

- One-bit compression implies arbitrary bit compression
- BUT: range cannot be too small
  - What would be the issue with a hash function that has a 1 bit range?

Collision-resistant Hash Functions

$H = \{h_i : D_i \to R_i\}_{i \in I}$

- One-bit compression implies arbitrary bit compression
- BUT: range cannot be too small
  - What would be the issue with a hash function that has a 1 bit range?
- Do they actually exist?

$$H = \{h_i : D_i \to R_i\}_{i \in I}$$

Collision-resistant Hash Functions

- One-bit compression implies arbitrary bit compression
- BUT: range cannot be too small
  - What would be the issue with a hash function that has a 1 bit range?
- Do they actually exist?
  - Unlikely to be constructed from OWF or OWP

Collision-resistant Hash Functions

$H = \{h_i : D_i \to R_i\}_{i \in I}$

- One-bit compression implies arbitrary bit compression
- BUT: range cannot be too small
  - What would be the issue with a hash function that has a 1 bit range?
- Do they actually exist?
  - Unlikely to be constructed from OWF or OWP
  - Can be constructed from assumptions like factoring and discrete log (on their own!)

MAC Construction II

MAC Construction II

Hash-and-MAC

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathscr{M}$, and let $H_i = \{h_i : D \to \mathscr{M}\}$ be a CRHF family

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathscr{M}$, and let $H_i = \{h_i : D \to \mathscr{M}\}$ be a CRHF family

- KeyGen(1^λ):

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathscr{M}$, and let $H_i = \{h_i : D \to \mathscr{M}\}$ be a CRHF family

- KeyGen(1^λ):
  - $k \leftarrow$ KeyGen(1^λ)

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathcal{M}$, and let $H_i = \{h_i : D \to \mathcal{M}\}$ be a CRHF family

- KeyGen(1^λ):
  - $k \leftarrow$ KeyGen(1^λ)
  - $i \leftarrow$ Gen(1^λ)

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathscr{M}$, and let $H_i = \{h_i : D \to \mathscr{M}\}$ be a CRHF family

- KeyGen(1^λ):
  - $k \leftarrow$ KeyGen(1^λ)
  - $i \leftarrow$ Gen(1^λ)
  - return (k, i)

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathcal{M}$, and let $H_i = \{h_i : D \to \mathcal{M}\}$ be a CRHF family

- KeyGen(1^λ):
  - $k \leftarrow$ KeyGen(1^λ)
  - $i \leftarrow$ Gen(1^λ)
  - return (k, i)
- Tag$((k, i), m)$: $\sigma :=$ Tag$(k, h_i(m))$

MAC Construction II

Hash-and-MAC

Let λ be the security parameter and, Let (KeyGen, Tag, Ver) be a MAC scheme with message space $\mathcal{M}$, and let $H_i = \{h_i : D \to \mathcal{M}\}$ be a CRHF family

- KeyGen(1^λ):
  - $k \leftarrow$ KeyGen(1^λ)
  - $i \leftarrow$ Gen(1^λ)
  - return (k, i)
- Tag$((k, i), m)$: $\sigma :=$ Tag$(k, h_i(m))$
- Ver$((k, i), m, \sigma)$: Ver$(k, h_i(m), \sigma)$

Proof of Security

Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

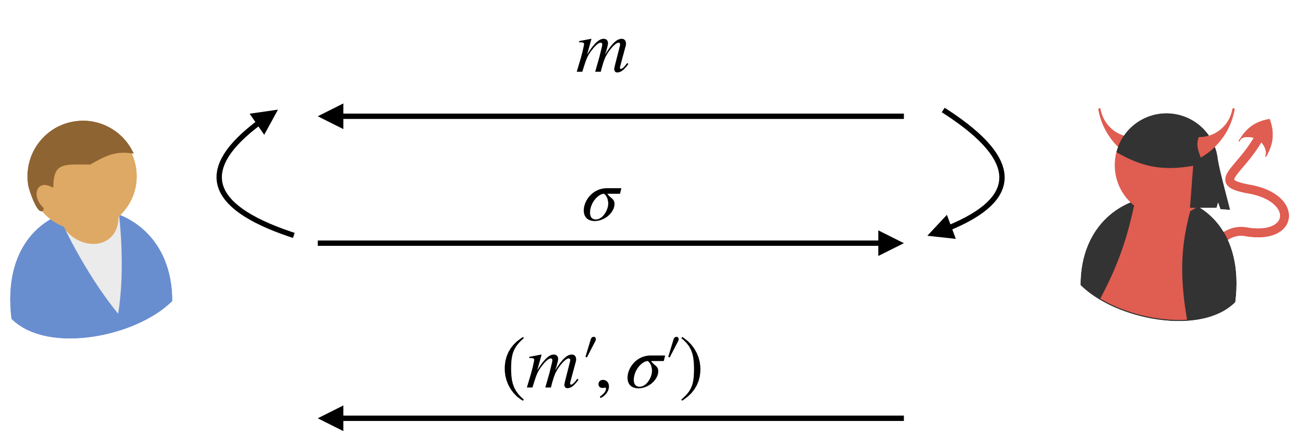
Proof of Security

H_0

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

$i \leftarrow \textsf{Gen}(1^\lambda)$

$\sigma := \textsf{Tag}(k, h_i(m))$



Wins if
$\textsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

$\textsf{KeyGen}(1^\lambda) : (k \leftarrow \textsf{KeyGen}(1^\lambda), i)$

$\textsf{Tag}(k, m) : \sigma := \textsf{Tag}(k, h_i(m))$

$\textsf{Ver}(k, m, \sigma) : \textsf{Ver}(k, h_i(m), \sigma)$

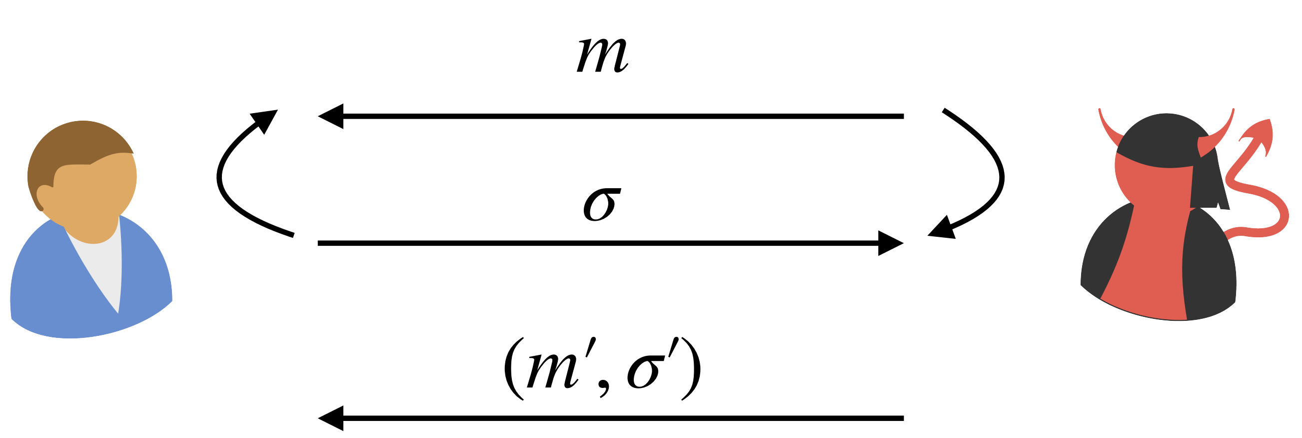
Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$



Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Claim: If $\{h_i\}$ is a secure family of CHRFs **and** (KeyGen, Tag, Ver) is a secure MAC, then **Pr**$[\mathscr{A}$ wins in $H_0] \leq$ **negl**(λ)

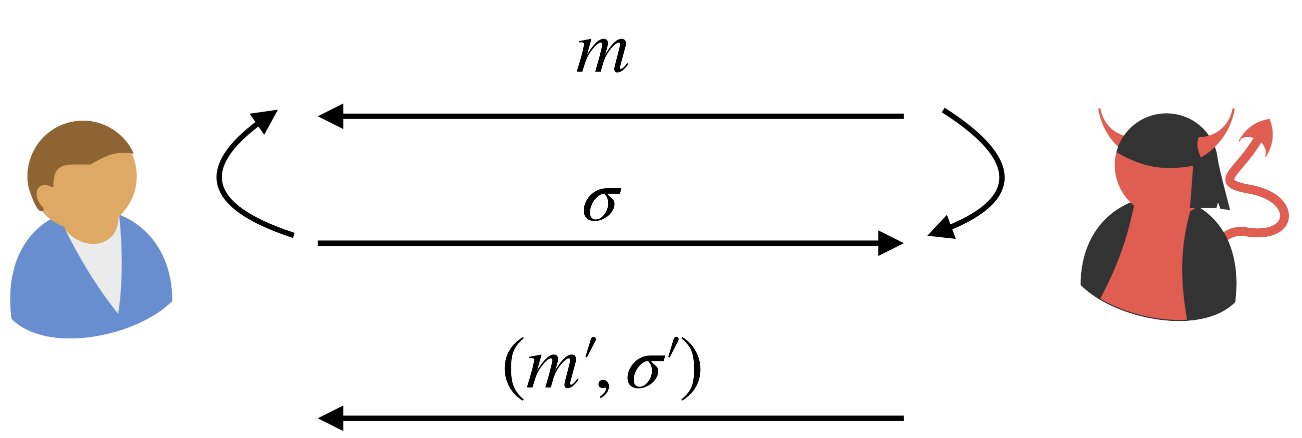
Proof of Security

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

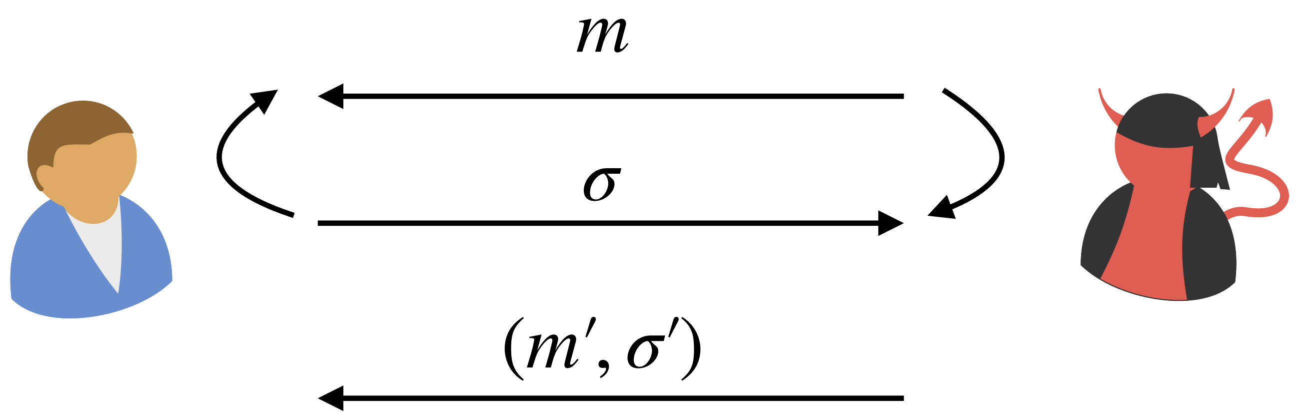
Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$



Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

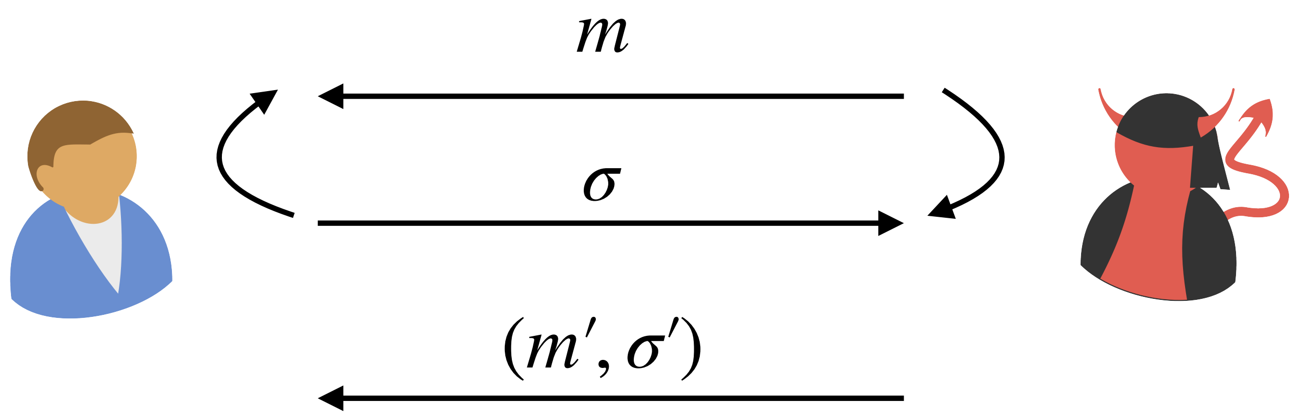
Proof of Security

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ **never queried** m'

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\Pr\left[\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$

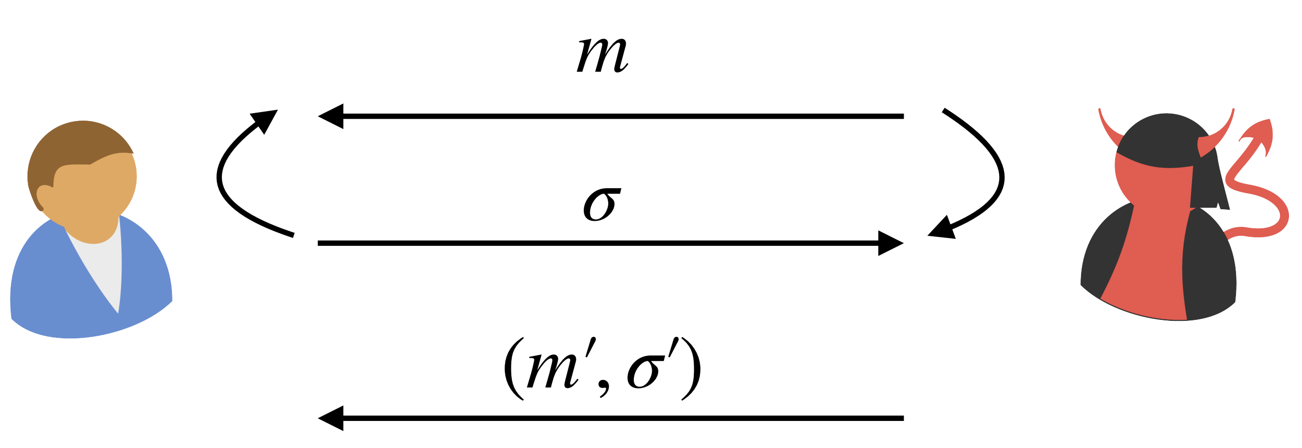
Proof of Security

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$



Wins if $\mathsf{Ver}(k, h_i(m'), \sigma') = 1$ and $\mathscr{A}$ never queried m'

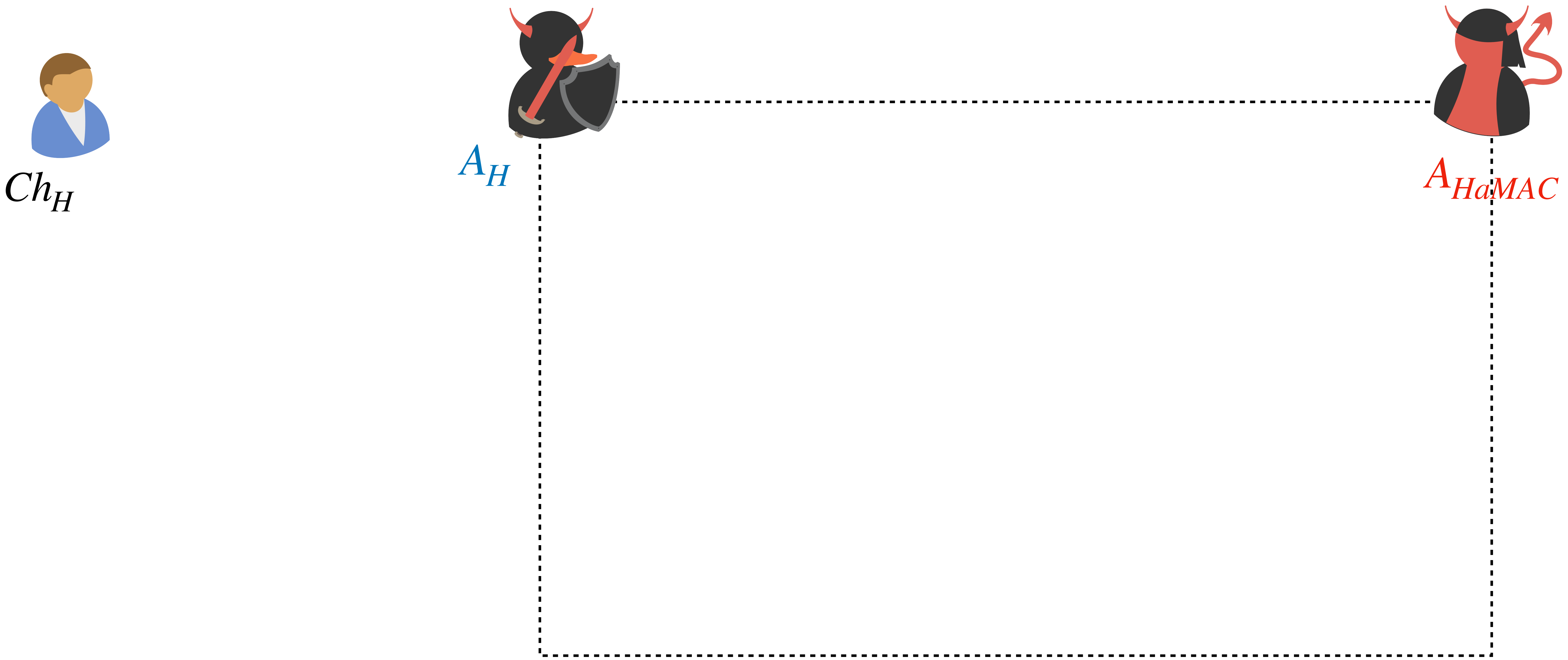
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

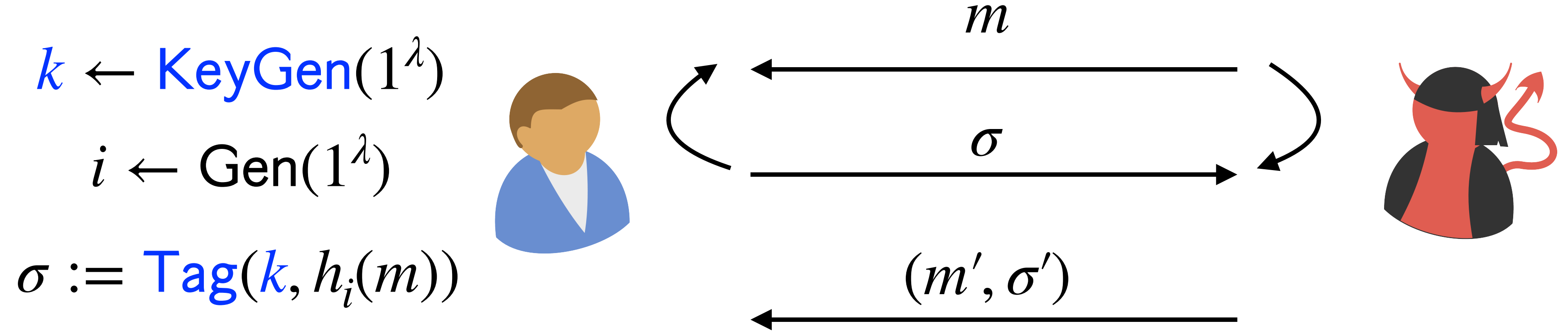
Claim: **Pr $\left[\text{col}\right] \leq$ negl(λ)**

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$



Proof of Security

H_0



Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

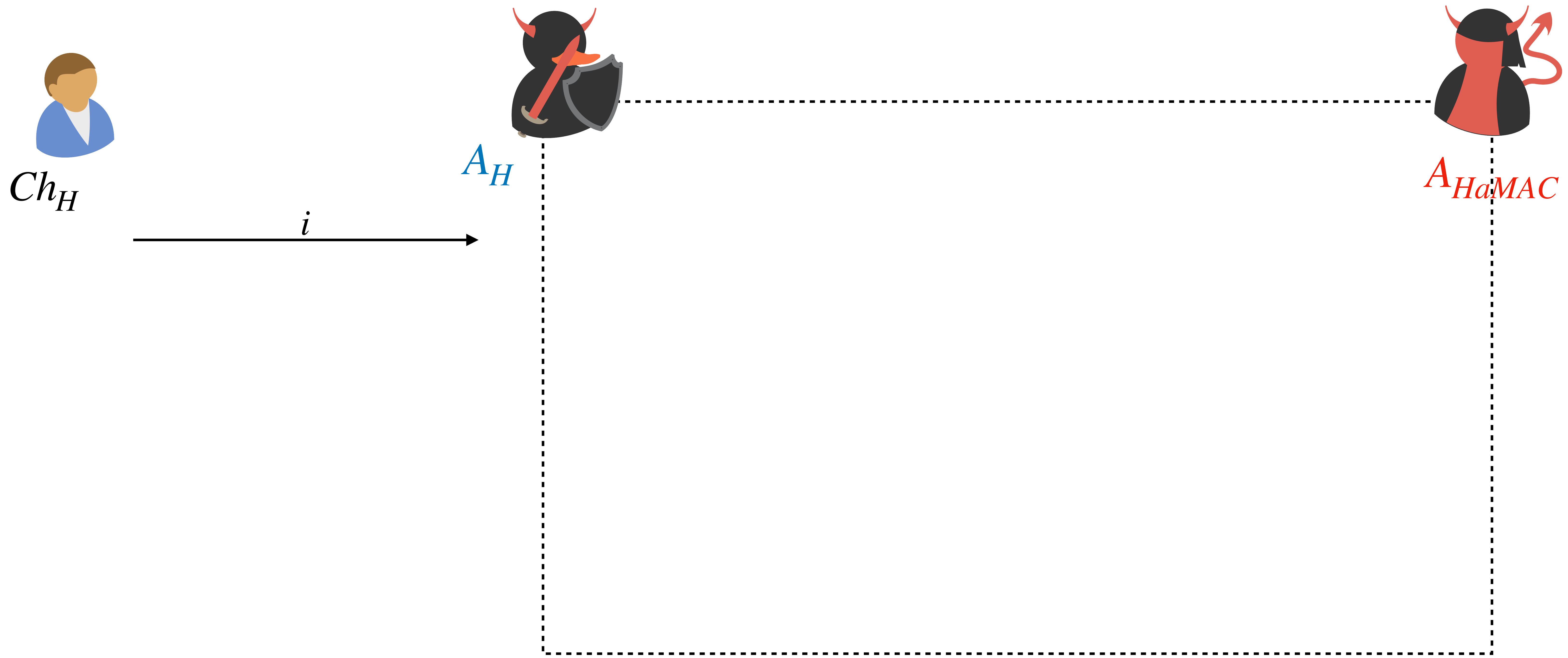
KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

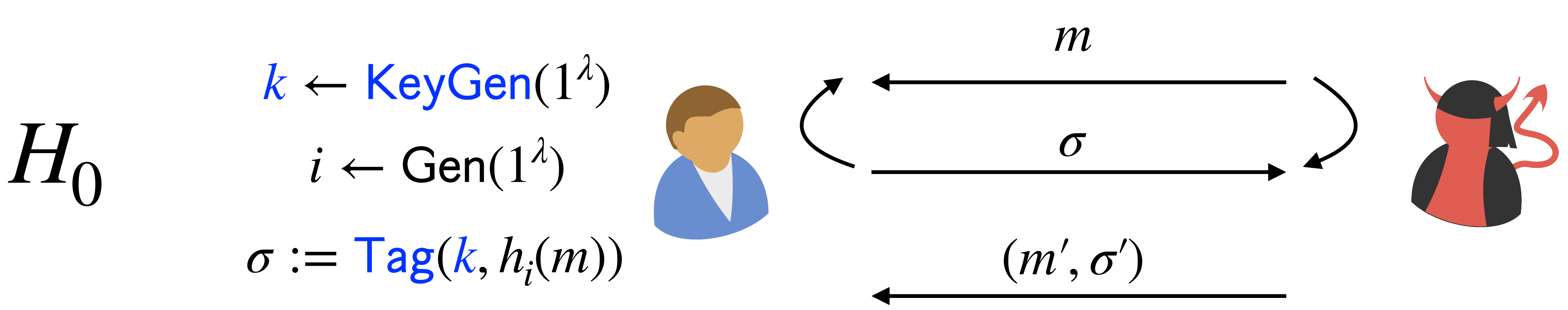
Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: **Pr $\left[\text{col}\right] \leq$ negl(λ)**



Proof of Security



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

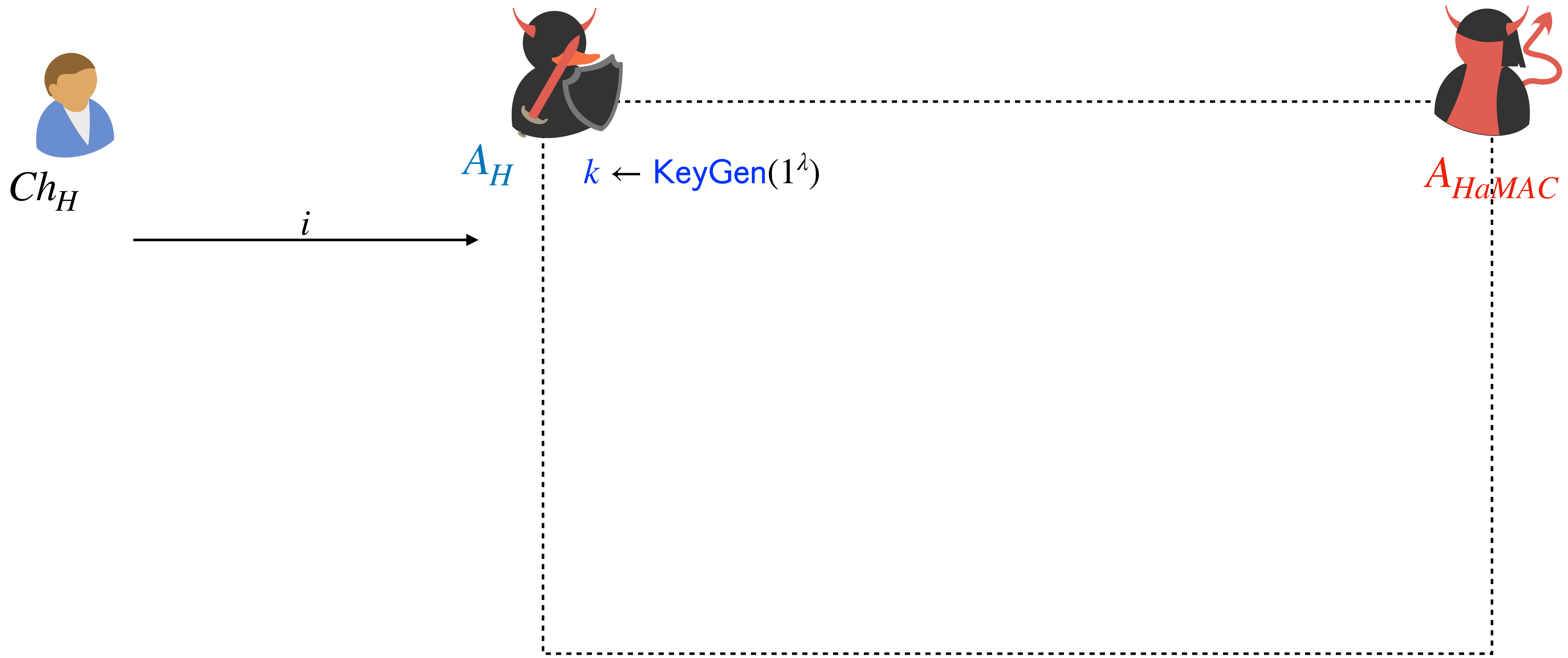
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\Pr\left[\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$



Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$

m

σ

(m', σ')

Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: **Pr $\left[\text{col}\right] \leq$ negl(λ)**

Ch_H

i

A_H

$k \leftarrow$ KeyGen(1^λ)

m

A_{HaMAC}

Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

Wins if $\mathsf{Ver}(k, h_i(m'), \sigma') = 1$ and $\mathscr{A}$ never queried m'

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\mathbf{Pr}\left[\mathbf{col}\right] \leq \mathbf{negl}(\lambda)$

Ch_H

i

A_H

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

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

A_{HaMAC}

Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$

m

σ

(m', σ')

Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: **Pr $\left[\text{col}\right] \leq$ negl(λ)**

Ch_H

i

A_H

$k \leftarrow$ KeyGen(1^λ)

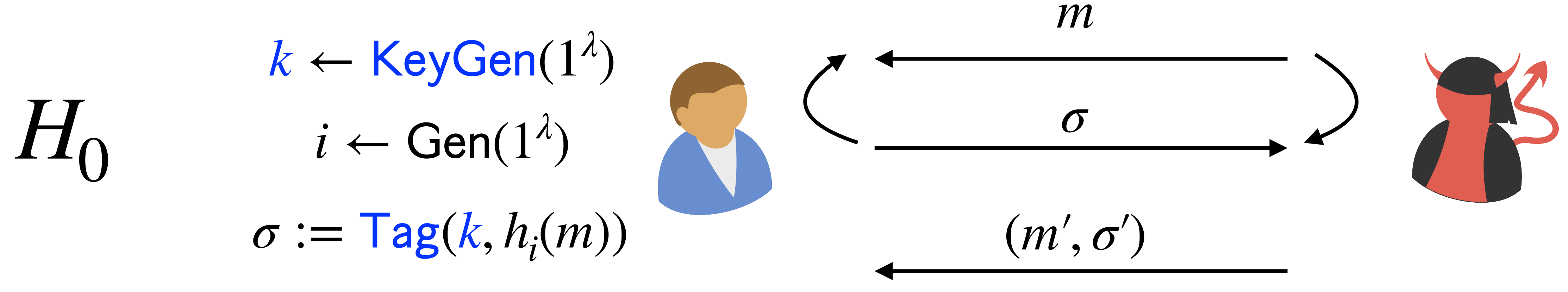
$\sigma :=$ Tag$(k, h_i(m))$

m

σ

A_{HaMAC}

Proof of Security



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

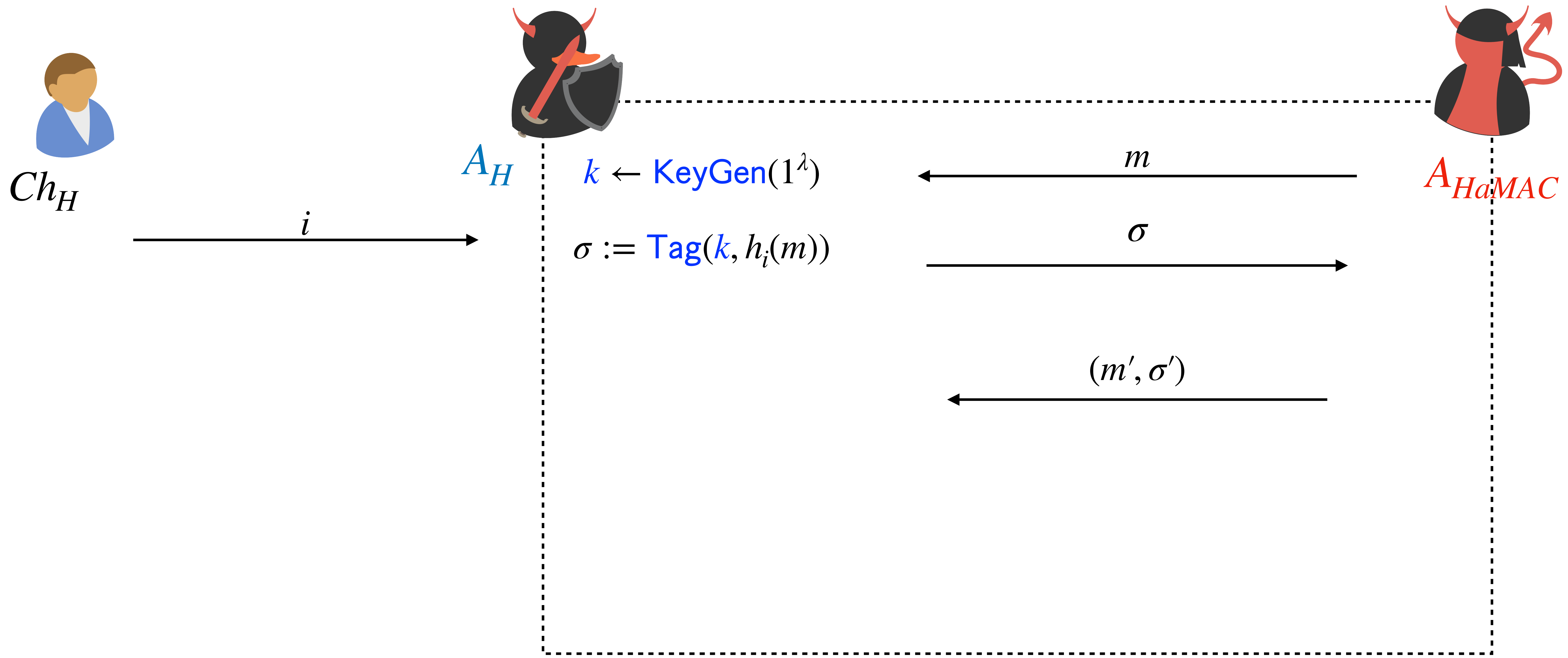
$\text{KeyGen}(1^\lambda) : (k \leftarrow \text{KeyGen}(1^\lambda), i)$

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\textbf{Pr}\left[\textbf{col}\right] \leq \textbf{negl}(\lambda)$



Proof of Security

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\mathbf{Pr}\left[\mathbf{col}\right] \leq \mathbf{negl}(\lambda)$

Ch_H

i

A_H

A_{HaMAC}

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

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

If $\exists m$ such that $h_i(m') = h_i(m)$ return (m, m')

Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

Wins if $\mathsf{Ver}(k, h_i(m'), \sigma') = 1$ and $\mathscr{A}$ never queried m'

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\Pr\left[\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$

Ch_H

i

A_H

A_{HaMAC}

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

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

If $\exists m$ such that $h_i(m') = h_i(m)$ return (m, m')

Else return $(0,0)$

Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

Wins if $\mathsf{Ver}(k, h_i(m'), \sigma') = 1$ and $\mathscr{A}$ never queried m'

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim: $\Pr[\mathbf{col}] \leq \mathsf{negl}(\lambda)$

Ch_H

A_H

A_{HaMAC}

i

(m, m')

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

$\sigma := \mathsf{Tag}(k, h_i(m))$

m

σ

(m', σ')

If $\exists m$ such that $h_i(m') = h_i(m)$ return (m, m')

Else return $(0,0)$

Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$

m ←

σ →

(m', σ') ←

Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Claim: Pr $\left[\text{col}\right] \leq$ negl(λ)

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Ch_H

i →

(m, m') ←

A_H

$k \leftarrow$ KeyGen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$

If $\exists m$ such that $h_i(m') = h_i(m)$ return (m, m')

Else return $(0,0)$

A_{HaMAC}

m ←

σ →

(m', σ') ←

Pr[col] = Pr[$\mathscr{A}_H$ wins] $\leq$ negl(λ)

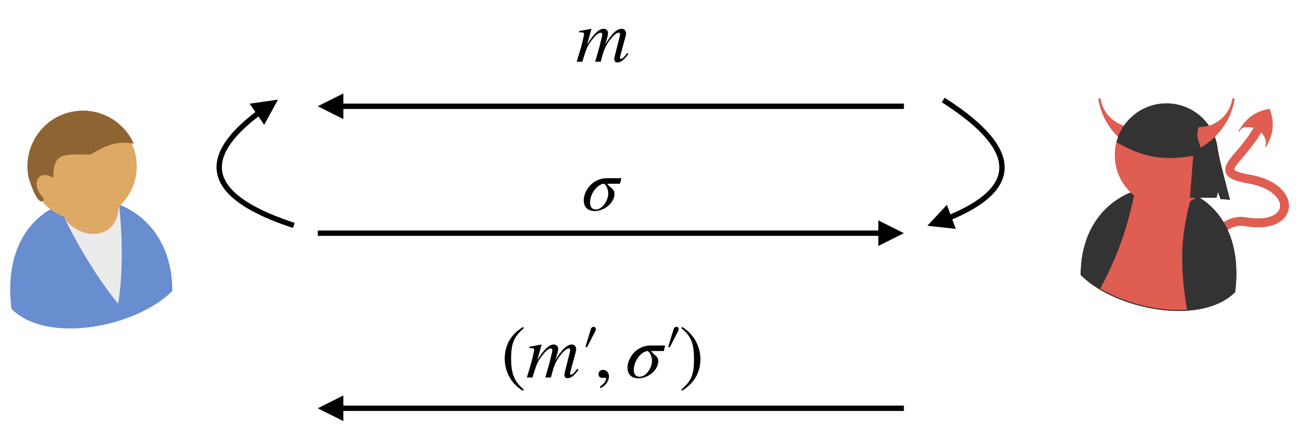
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

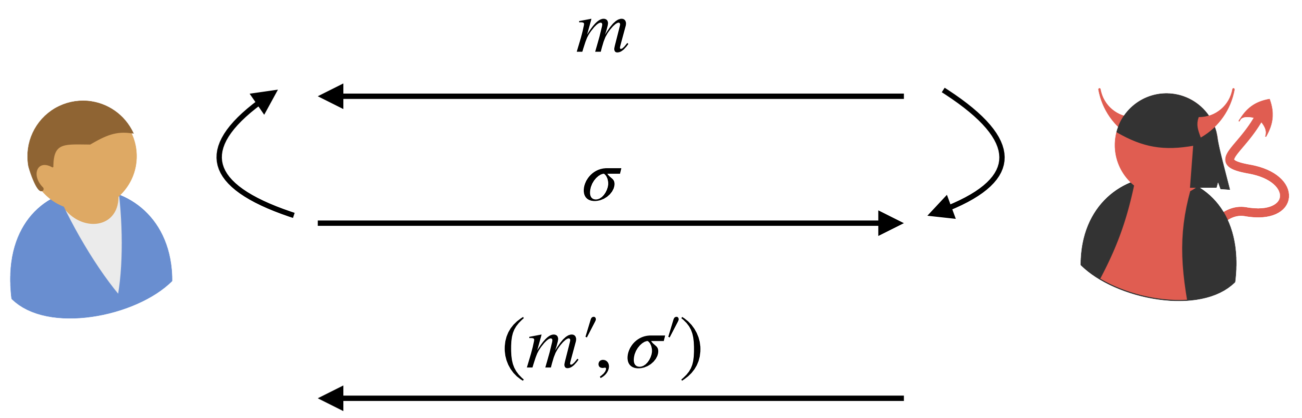
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$\Pr[\mathscr{A} \text{ wins in } H_0]$

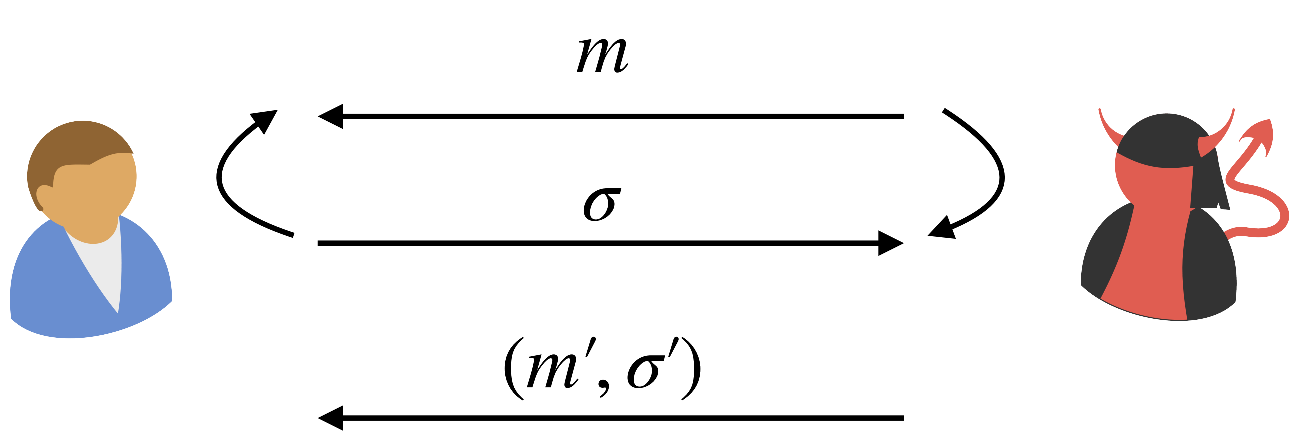
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr[\mathscr{A} \text{ wins in } H_0]$$

$$= \Pr[\mathscr{A} \text{ wins in } H_0 \mid \text{col}]\Pr[\text{col}] + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

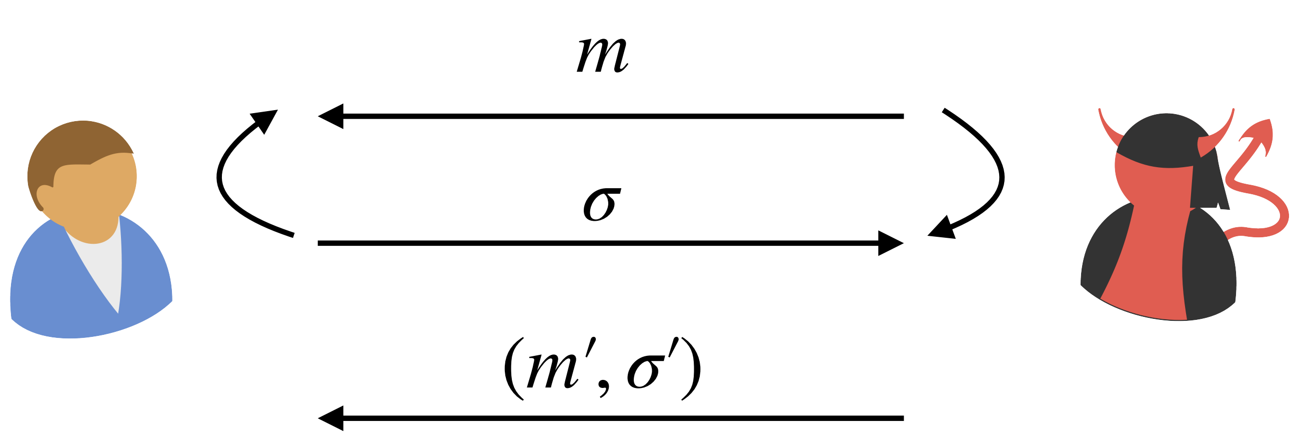
Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$



Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr[\mathscr{A} \text{ wins in } H_0]$$

$$= \Pr[\mathscr{A} \text{ wins in } H_0 \mid \text{col}]\Pr[\text{col}] + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

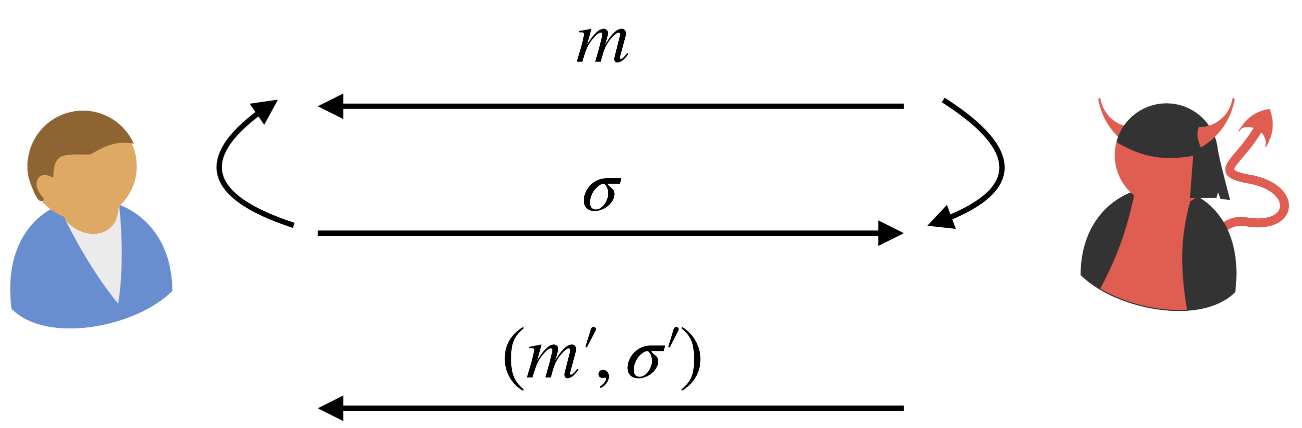
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr[\mathscr{A} \text{ wins in } H_0]$$

$$= \Pr[\mathscr{A} \text{ wins in } H_0 \mid \text{col}]\Pr[\text{col}] + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]$$

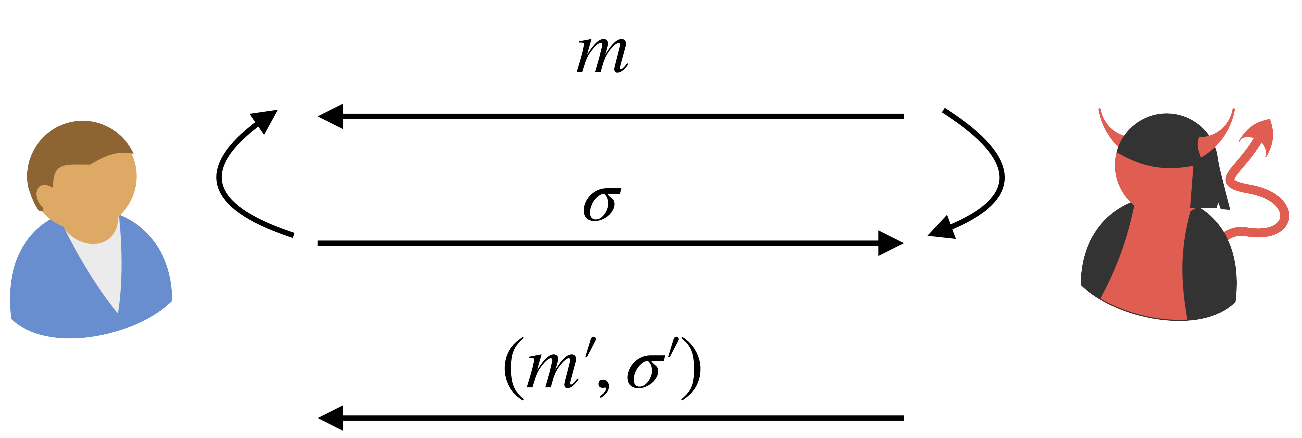
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}\right] \leq \text{negl}(\lambda)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr[\mathscr{A} \text{ wins in } H_0]$$

$$= \Pr[\mathscr{A} \text{ wins in } H_0 \mid \text{col}]\Pr[\text{col}] + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]$$

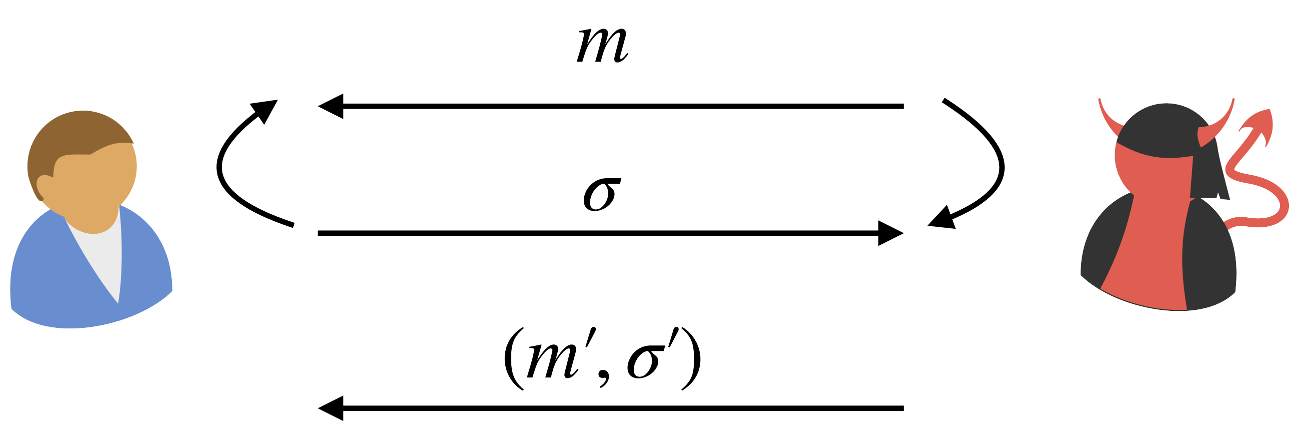
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}\right] \leq \text{negl}(\lambda)$

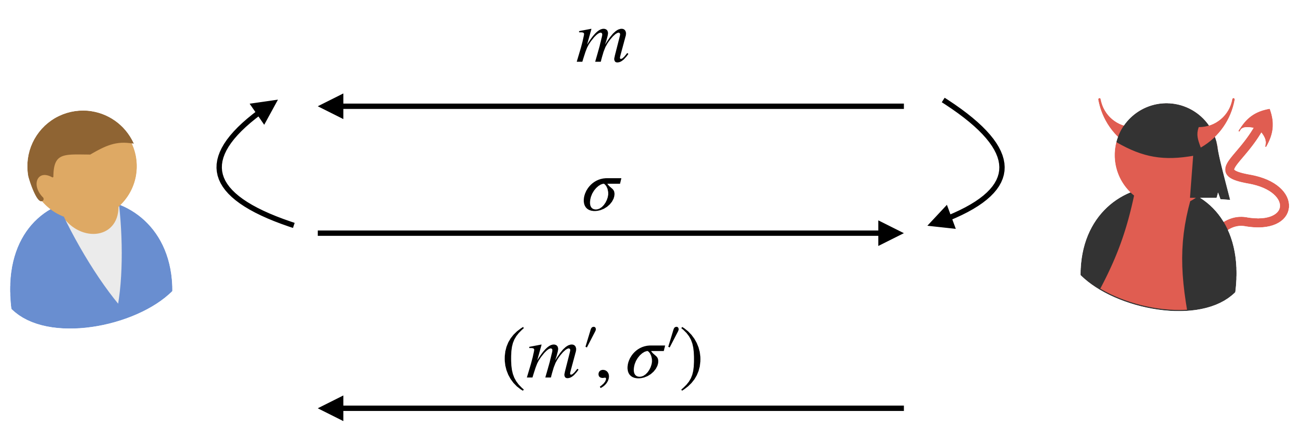
Proof of Security

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

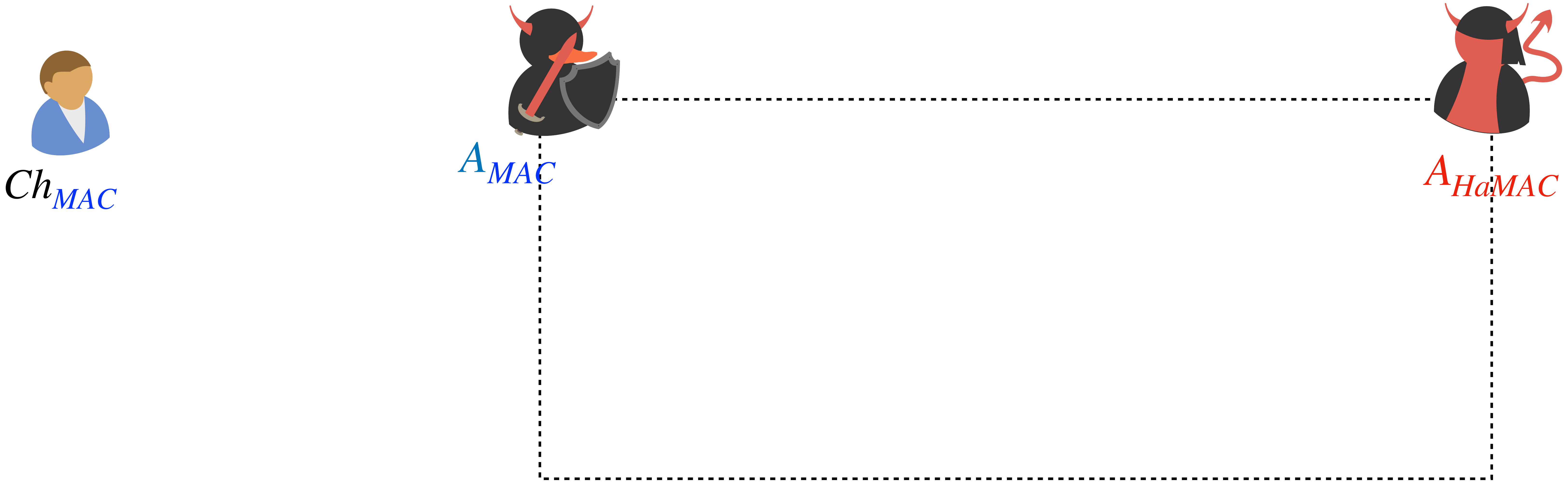
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg \textbf{col}\right] \leq \mathsf{negl}(\lambda)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$



Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$

m

σ

(m', σ')

Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Claim:
Pr $\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}\right] \leq \text{negl}(\lambda)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Ch_{MAC}

A_{MAC}

$i \leftarrow$ Gen(1^λ)

A_{HaMAC}

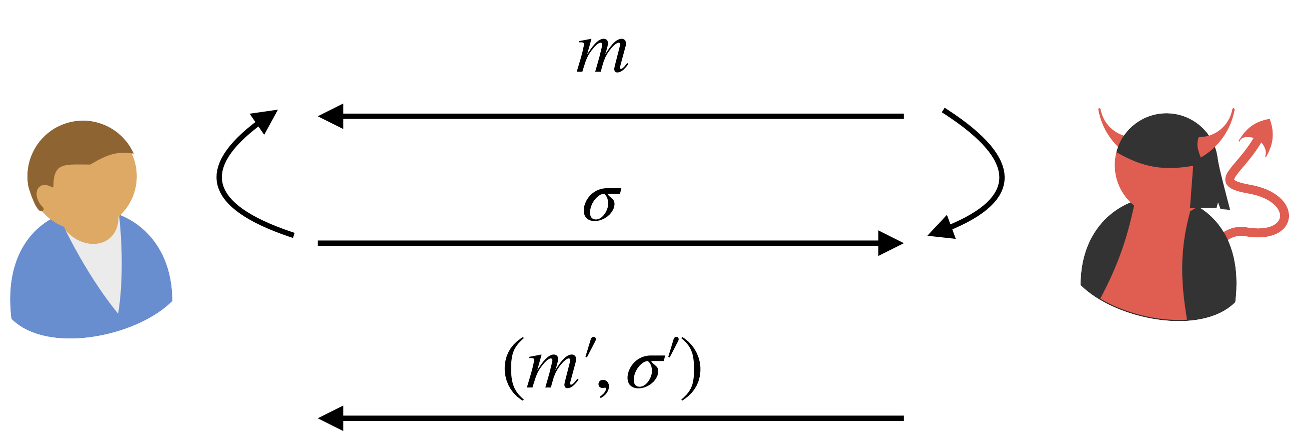
Proof of Security

H_0

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

$i \leftarrow \mathsf{Gen}(1^\lambda)$

$\sigma := \mathsf{Tag}(k, h_i(m))$



Wins if $\mathsf{Ver}(k, h_i(m'), \sigma') = 1$ and $\mathscr{A}$ never queried m'

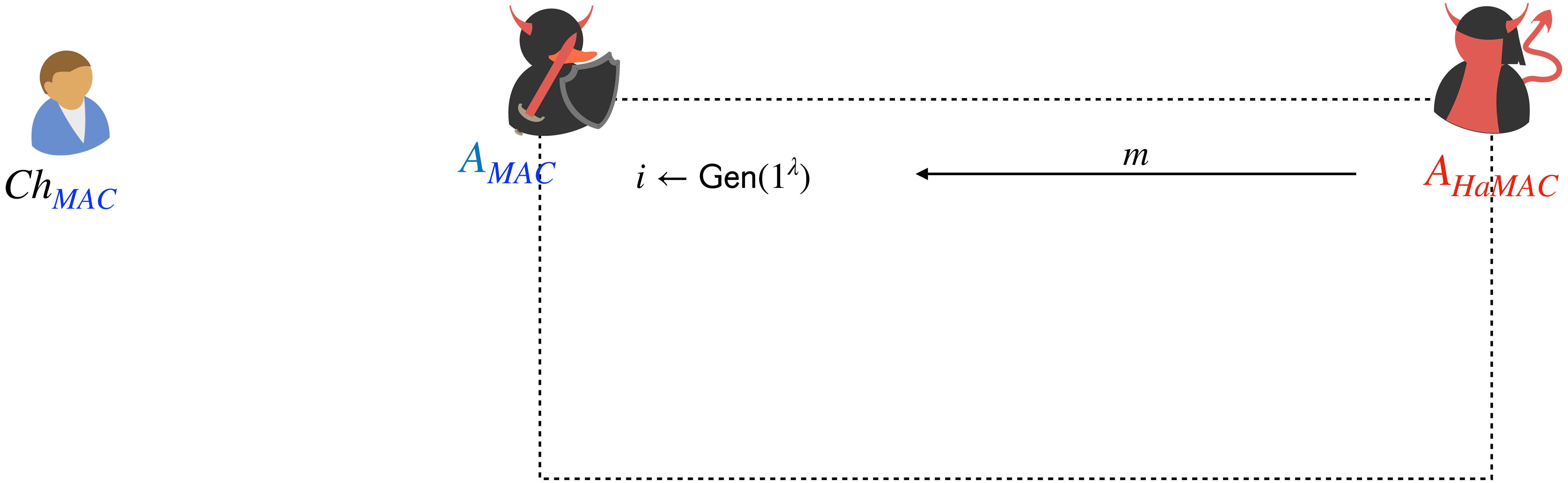
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$

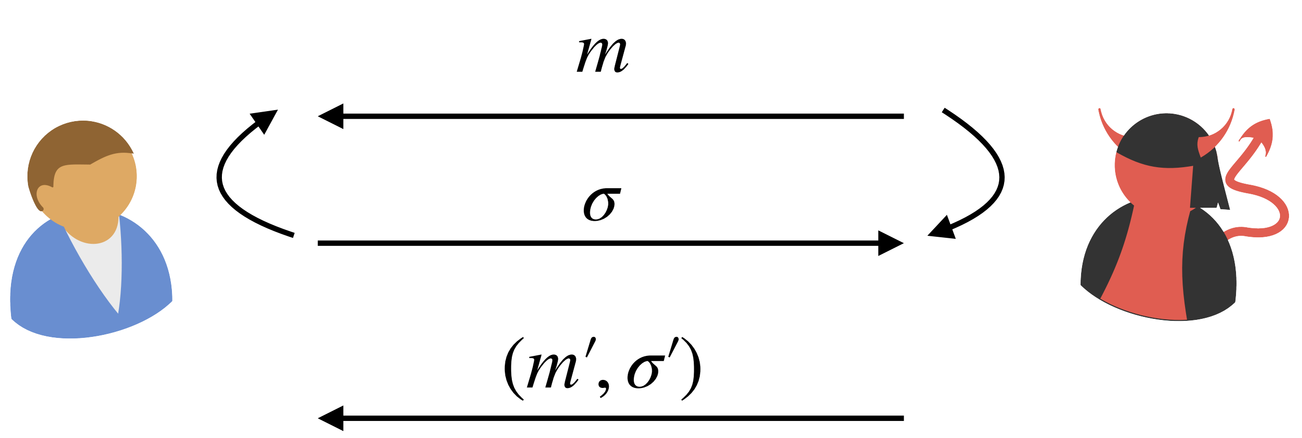
Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$



Proof of Security

H_0

$k \leftarrow \mathsf{KeyGen}(1^\lambda)$
$i \leftarrow \mathsf{Gen}(1^\lambda)$
$\sigma := \mathsf{Tag}(k, h_i(m))$



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

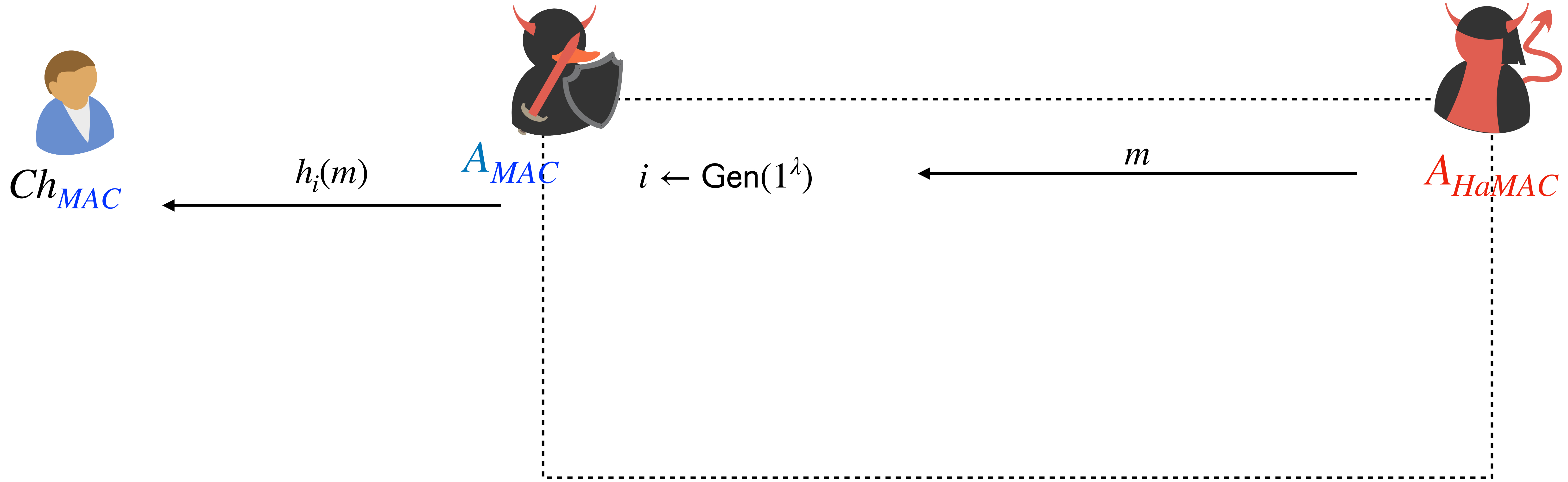
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$



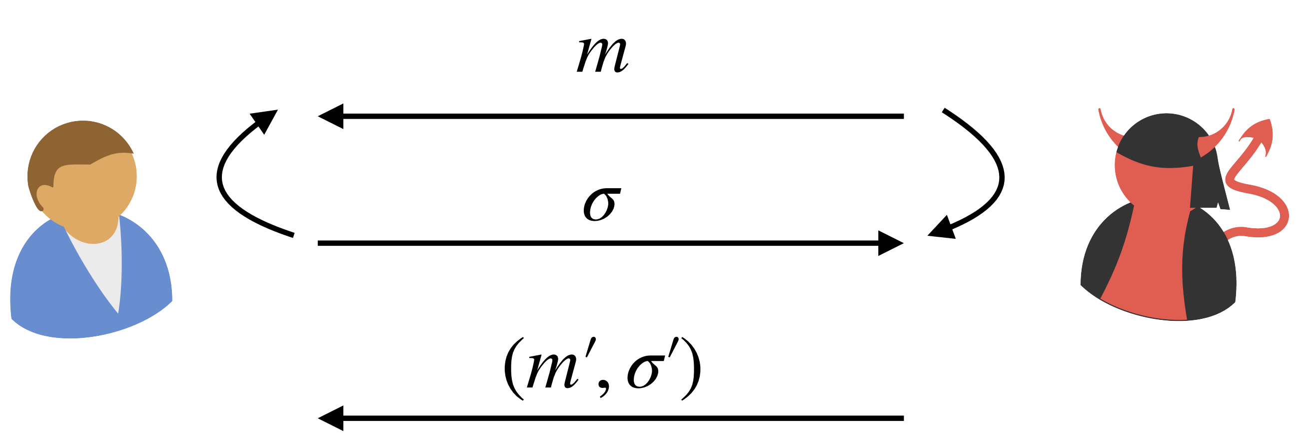
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

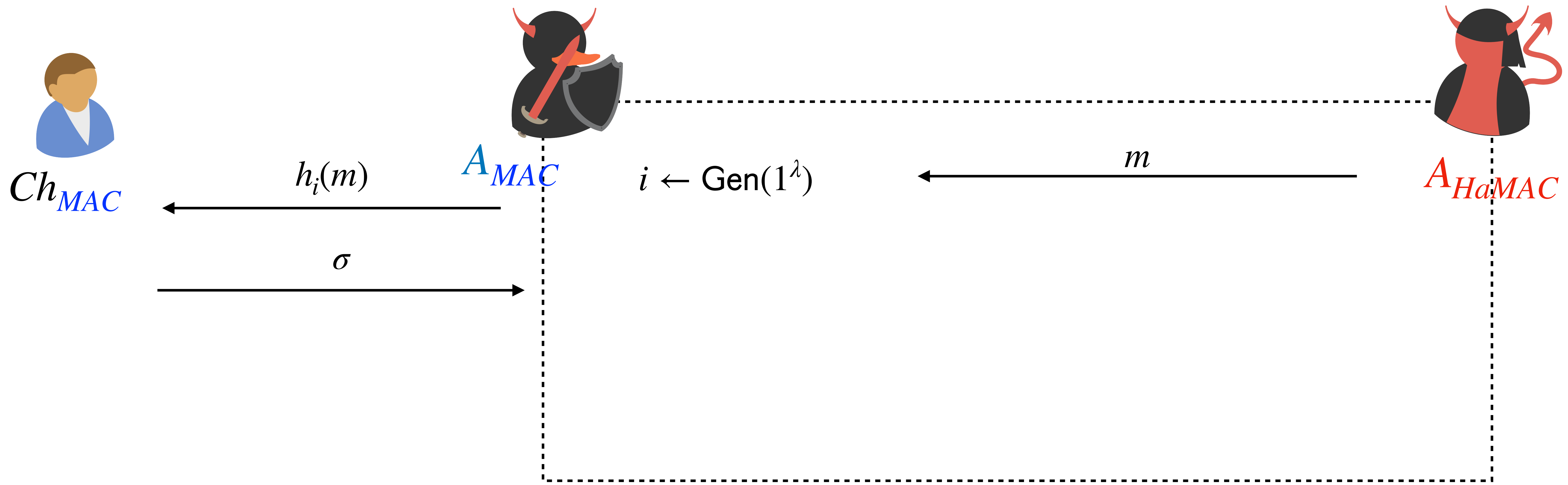
$\text{KeyGen}(1^\lambda) : (k \leftarrow \text{KeyGen}(1^\lambda), i)$

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

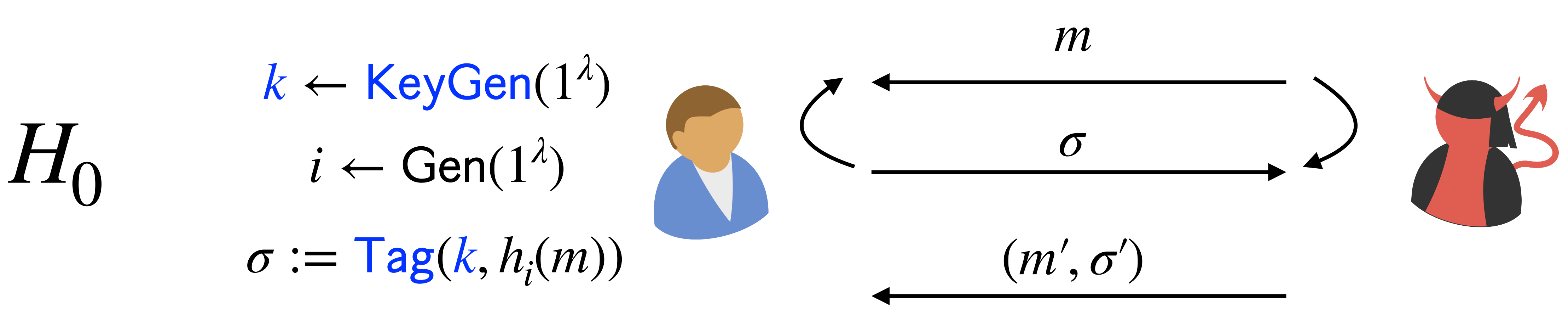
$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\textbf{col}\right] \leq \text{negl}(\lambda)$



Proof of Security



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never queried m'

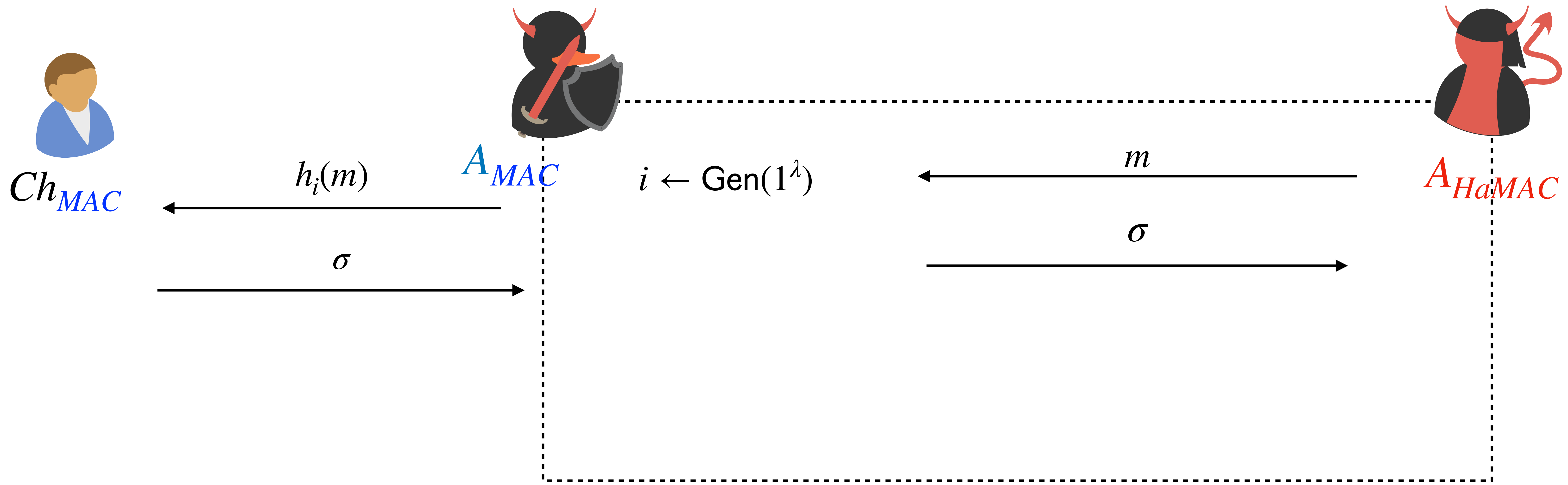
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

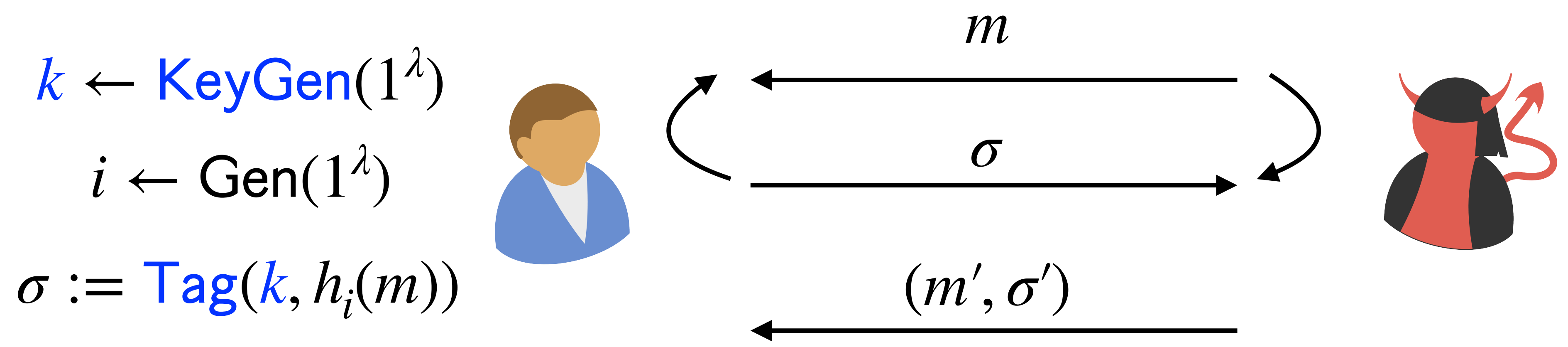
Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$



Proof of Security

H_0



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

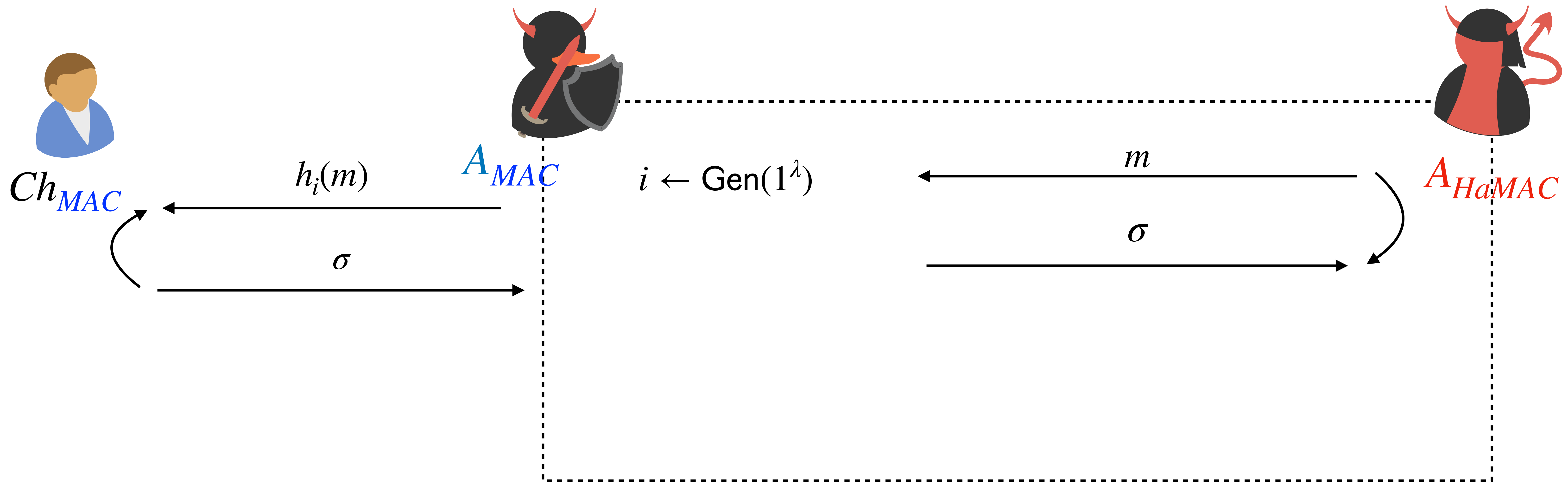
$\mathsf{KeyGen}(1^\lambda) : (k \leftarrow \mathsf{KeyGen}(1^\lambda), i)$

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$



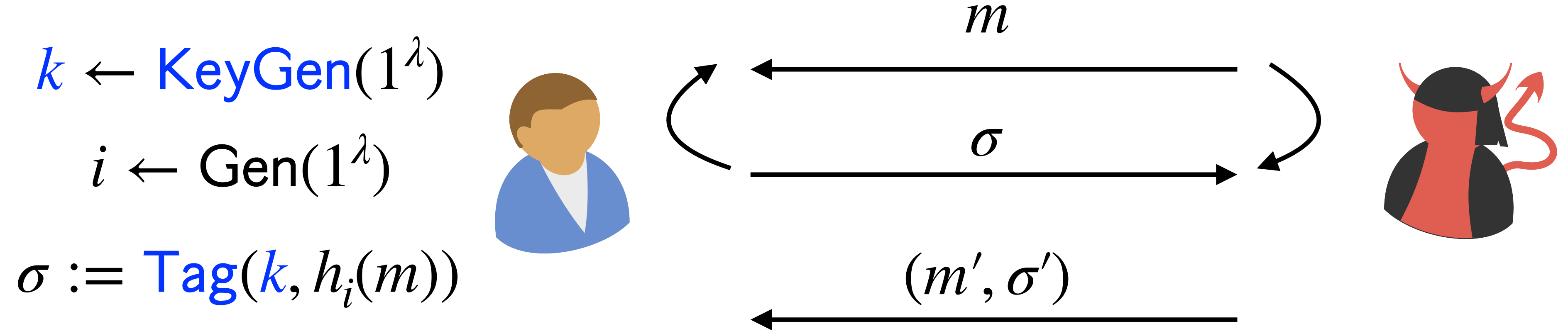
Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$

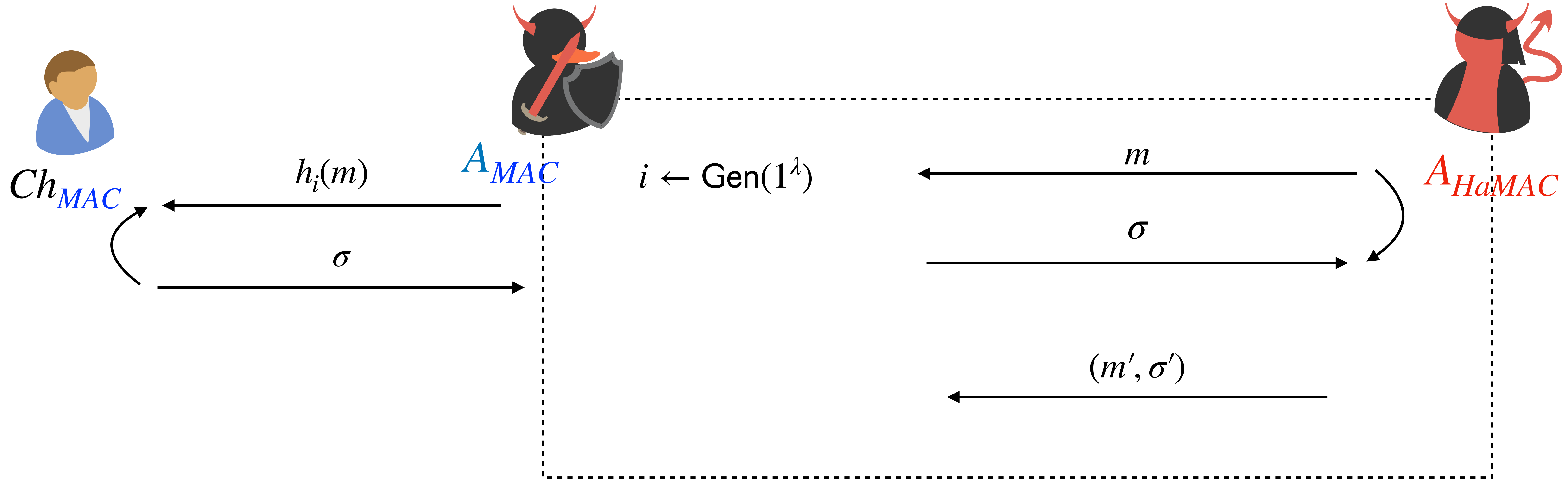
H_0



Wins if
$\mathsf{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$



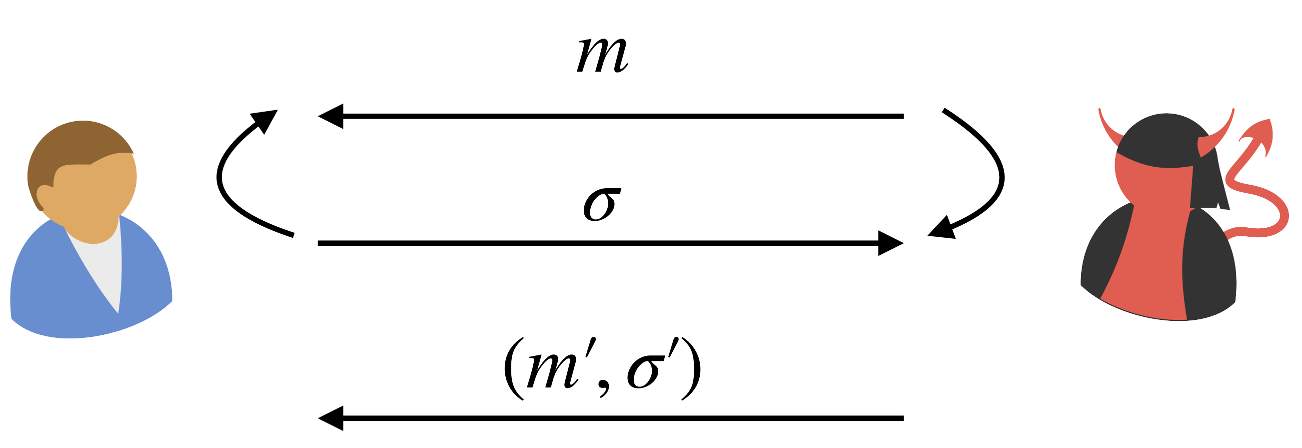
Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$



Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

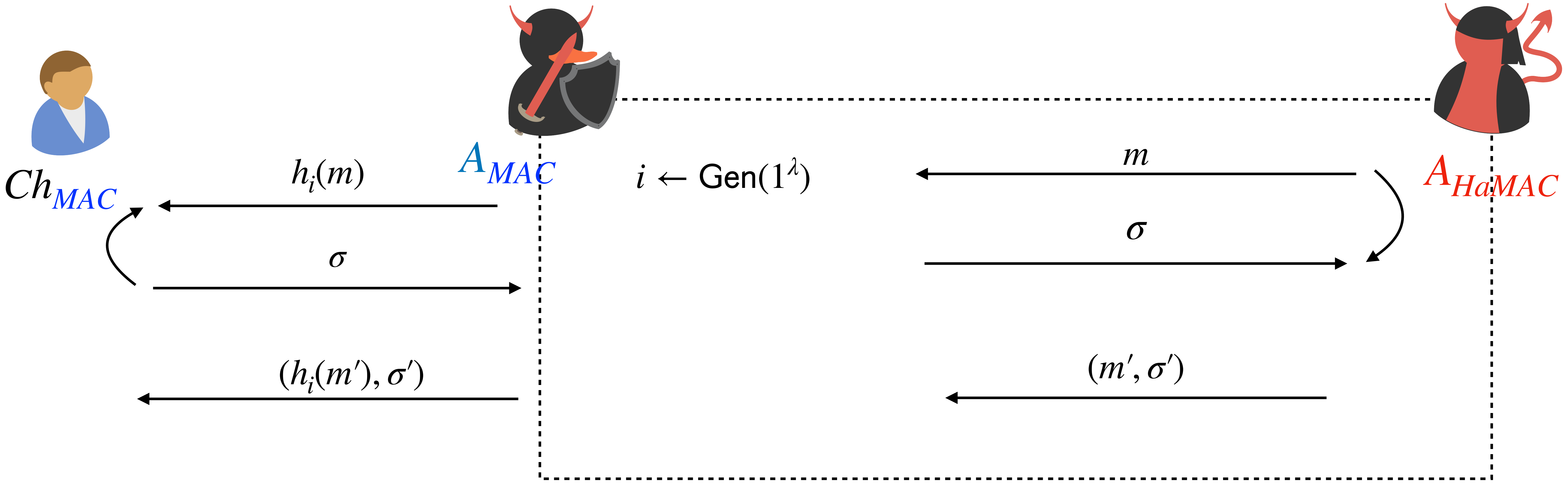
KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Claim:
Pr $\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\textbf{col}\right] \leq$ negl(λ)

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$

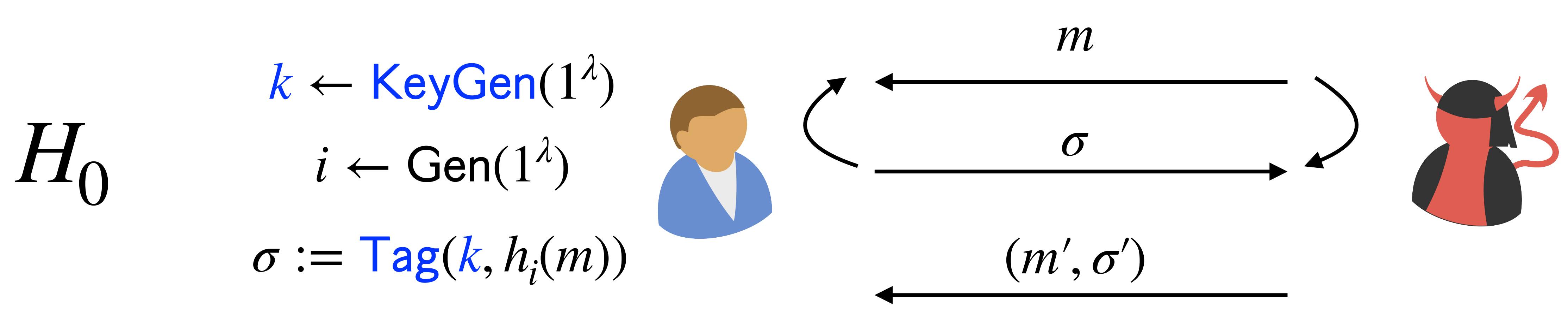


Proof of Security

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

$\mathsf{Tag}(k, m) : \sigma := \mathsf{Tag}(k, h_i(m))$

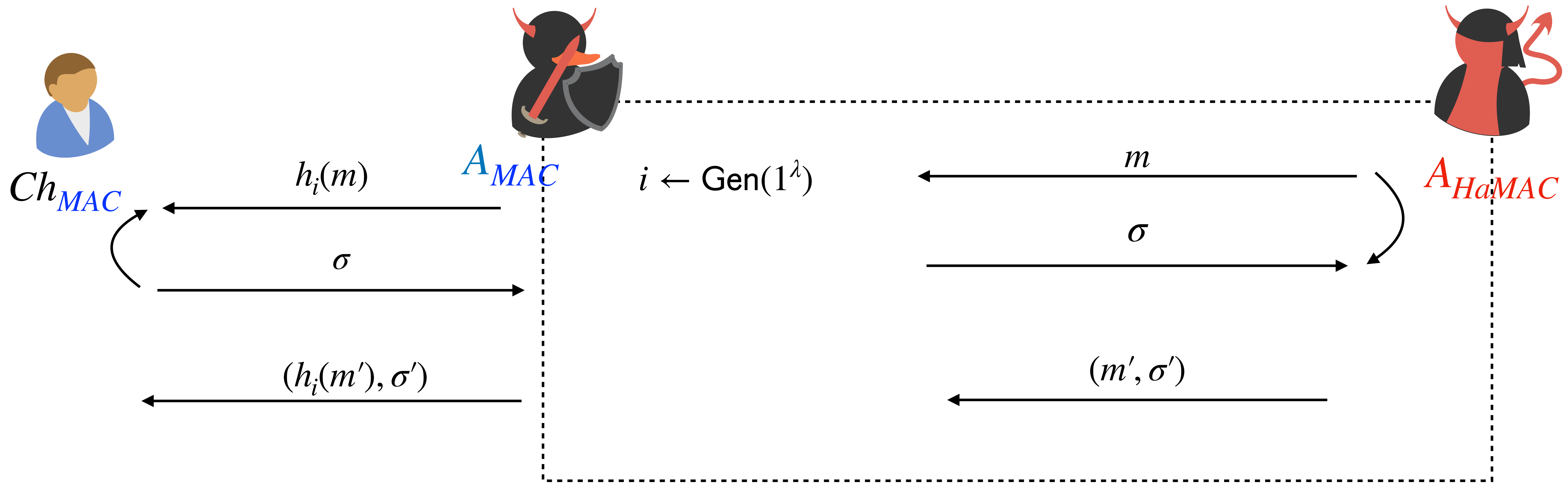
$\mathsf{Ver}(k, m, \sigma) : \mathsf{Ver}(k, h_i(m), \sigma)$



Wins if $\mathsf{Ver}(k, h_i(m'), \sigma') = 1$ and $\mathscr{A}$ never queried m'

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

Claim:
$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\mathsf{col}\right] \leq \mathsf{negl}(\lambda)$



$\Pr[\mathscr{A} \text{ wins} \mid \neg\mathsf{col}] = \Pr[\mathscr{A}_{MAC} \text{ wins}] \leq \mathsf{negl}(\lambda)$

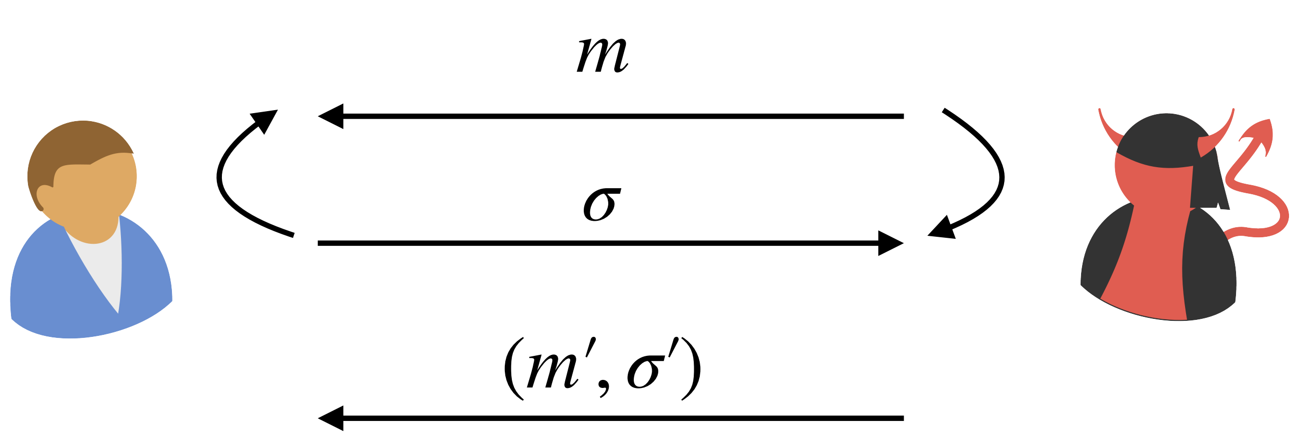
Proof of Security

H_0

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

$i \leftarrow \text{Gen}(1^\lambda)$

$\sigma := \text{Tag}(k, h_i(m))$



Wins if
$\text{Ver}(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

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

$\text{Tag}(k, m) : \sigma := \text{Tag}(k, h_i(m))$

$\text{Ver}(k, m, \sigma) : \text{Ver}(k, h_i(m), \sigma)$

Let **col** be the event that
$h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr[\mathscr{A} \text{ wins in } H_0]$$

$$= \Pr[\mathscr{A} \text{ wins in } H_0 \mid \text{col}]\Pr[\text{col}] + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$$

$$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]$$

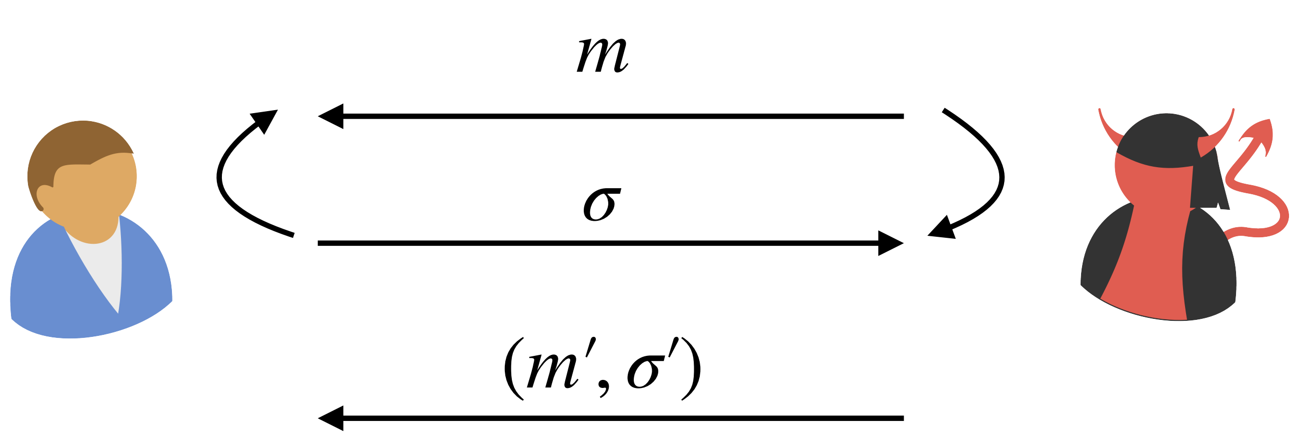
Proof of Security

H_0

$k \leftarrow$ KeyGen(1^λ)

$i \leftarrow$ Gen(1^λ)

$\sigma :=$ Tag$(k, h_i(m))$



Wins if
Ver$(k, h_i(m'), \sigma') = 1$
and $\mathscr{A}$ never
queried m'

KeyGen(1^λ) : $(k \leftarrow$ KeyGen$(1^\lambda), i)$

Tag(k, m) : $\sigma :=$ Tag$(k, h_i(m))$

Ver(k, m, σ) : Ver$(k, h_i(m), \sigma)$

Let **col** be the event that $h(m') = h(m)$ for some $m \neq m'$

$$\Pr\left[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}\right] \leq \text{negl}(\lambda)$$

$$\Pr\left[\text{col}\right] \leq \text{negl}(\lambda)$$

$\Pr[\mathscr{A}$ wins in $H_0]$

$= \Pr[\mathscr{A} \text{ wins in } H_0 \mid \text{col}]\Pr[\text{col}] + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$

$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]\Pr[\neg\text{col}]$

$\leq \text{negl}(\lambda) + \Pr[\mathscr{A} \text{ wins in } H_0 \mid \neg\text{col}]$

$\leq \text{negl}(\lambda) + \text{negl}(\lambda)$