CCA Security II

601.442/642 Modern Cryptography

14th April 2026

Logistics

Logistics

- HW 8 due Thursday

Logistics

- HW 8 due Thursday
- Please do your course evaluations! This is our first time teaching the course so feedback is extremely valuable

Recap: IND-CCA Security

Recap: IND-CCA Security

- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can’t learn *anything* about ciphertexts

Recap: IND-CCA Security

- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.

Recap: IND-CCA Security

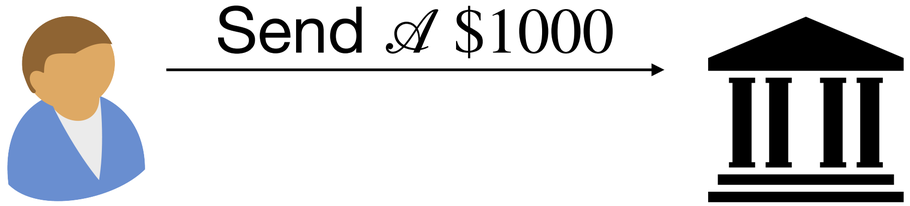
- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.
- Further motivation:

Recap: IND-CCA Security

- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.
- Further motivation:

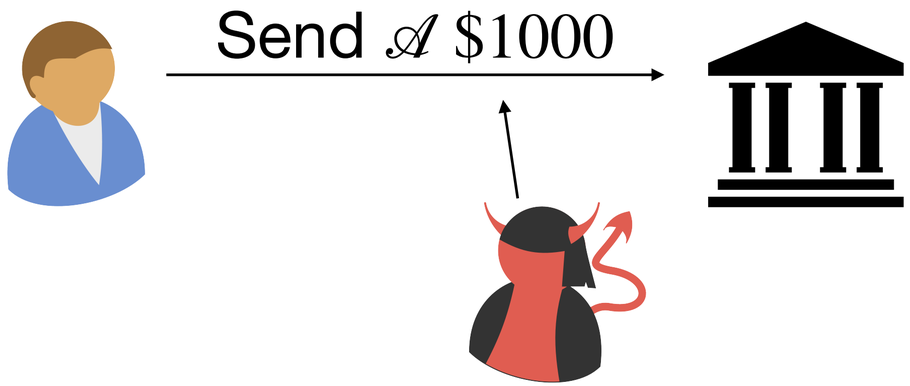
Recap: IND-CCA Security

- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.
- Further motivation:



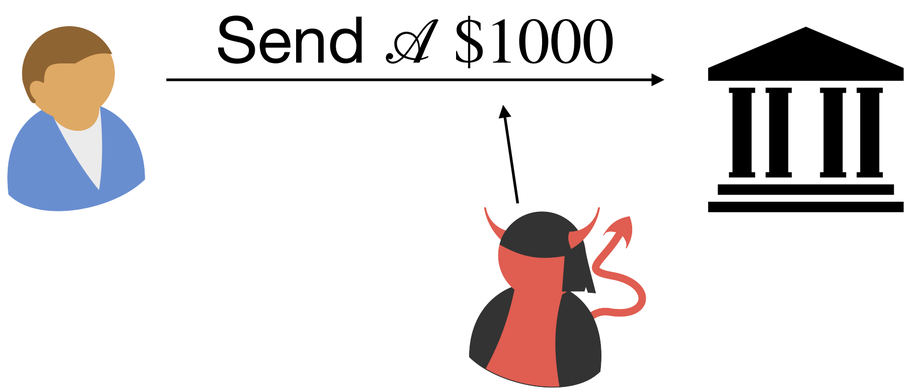
Recap: IND-CCA Security

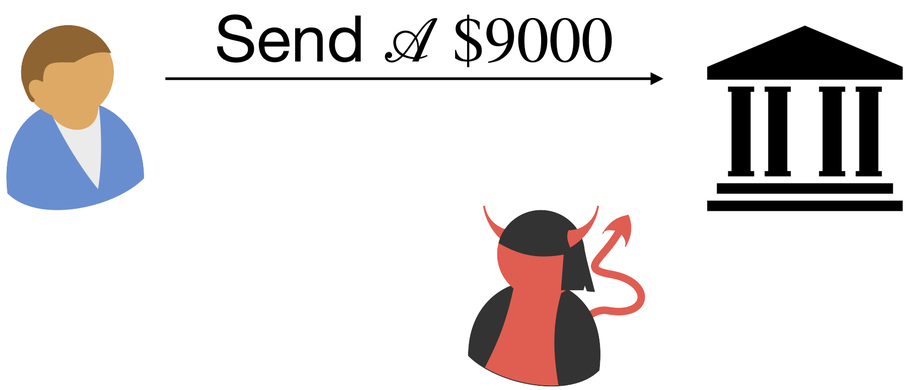
- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.
- Further motivation:



Recap: IND-CCA Security

- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.
- Further motivation:

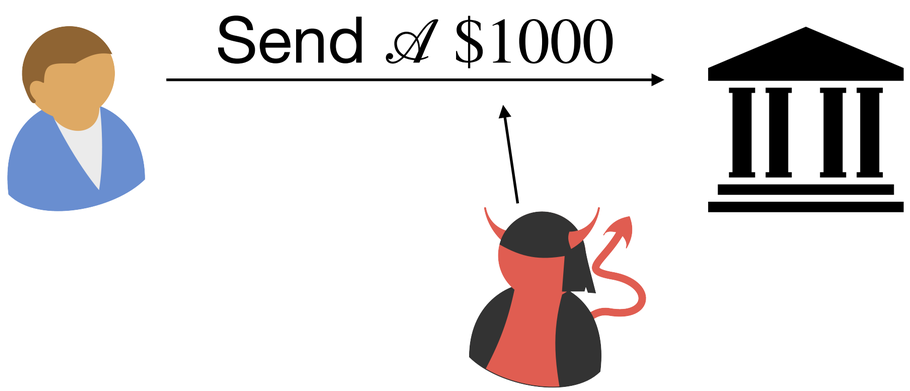


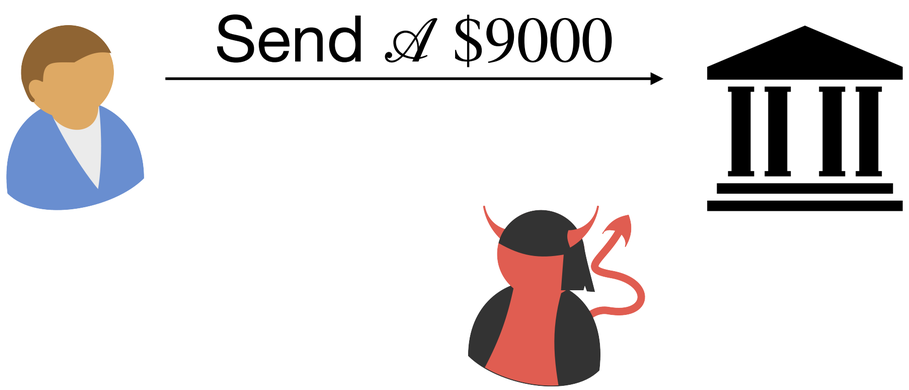


Recap: IND-CCA Security

- For real-world protocols, IND-CPA security is too weak. It only guarantees security if the adversary can't learn *anything* about ciphertexts
- So, we describe a *stronger* security property that allows adversaries to learn the decryptions of ciphertexts.
- Further motivation:

Adversary can't *read* the ciphertext, but can edit it!





Recap: NIZKs

NIZKs

Let $L \subset \{0,1\}^*$ be a language.

- $\mathsf{CRSGen}(1^\lambda) \to \mathsf{crs}$
- $P(x, \mathsf{crs}) \to \pi$
- $V(\pi, \mathsf{crs}) \to \{0,1\}$

Completeness: for all $x \in L$ and for all P

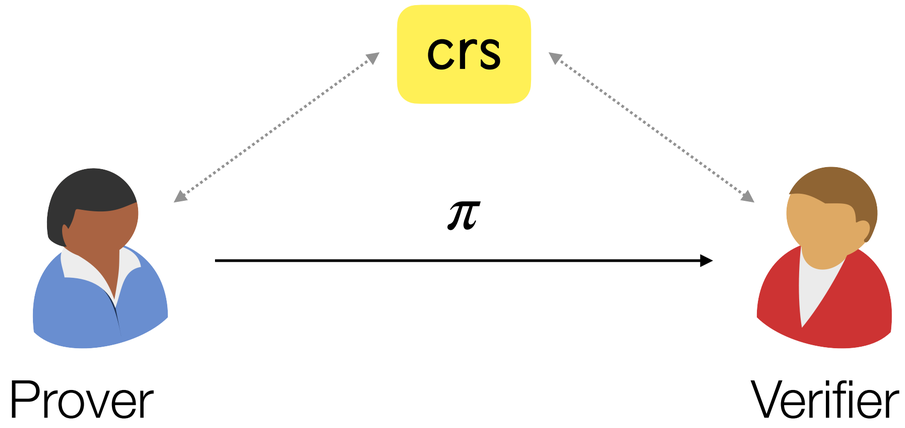
$$\Pr\left[V(\pi) = 1 : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow P(x, \mathsf{crs})\end{matrix}\right] = 1$$

Soundness: for all $x \notin L$ and for all $\hat{P}$

$$\Pr\left[V(\pi, \mathsf{crs}) : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow \hat{P}(x, \mathsf{crs})\end{matrix}\right] \leq \mathsf{negl}(|x|)$$

Zero Knowledge: for all PPT $\hat{V}$, there exists a PPT simulator S such that for all $x \in L$ and $z \in \{0,1\}^*$

$$\left\{\mathsf{View}_{\hat{V}}\left[P(x, \mathsf{crs}) \leftrightarrow \hat{V}(x, \mathsf{crs}, z)\right]\right\} \overset{c}{\approx} \{S(x, z)\}$$



Recap: NIZKs

NIZKs

Let $L \subset \{0,1\}^*$ be a language.

- $\mathsf{CRSGen}(1^\lambda) \to \mathsf{crs}$
- $P(x, \mathsf{crs}) \to \pi$
- $V(\pi, \mathsf{crs}) \to \{0,1\}$

Completeness: for all $x \in L$ and for all P

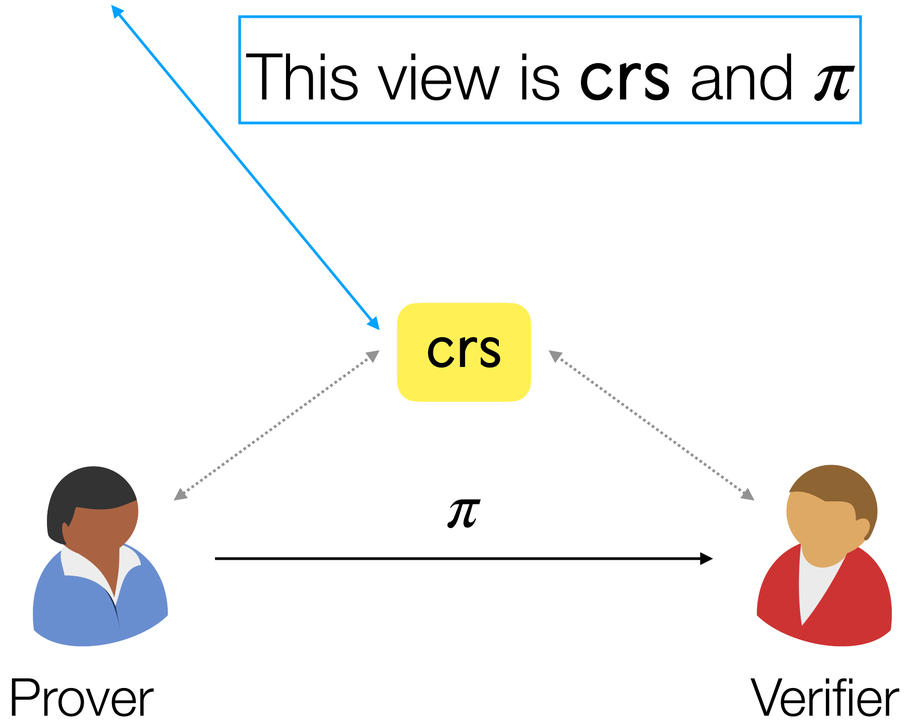
$$\Pr\left[V(\pi) = 1 : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow P(x, \mathsf{crs})\end{matrix}\right] = 1$$

Soundness: for all $x \notin L$ and for all $\hat{P}$

$$\Pr\left[V(\pi, \mathsf{crs}) : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow \hat{P}(x, \mathsf{crs})\end{matrix}\right] \leq \mathsf{negl}(|x|)$$

Zero Knowledge: for all PPT $\hat{V}$, there exists a PPT simulator S such that for all $x \in L$ and $z \in \{0,1\}^*$

$$\left\{\mathsf{View}_{\hat{V}}\left[P(x, \mathsf{crs}) \leftrightarrow \hat{V}(x, \mathsf{crs}, z)\right]\right\} \overset{c}{\approx} \{S(x, z)\}$$



Recap: NIZKs

Let $L \subset \{0,1\}^*$ be a language.

- $\mathsf{CRSGen}(1^\lambda) \to \mathsf{crs}$
- $P(x, \mathsf{crs}) \to \pi$
- $V(\pi, \mathsf{crs}) \to \{0,1\}$

Completeness: for all $x \in L$ and for all P

$$\Pr\left[V(\pi) = 1 : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow P(x, \mathsf{crs})\end{matrix}\right] = 1$$

Soundness: for all $x \notin L$ and for all $\hat{P}$

$$\Pr\left[V(\pi, \mathsf{crs}) : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow \hat{P}(x, \mathsf{crs})\end{matrix}\right] \leq \mathsf{negl}(|x|)$$

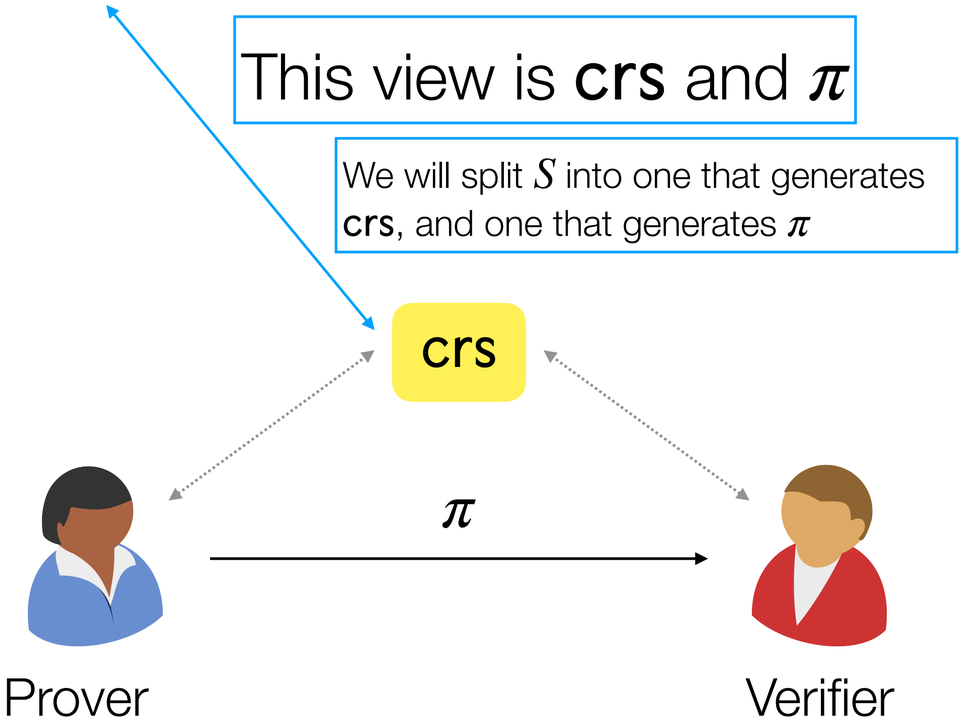
NIZKs

Zero Knowledge: for all PPT $\hat{V}$, there exists a PPT simulator S such that for all $x \in L$ and $z \in \{0,1\}^*$

$$\left\{\mathsf{View}_{\hat{V}}\left[P(x, \mathsf{crs}) \leftrightarrow \hat{V}(x, \mathsf{crs}, z)\right]\right\} \overset{c}{\approx} \{S(x, z)\}$$

This view is **crs** and π

We will split S into one that generates **crs**, and one that generates π



Re NIZK

Says that simulators can always make proofs for *true* statements.

Doesn't say they *can't* make proofs for false statements, just that we can't guarantee it.

NIZKs

Zero Knowledge: for all PPT $\hat{V}$, there exists a PPT simulator S such that for all $x \in L$ and $z \in \{0,1\}^*$

$$\left\{ \text{View}_{\hat{V}} \left[P(x, \text{crs}) \leftrightarrow \hat{V}(x, \text{crs}, z) \right] \right\} \overset{c}{\approx} \{S(x, z)\}$$

This view is **crs** and π

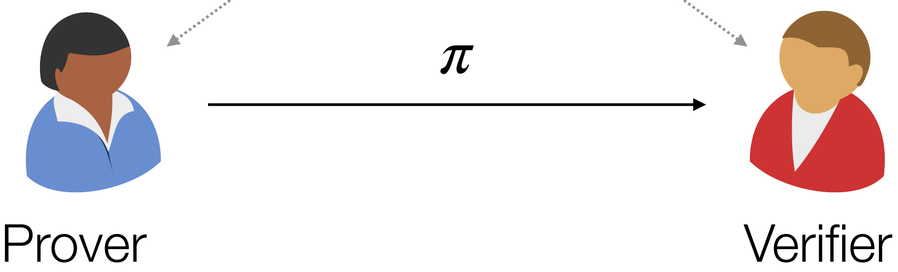
We will split S into one that generates **crs**, and one that generates π

Completeness: for all $x \in L$ and for all P

$$\Pr\left[V(\pi) = 1 : \begin{array}{c} \text{crs} \leftarrow \text{CRSGen}(1^\lambda) \\ \pi \leftarrow P(x, \text{crs}) \end{array} \right] = 1$$

Soundness: for all $x \notin L$ and for all $\hat{P}$

$$\Pr\left[V(\pi, \text{crs}) : \begin{array}{c} \text{crs} \leftarrow \text{CRSGen}(1^\lambda) \\ \pi \leftarrow \hat{P}(x, \text{crs}) \end{array} \right] \leq \text{negl}(|x|)$$



How to Use Proofs in Crypto Protocols

Why are they useful?

How to Use Proofs in Crypto Protocols

Why are they useful?

- Some protocols are only secure if parties can’t *lie* about what they’re doing

How to Use Proofs in Crypto Protocols

Why are they useful?

- Some protocols are only secure if parties can't *lie* about what they're doing
  - e.g. "This ciphertext is a valid encryption of something"

How to Use Proofs in Crypto Protocols

Why are they useful?

- Some protocols are only secure if parties can't *lie* about what they're doing
  - e.g. "This ciphertext is a valid encryption of something"
- If we force parties to add a *proof* that they did something correctly, we don't have to worry about them lying.

How to Use Proofs in Crypto Protocols

Why are they useful?

- Some protocols are only secure if parties can't *lie* about what they're doing
  - e.g. "This ciphertext is a valid encryption of something"
- If we force parties to add a *proof* that they did something correctly, we don't have to worry about them lying.
- Sometimes they might be able to lie about something that should be *secret*

How to Use Proofs in Crypto Protocols

Why are they useful?

- Some protocols are only secure if parties can't *lie* about what they're doing
  - e.g. "This ciphertext is a valid encryption of something"
- If we force parties to add a *proof* that they did something correctly, we don't have to worry about them lying.
- Sometimes they might be able to lie about something that should be *secret*
  - e.g. "This ciphertext is a valid encryption of my secret"

How to Use Proofs in Crypto Protocols

Why are they useful?

- Some protocols are only secure if parties can't *lie* about what they're doing
  - e.g. "This ciphertext is a valid encryption of something"
- If we force parties to add a *proof* that they did something correctly, we don't have to worry about them lying.
- Sometimes they might be able to lie about something that should be *secret*
  - e.g. "This ciphertext is a valid encryption of my secret"
- In this case we will use *zero-knowledge* proofs. Then the secret isn't leaked!

Proof Example: “This is a Valid Encryption”

Proof Example: “This is a Valid Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$

Proof Example: “This is a Valid Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - I give you a ciphertext **ct**. I then prove to you that this is a *valid ciphertext*

Proof Example: “This is a Valid Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - I give you a ciphertext **ct**. I then prove to you that this is a *valid ciphertext*
  - What does that mean? I could have constructed it using **Enc**

Proof Example: “This is a Valid Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - I give you a ciphertext **ct**. I then prove to you that this is a *valid ciphertext*
  - What does that mean? I could have constructed it using **Enc**
- The prover P generates this proof while knowing (k, m, r). This makes it easy.

Proof Example: “This is a Valid Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - I give you a ciphertext **ct**. I then prove to you that this is a *valid ciphertext*
  - What does that mean? I could have constructed it using **Enc**
- The prover P generates this proof while knowing (k, m, r). This makes it easy.
- The simulator S generates it *without knowing* (k, m, r)*!*

Proof Example: “This is a V

Specifying the randomness used.

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - I give you a ciphertext **ct**. I then prove to you that this is a *valid ciphertext*
  - What does that mean? I could have constructed it using **Enc**
- The prover P generates this proof while knowing (k, m, r). This makes it easy.
- The simulator S generates it *without knowing* (k, m, r)*!*

Proof Example: “This is a Valid Public Key Encryption”

Proof Example: “This is a Valid Public Key Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$

Proof Example: “This is a Valid Public Key Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - In a proof, the *statement* (x in the formal definition, the thing on the left side of the language definition here) contains all the *public* stuff, i.e. everything not secret.

Proof Example: “This is a Valid Public Key Encryption”

- $L = \{\mathsf{ct} \ : \ \exists k, m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(k, m; r)\}$
  - In a proof, the *statement* (x in the formal definition, the thing on the left side of the language definition here) contains all the *public* stuff, i.e. everything not secret.
- $L = \{(\mathsf{ct}, pk) \ : \ \exists m, r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(pk, m; r)\}$

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

pk ⟶

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

pk →

← ct

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

$m \leftarrow \text{Dec}(pk, \text{ct})$

pk →

← ct

Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

$m \leftarrow \text{Dec}(pk, \text{ct})$

pk →

ct ←

m →

Public Key IND-CCA-1 Security

IND-CCA-1 Security

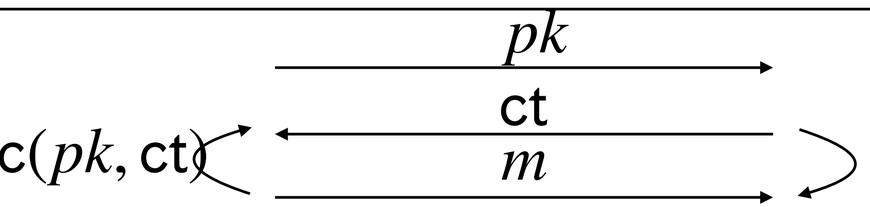
A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

$m \leftarrow \text{Dec}(pk, \text{ct})$



Public Key IND-CCA-1 Security

IND-CCA-1 Security

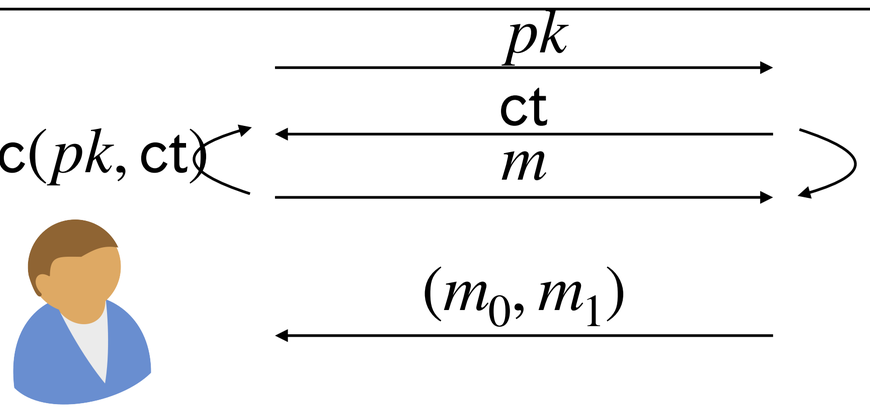
A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

$m \leftarrow \text{Dec}(pk, \text{ct})$

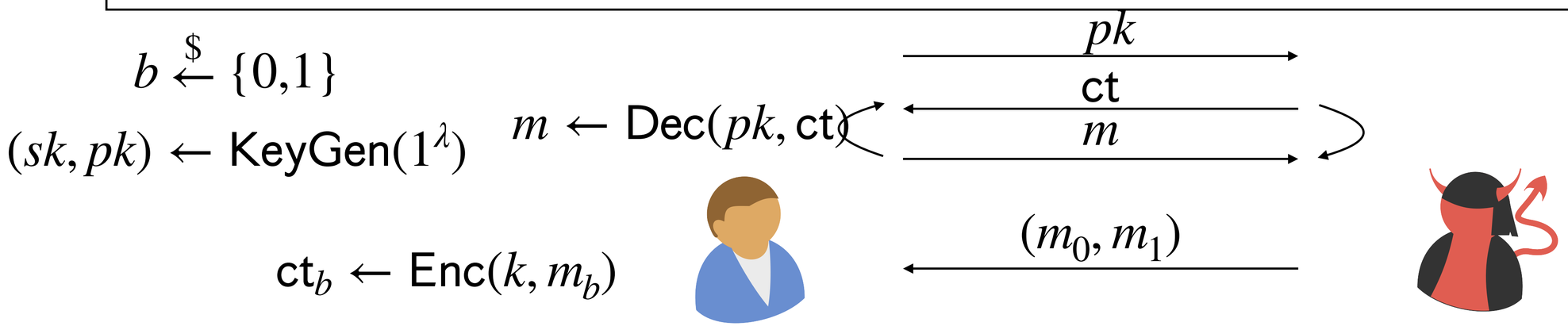


Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

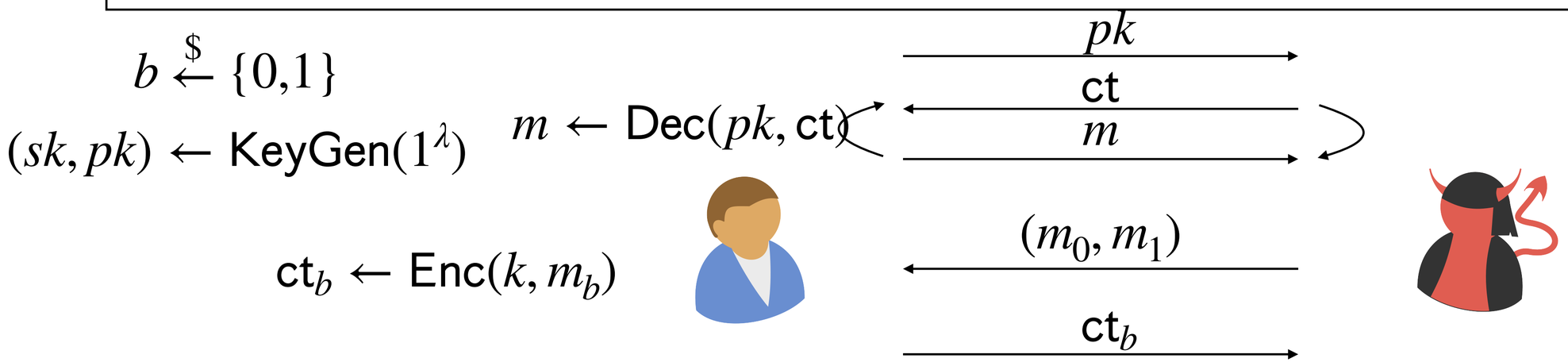


Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

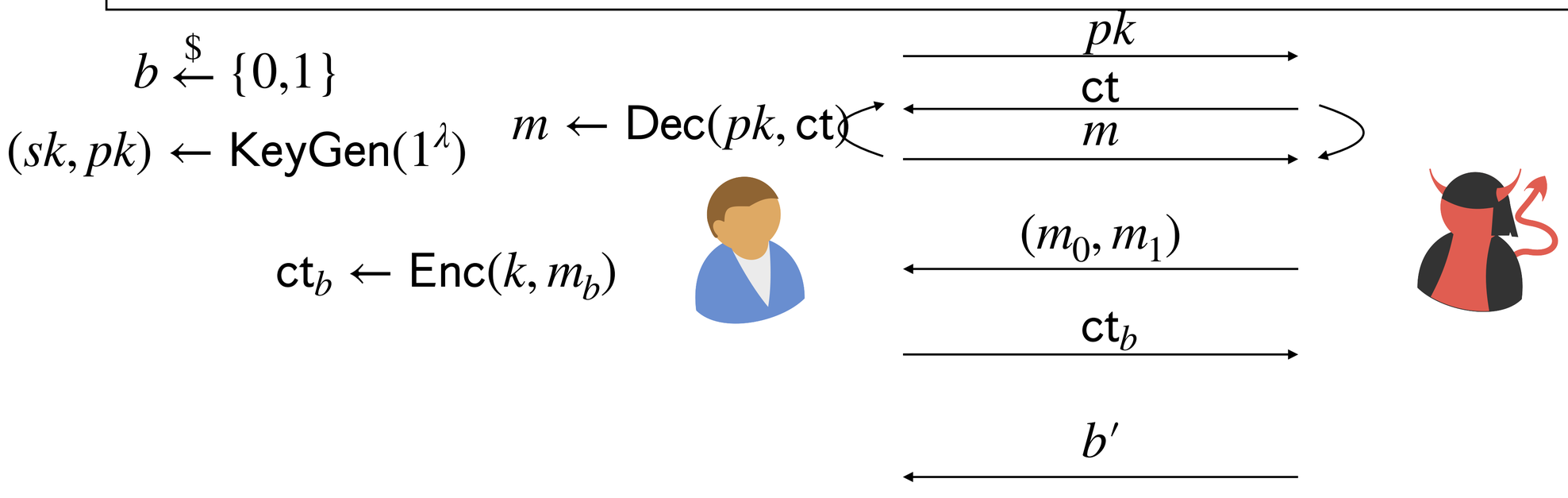


Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$



Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

$m \leftarrow \text{Dec}(pk, \text{ct})$

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

pk

ct

m

(m_0, m_1)

ct_b

b'

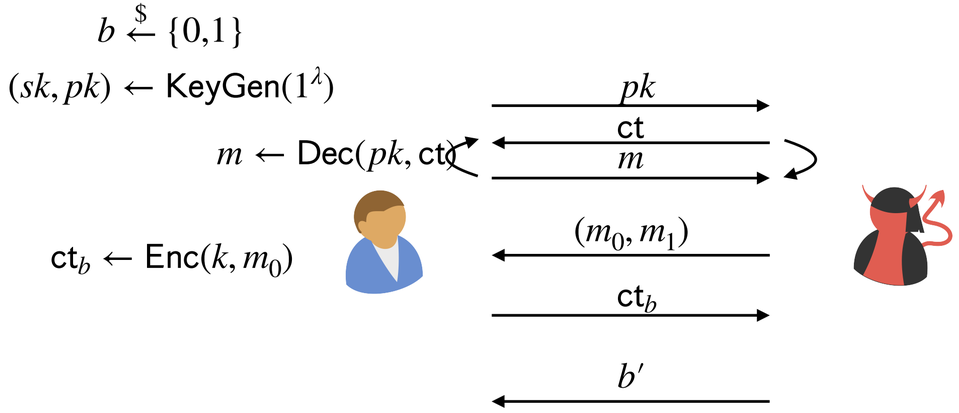
Wins if $b' = b$

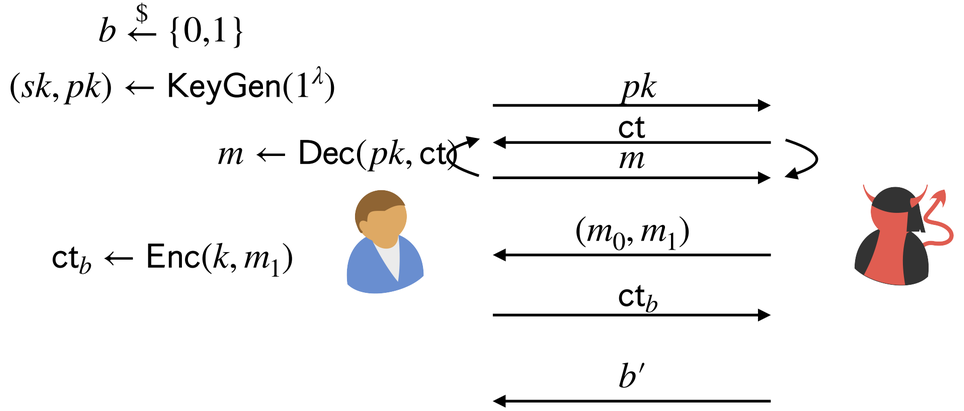
Public Key IND-CCA-1 Security

IND-CCA-1 Security

A public key encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

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





Attempted Proof: IND-CPA => IND-CCA-1

Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$

Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

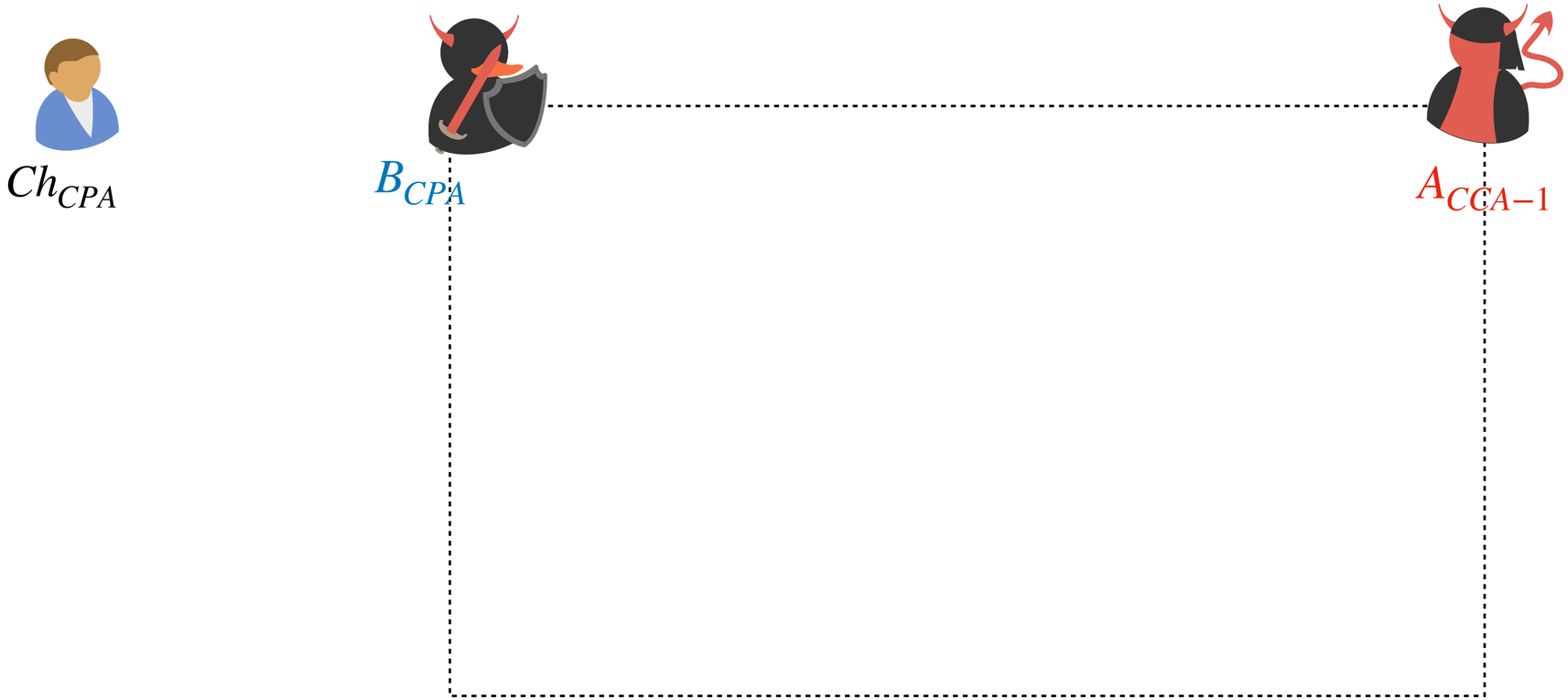
- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$

- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$

Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

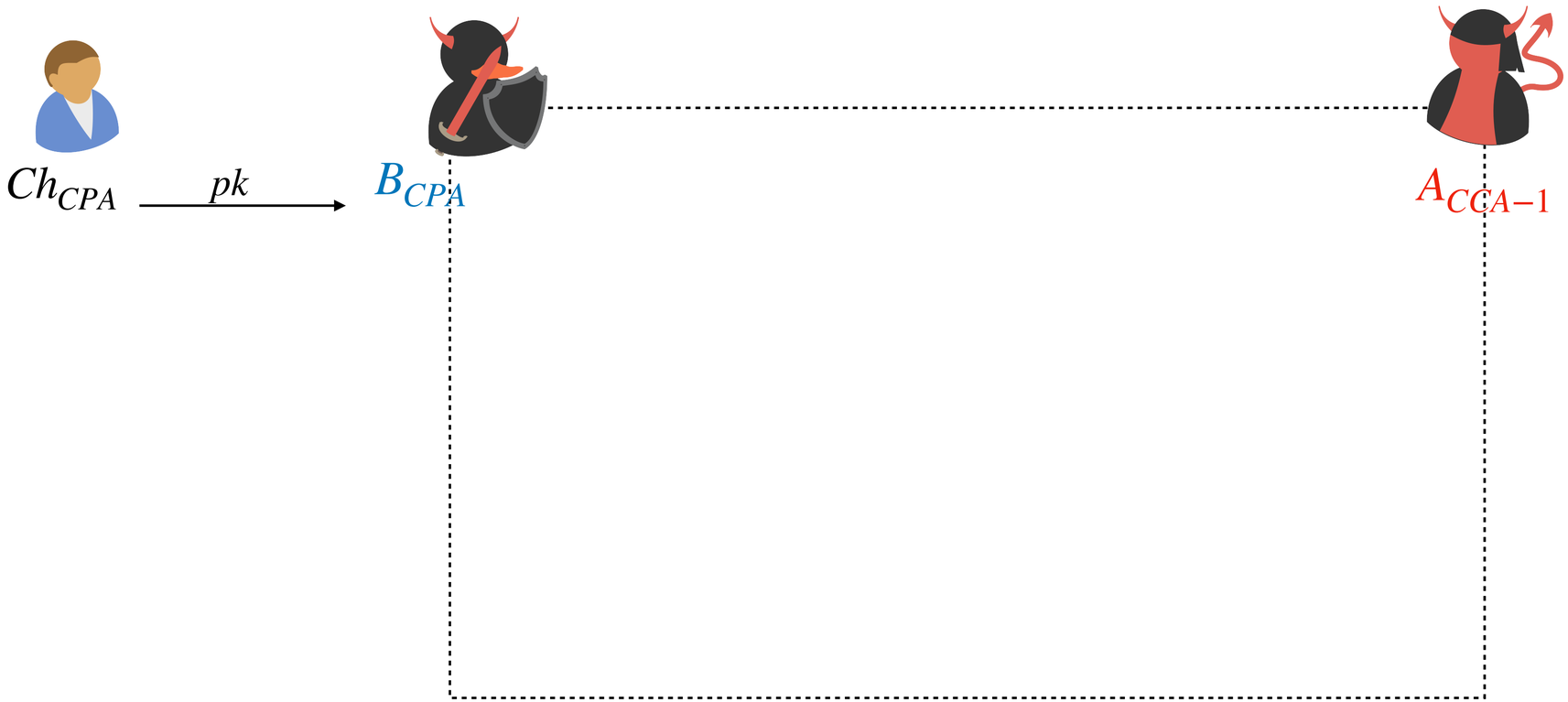
- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

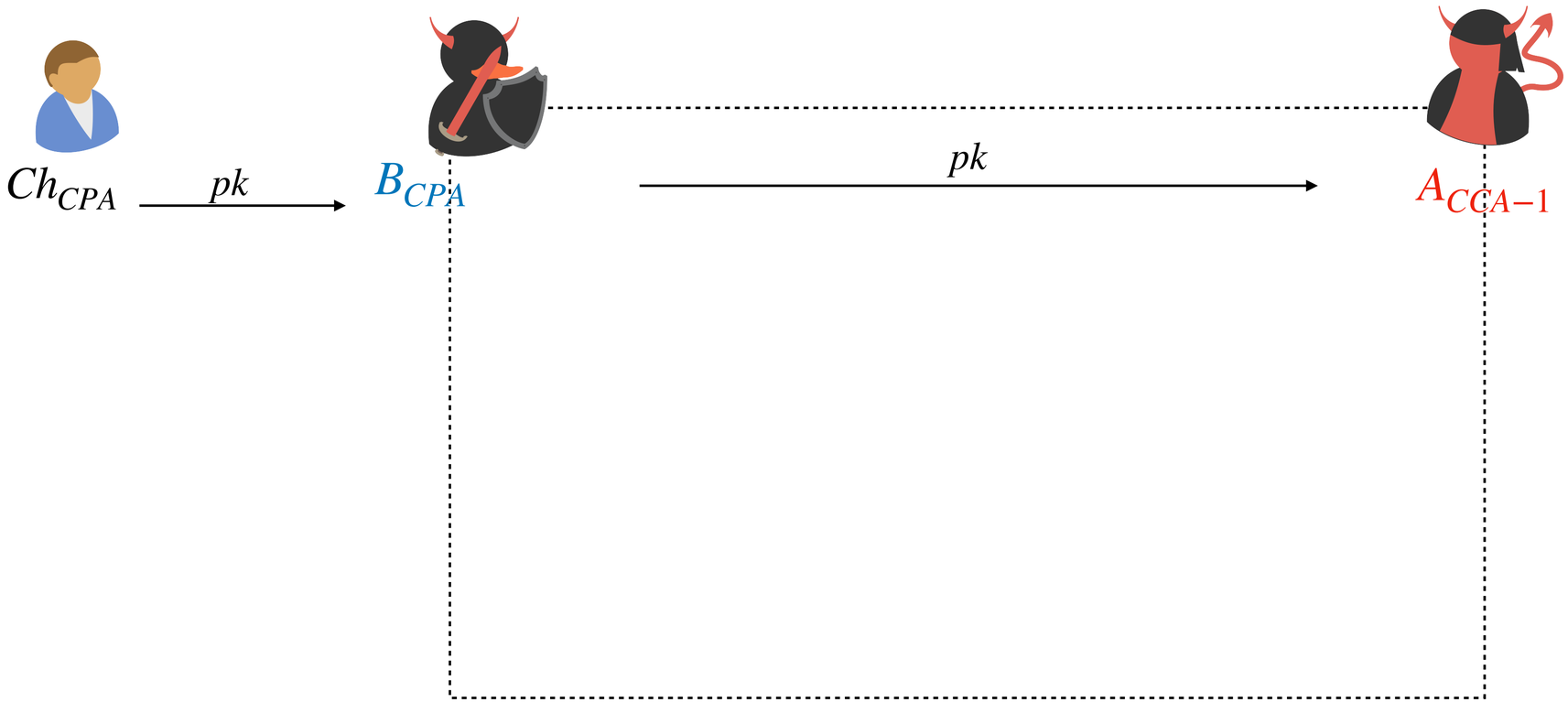
- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

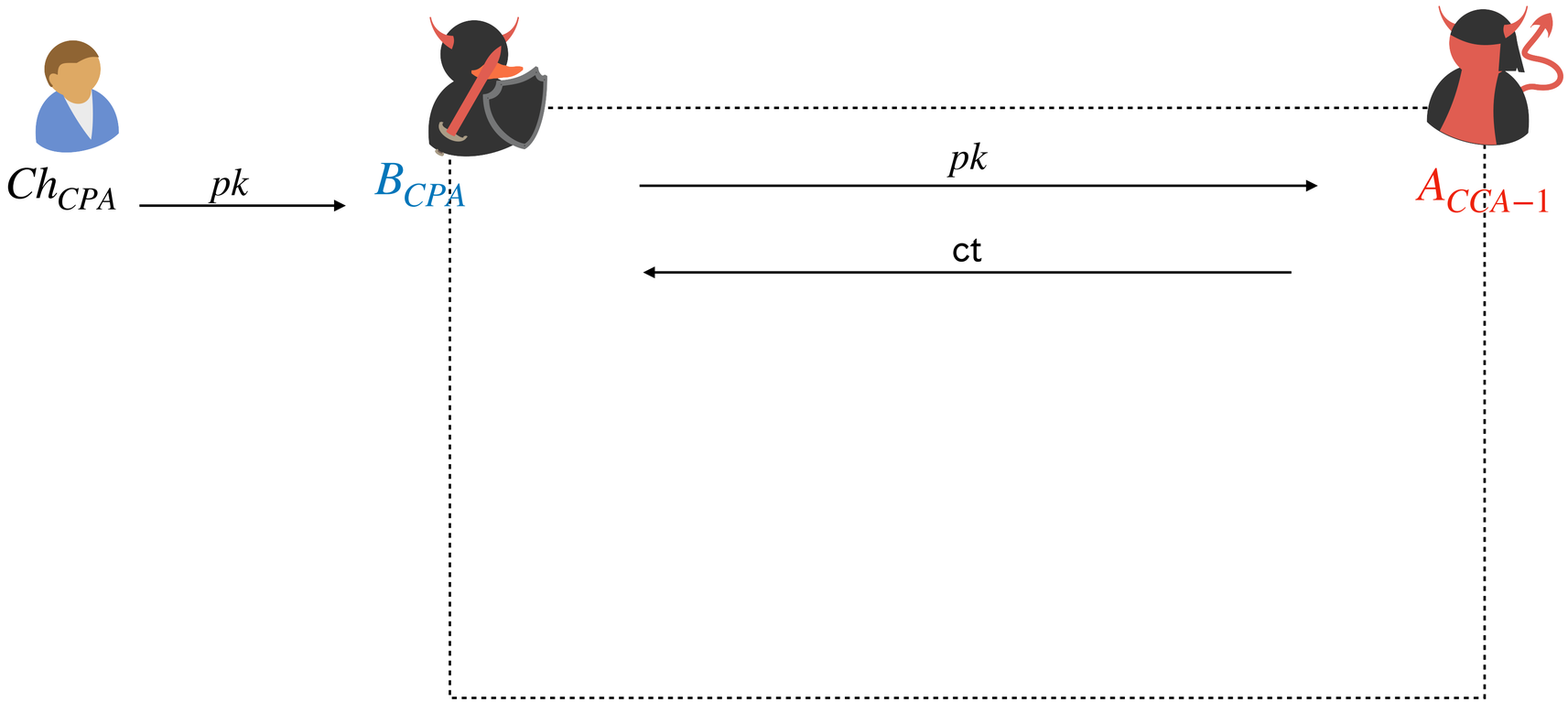
- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

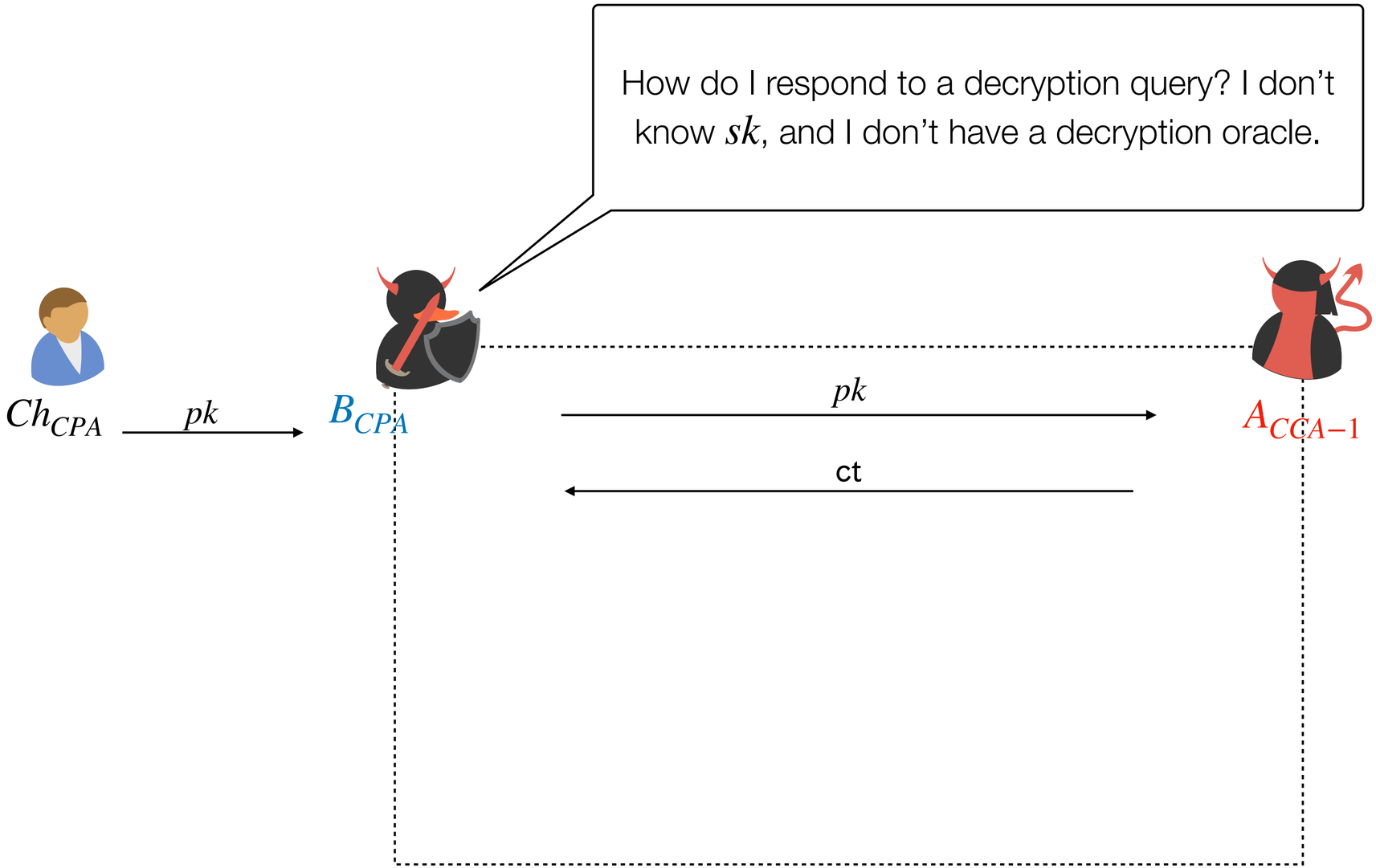
- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

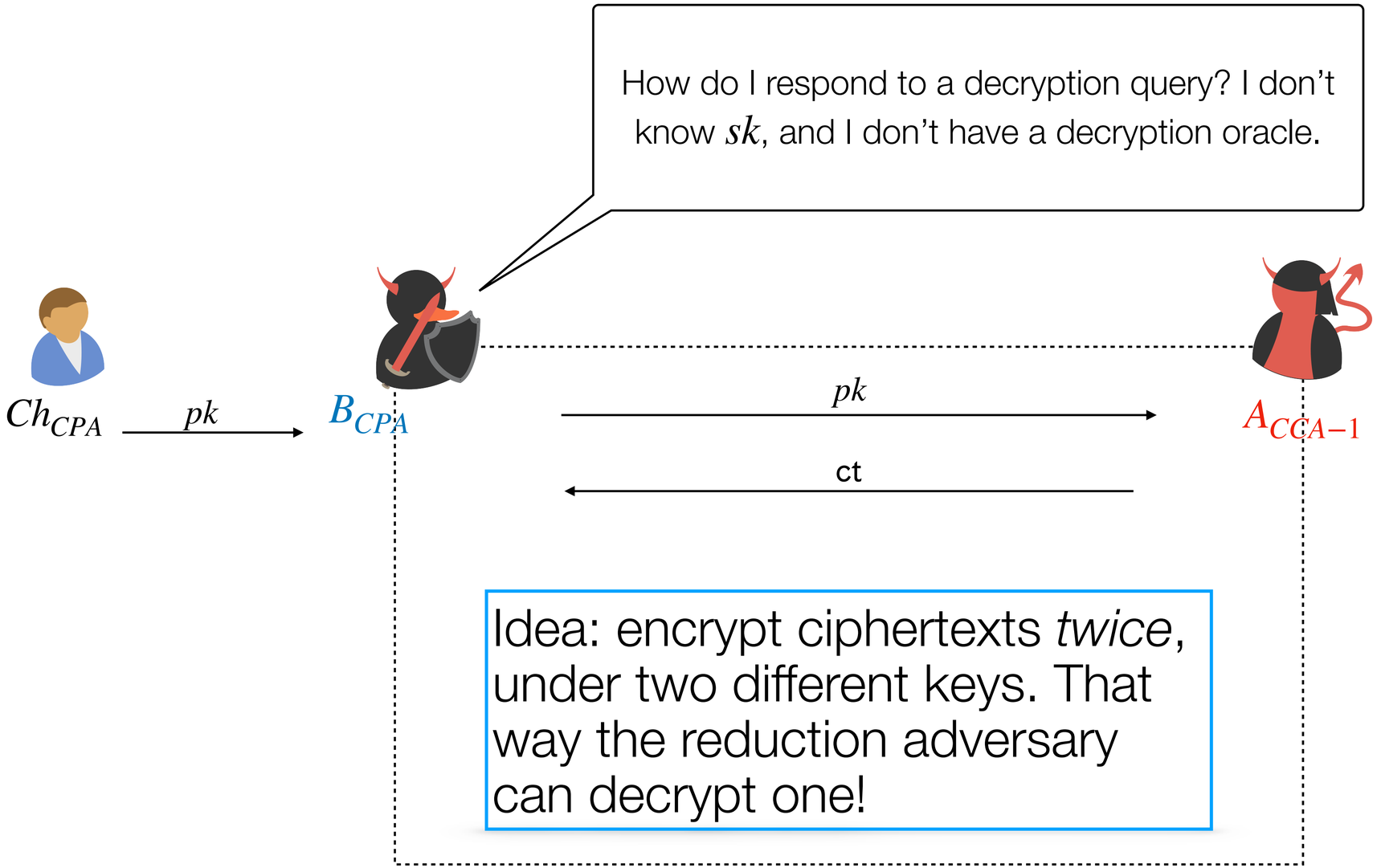
- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$



Attempted Proof: IND-CPA => IND-CCA-1

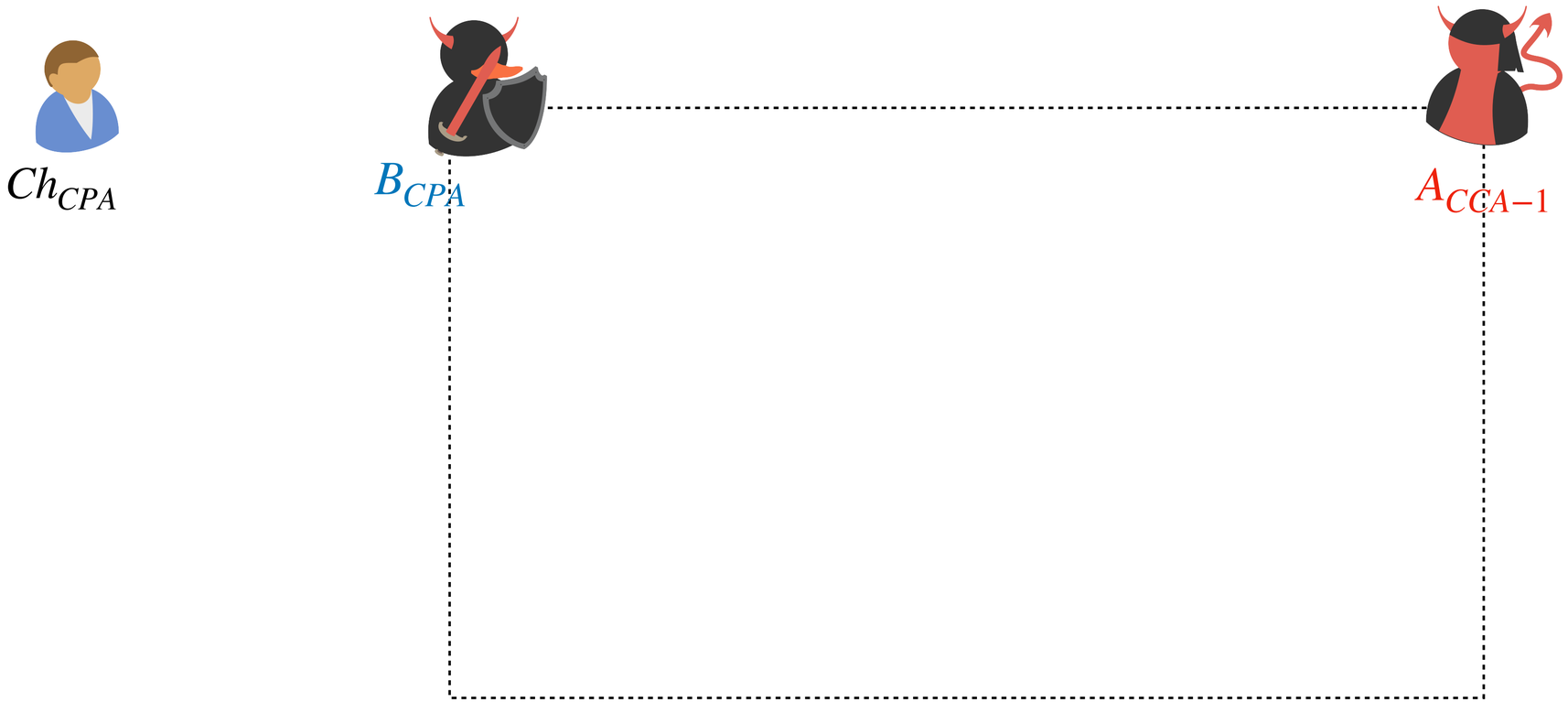
Our CCA-1 Scheme

- $\mathsf{KeyGen}'(1^\lambda)$
  - return $\mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'(pk, m)$
  - return $\mathsf{Enc}(pk, m)$



Attempted Proof: IND-CPA => IND-CCA-1

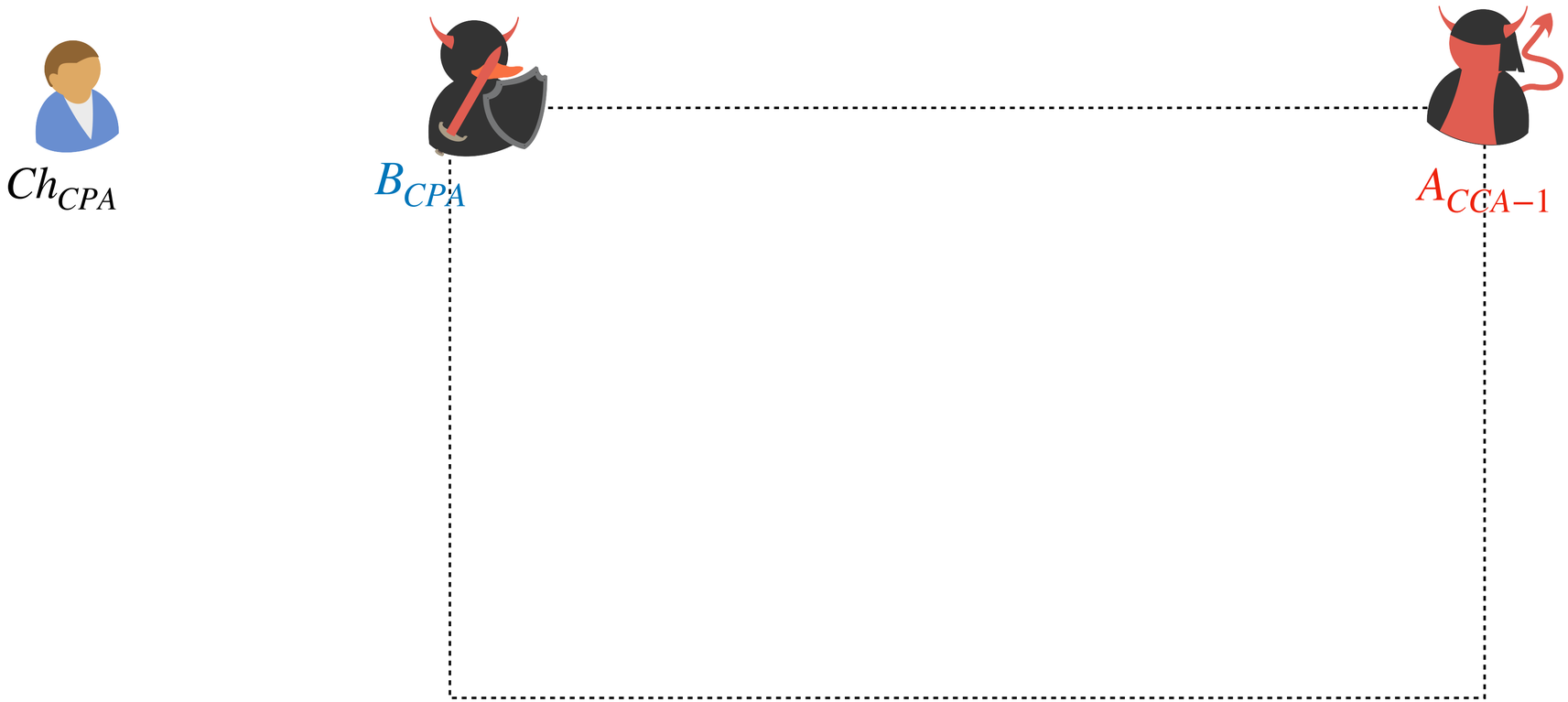
Our CCA-1 Scheme



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

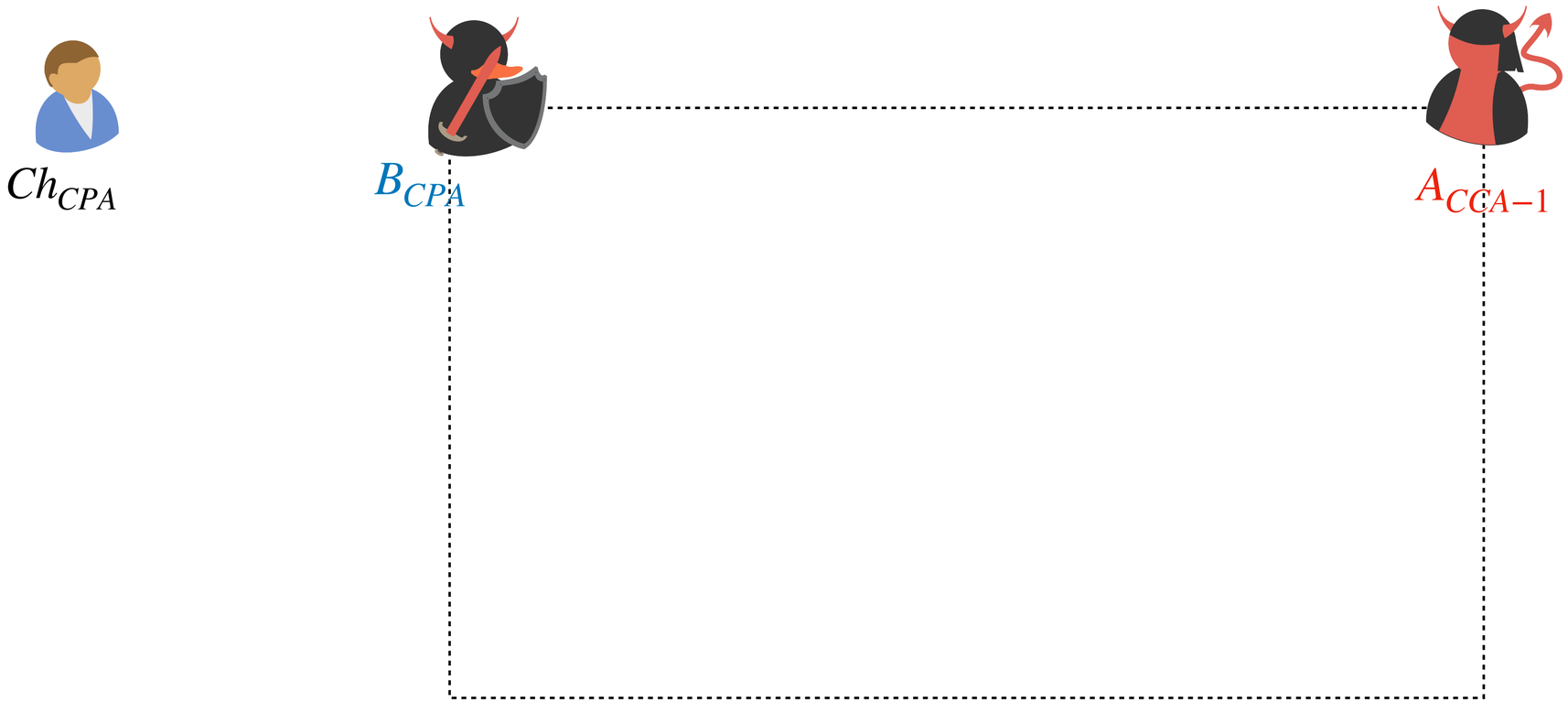
- KeyGen$'(1^{\lambda})$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

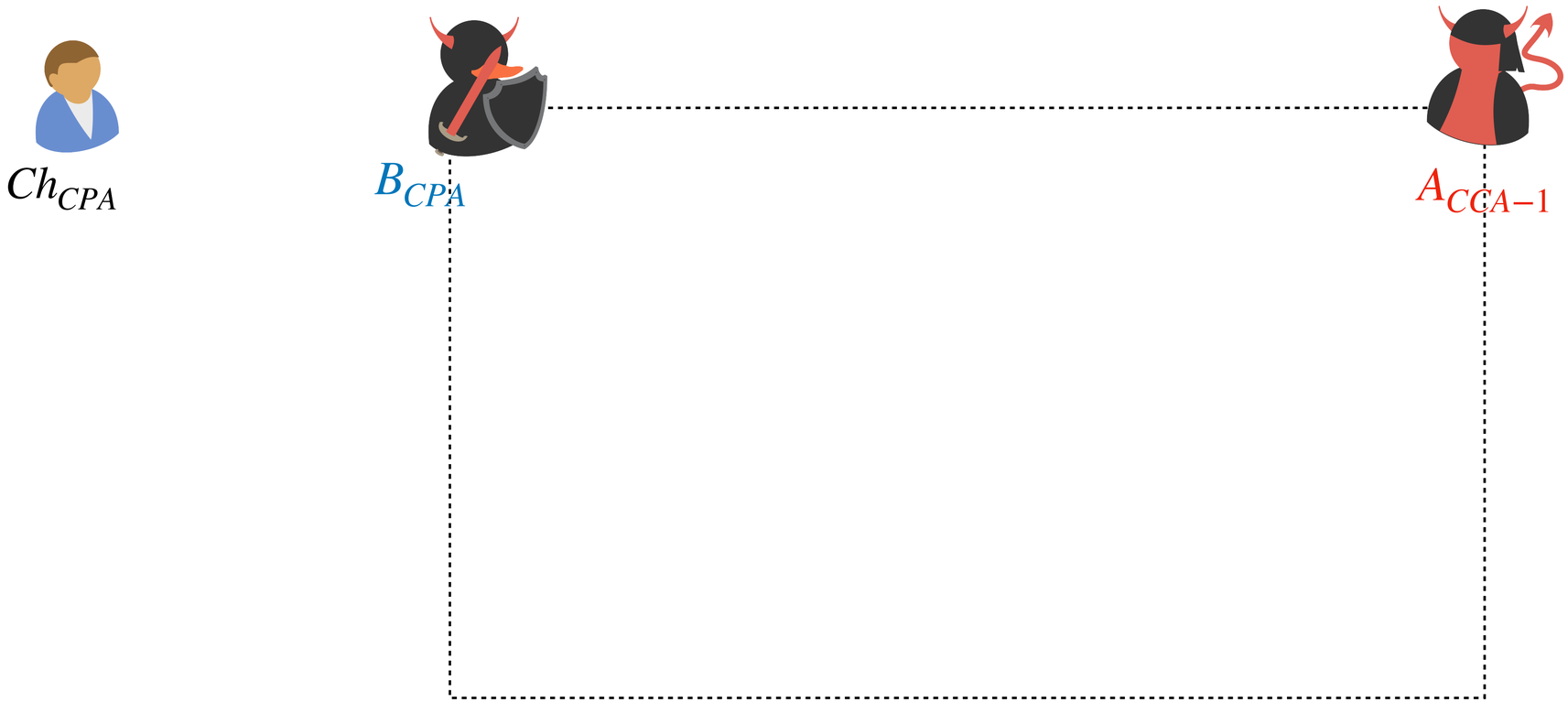
- KeyGen$'(1^\lambda)$
  - $pk_0 \leftarrow$ KeyGen(1^λ)



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

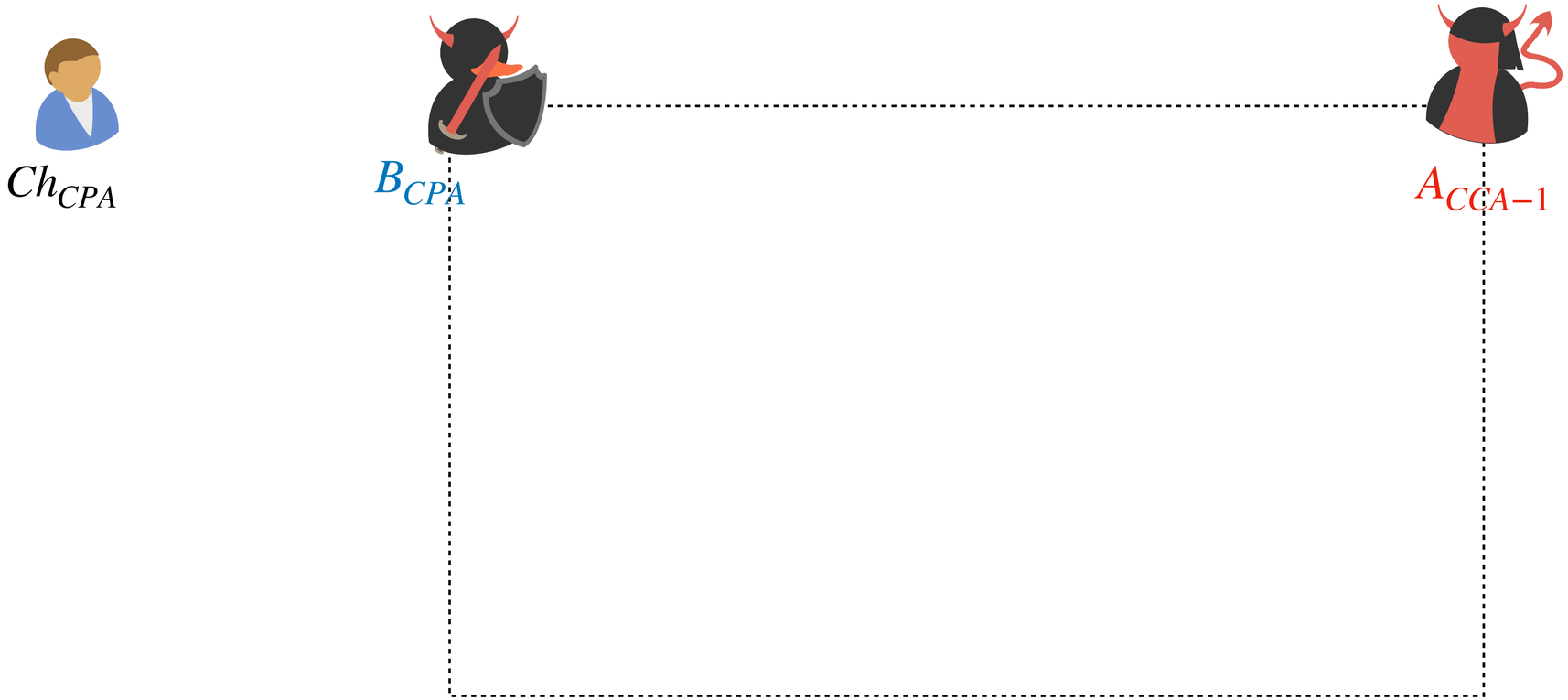
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

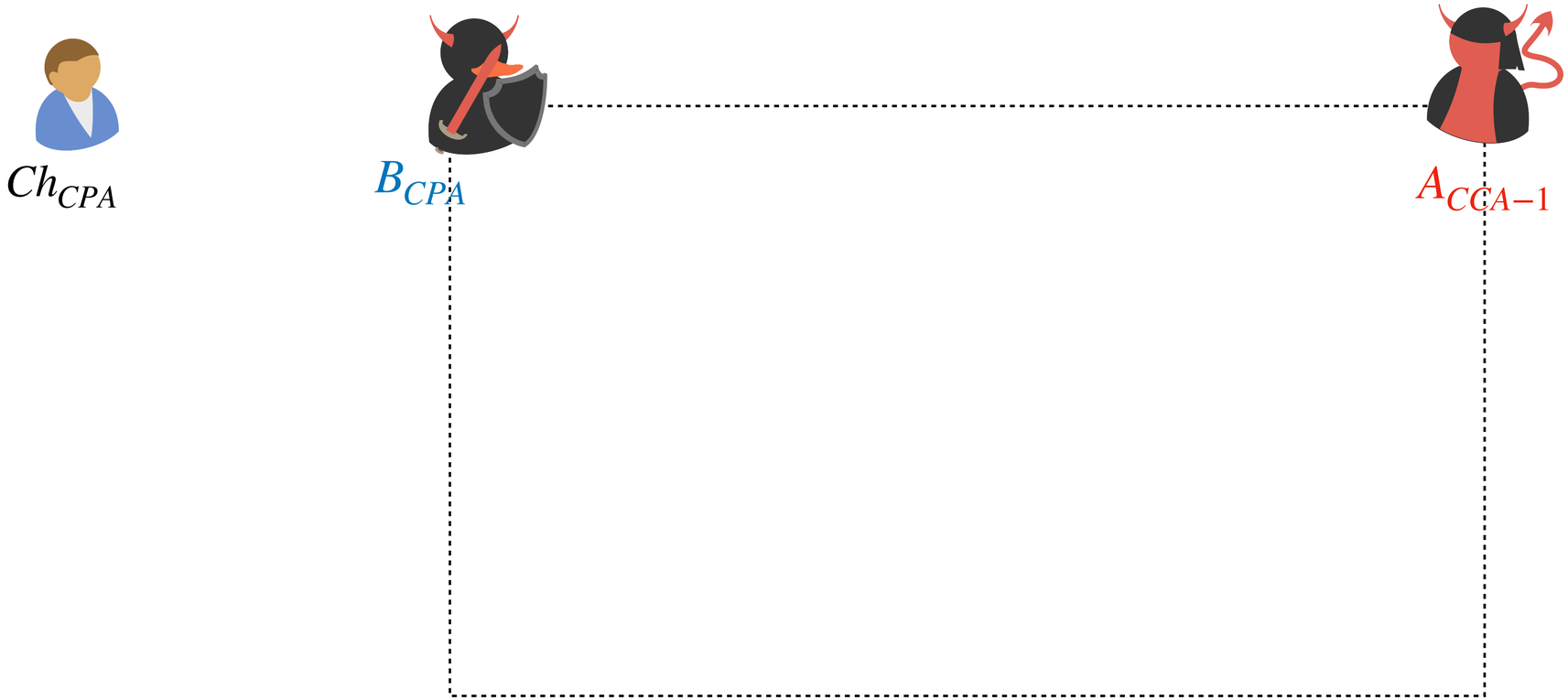
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

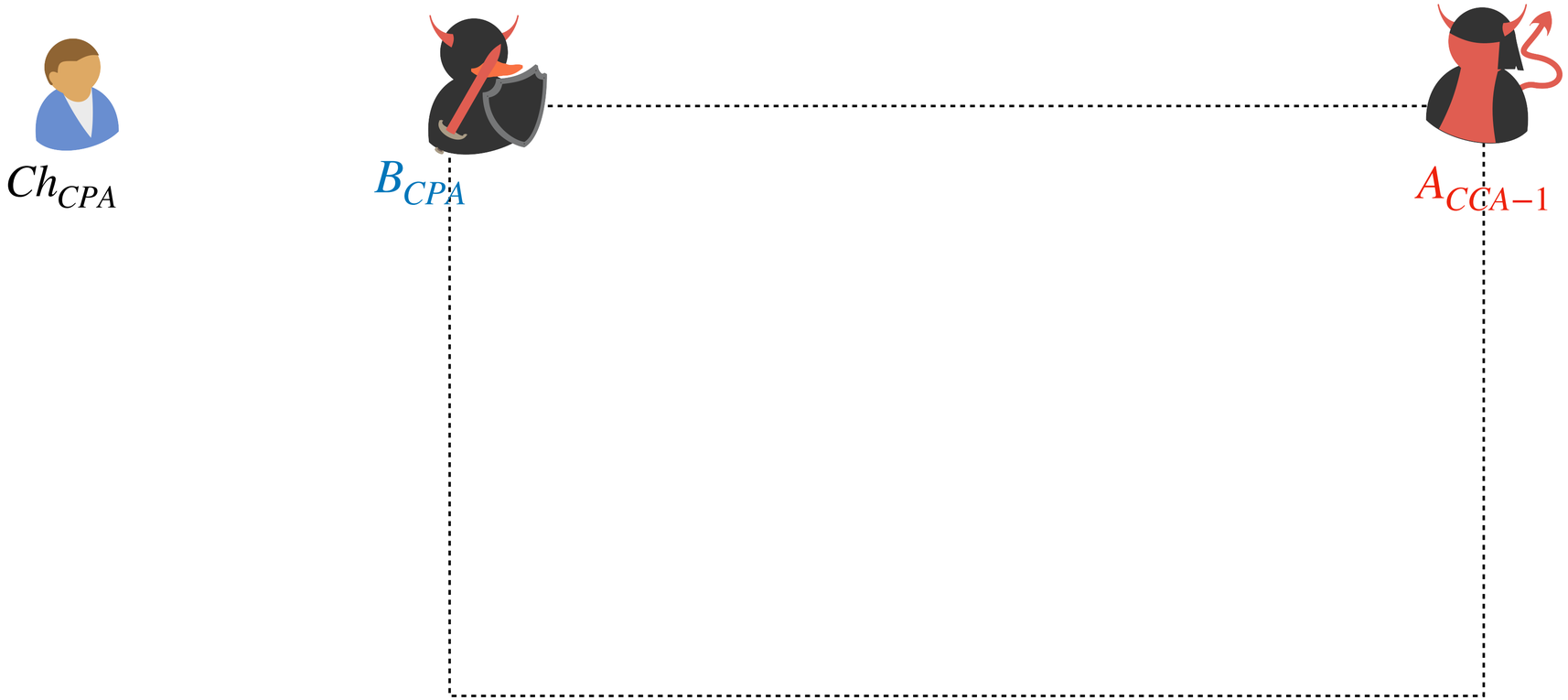
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

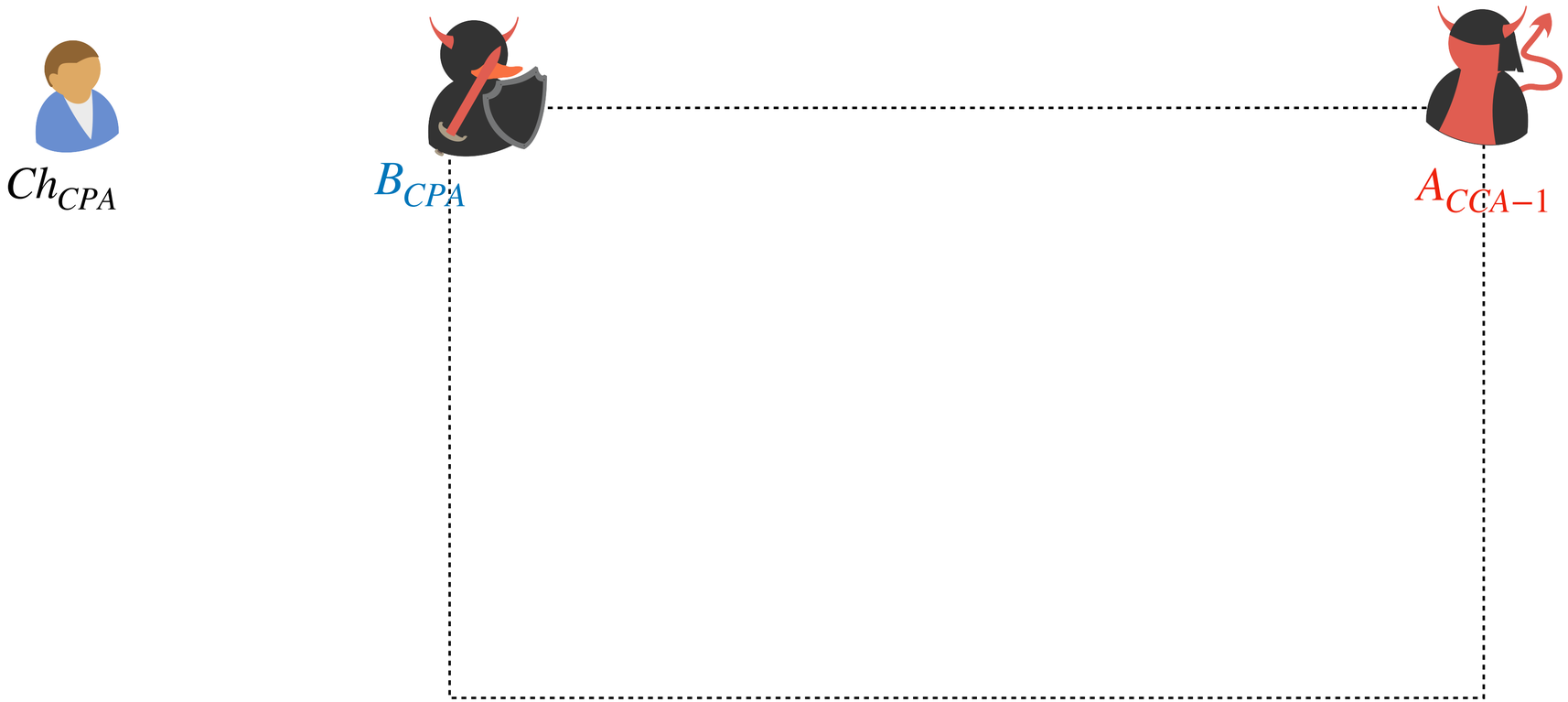
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

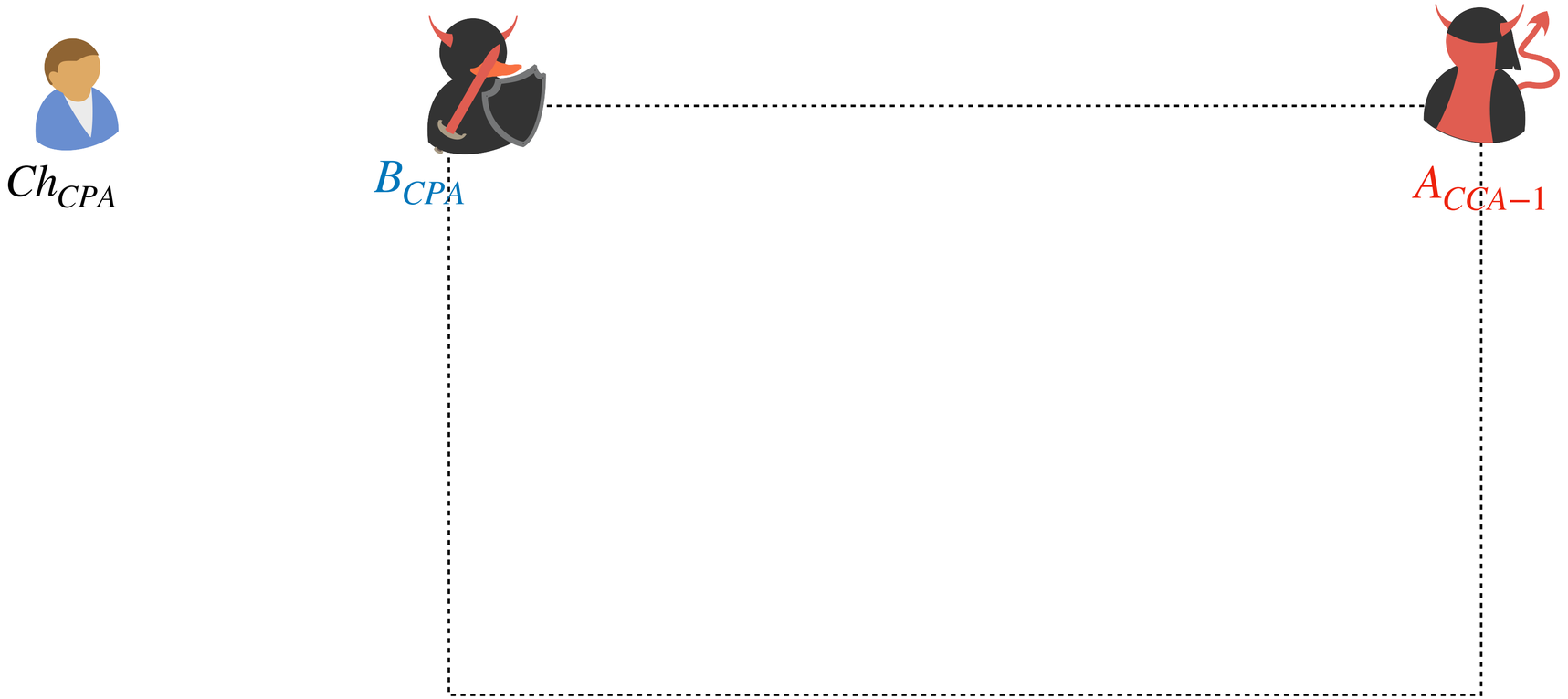
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

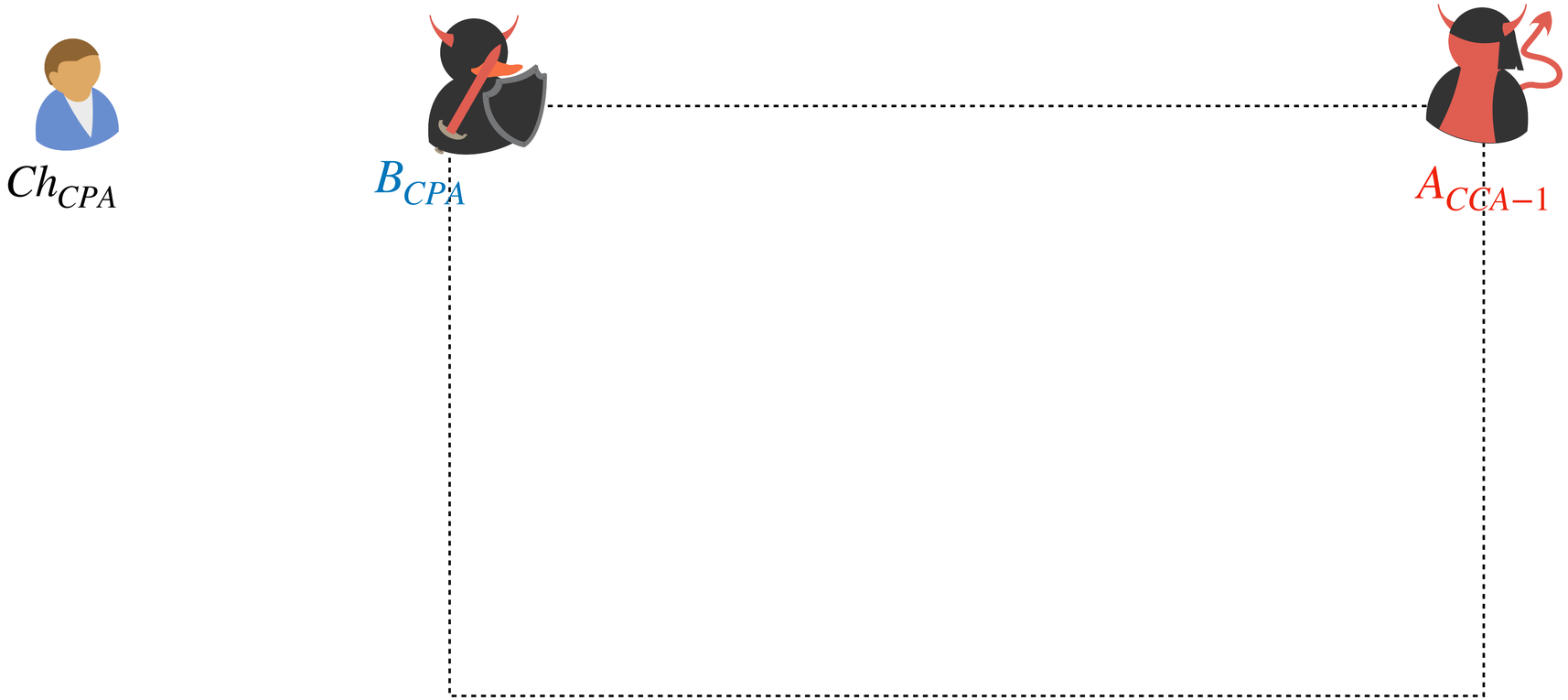
- KeyGen$'(1^\lambda)$
  - $pk_0 \leftarrow \text{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \text{KeyGen}(1^\lambda)$
- Enc$'((pk_0, pk_1), m)$
  - $\text{ct}_0 \leftarrow \text{Enc}(pk_0, m)$
  - $\text{ct}_1 \leftarrow \text{Enc}(pk_1, m)$
  - return $(\text{ct}_0, \text{ct}_1)$
- Dec$'((sk_0, sk_1), (\text{ct}_0, \text{ct}_1))$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

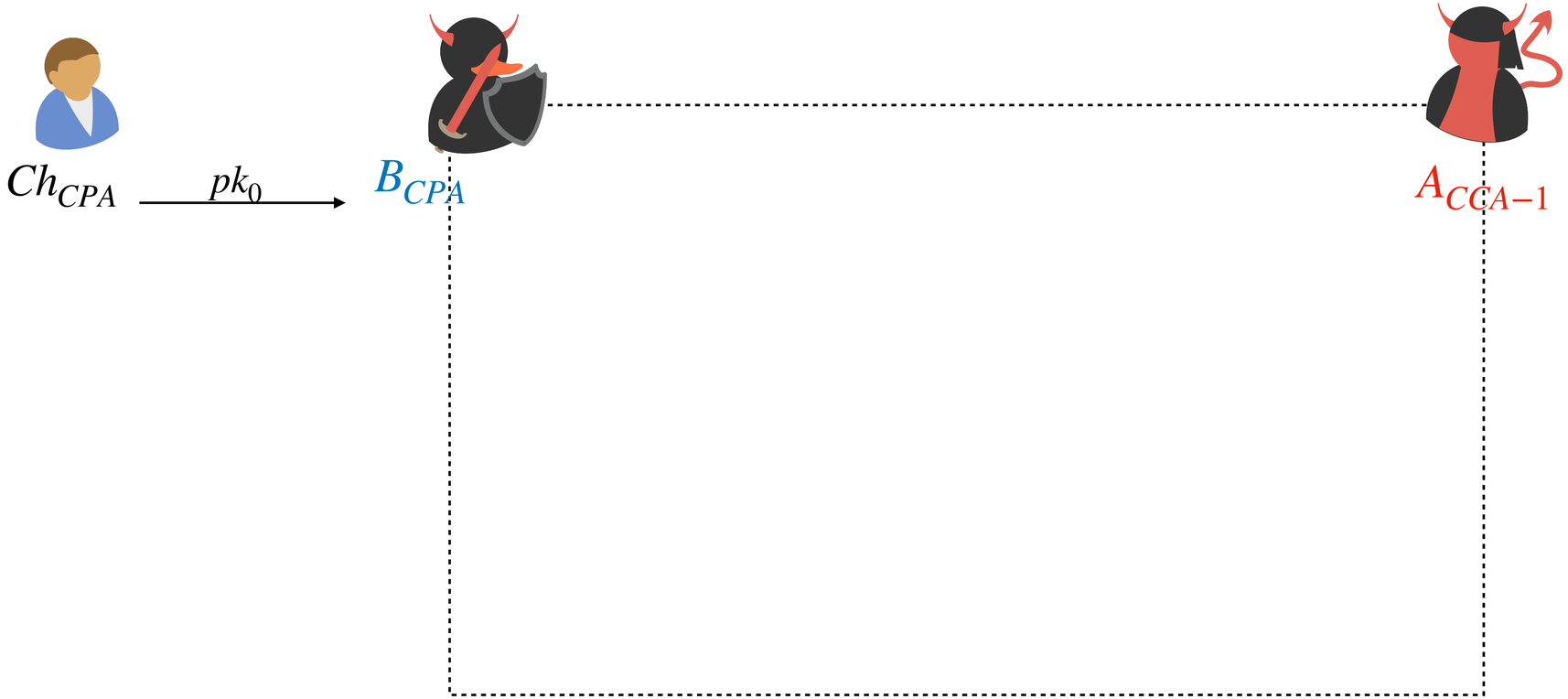
- KeyGen$'(1^\lambda)$
  - $pk_0 \leftarrow$ KeyGen(1^λ)
  - $pk_1 \leftarrow$ KeyGen(1^λ)
- Enc$'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- Dec$'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

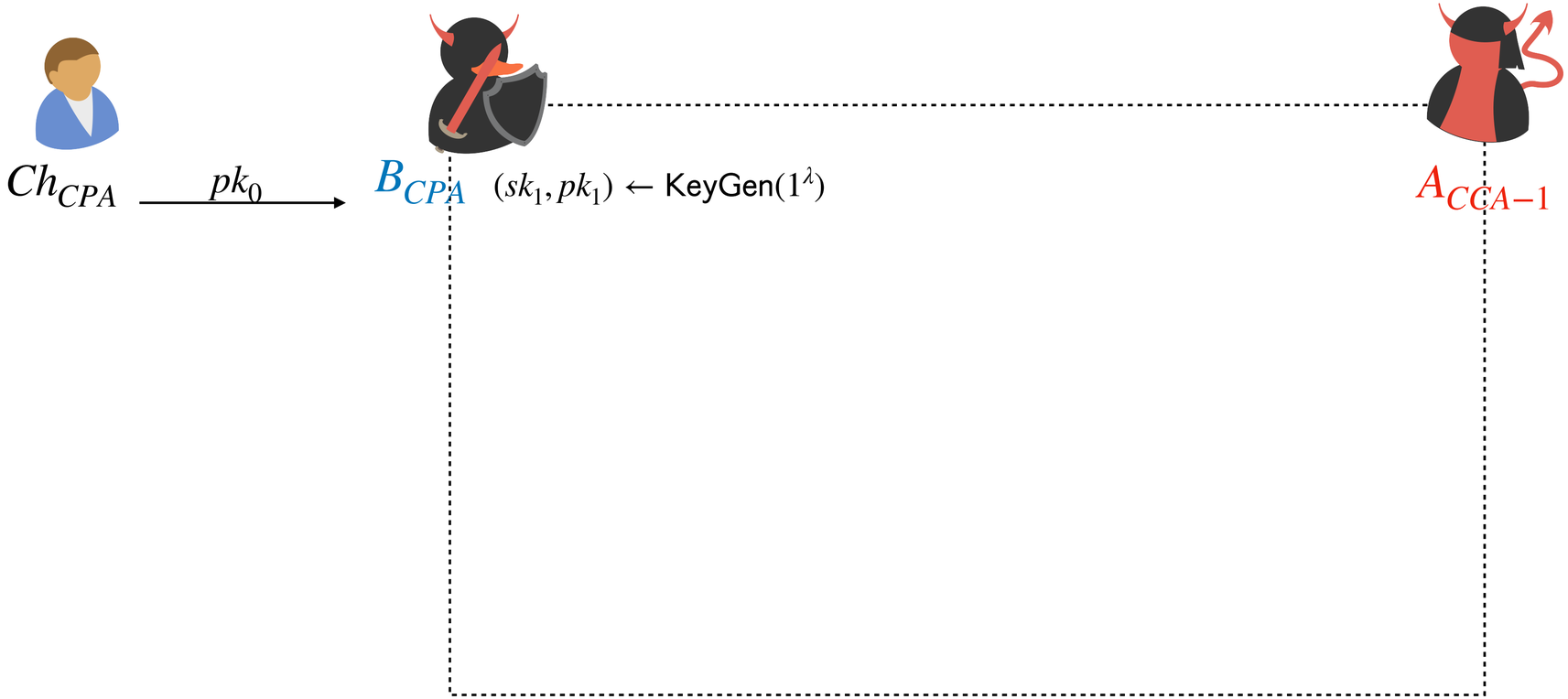
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

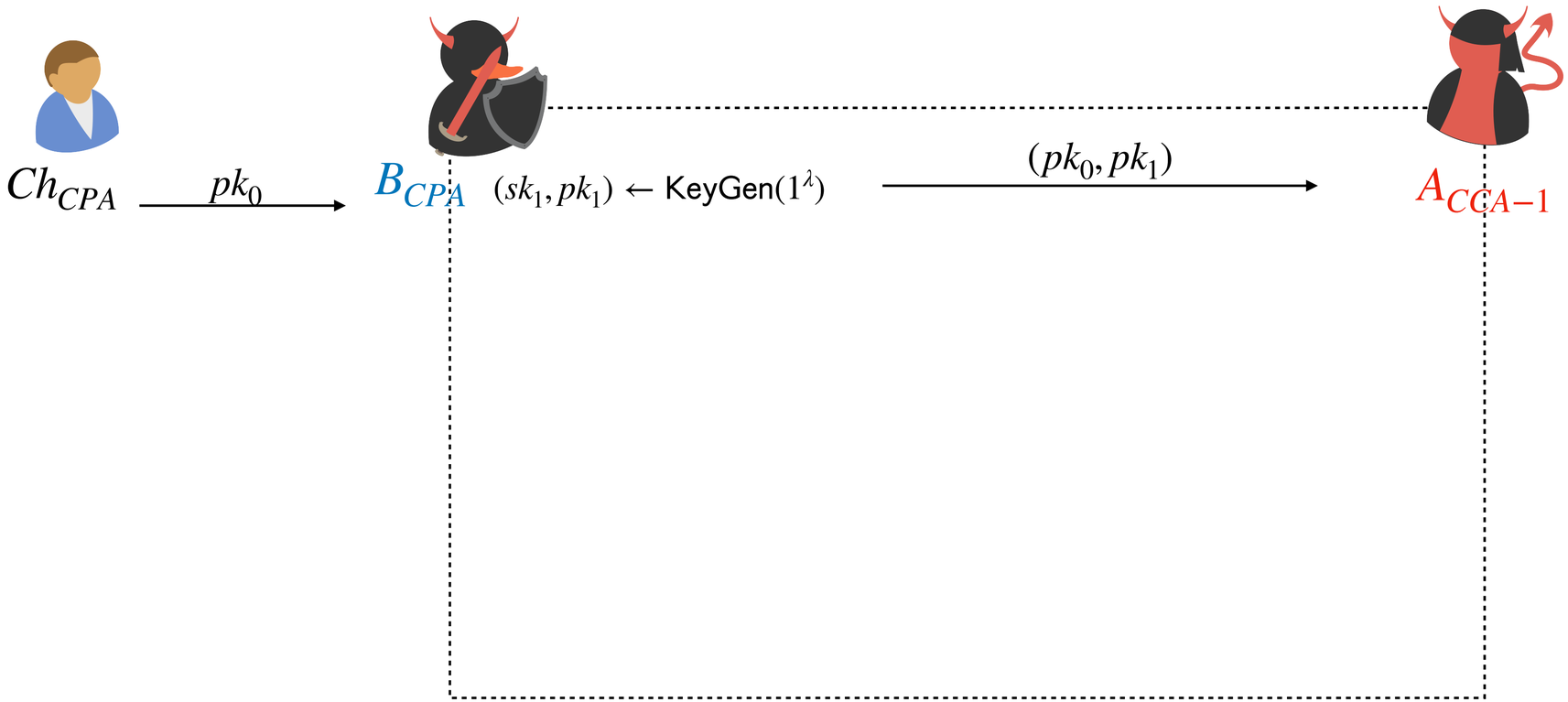
- KeyGen$'(1^\lambda)$
  - $pk_0 \leftarrow \text{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \text{KeyGen}(1^\lambda)$
- Enc$'((pk_0, pk_1), m)$
  - $\text{ct}_0 \leftarrow \text{Enc}(pk_0, m)$
  - $\text{ct}_1 \leftarrow \text{Enc}(pk_1, m)$
  - return $(\text{ct}_0, \text{ct}_1)$
- Dec$'((sk_0, sk_1), (\text{ct}_0, \text{ct}_1))$
  - return $\text{Dec}(sk_0, \text{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

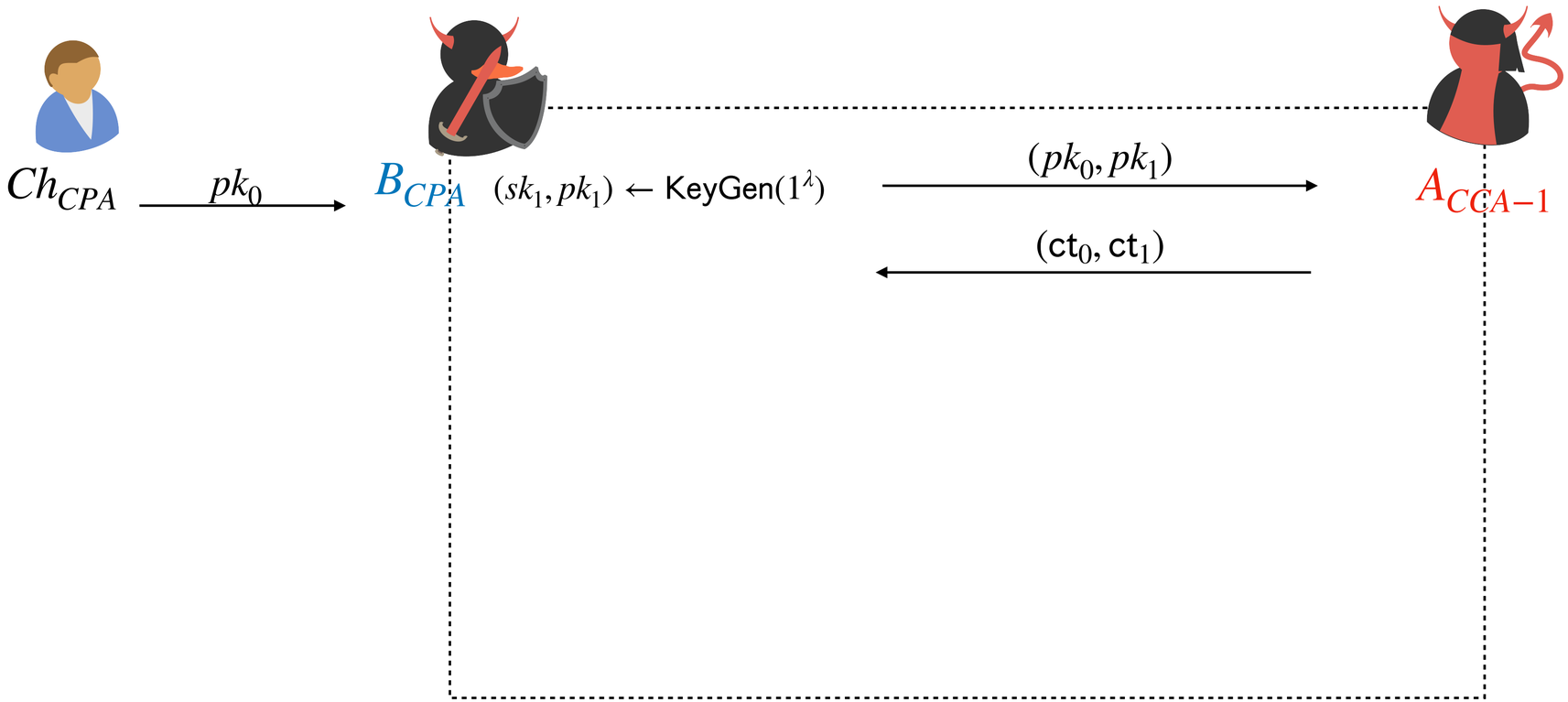
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

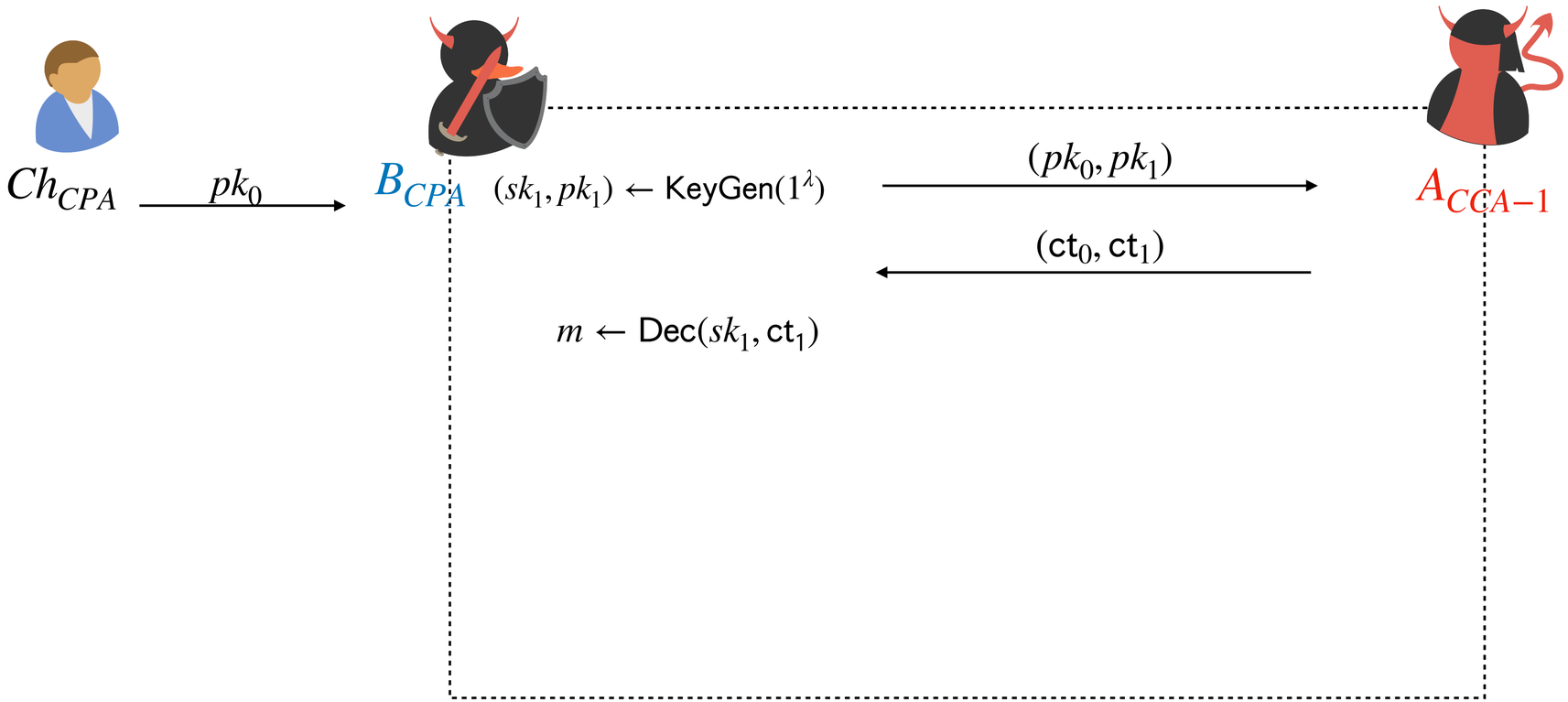
- KeyGen$'(1^\lambda)$
  - $pk_0 \leftarrow \text{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \text{KeyGen}(1^\lambda)$
- Enc$'((pk_0, pk_1), m)$
  - $\text{ct}_0 \leftarrow \text{Enc}(pk_0, m)$
  - $\text{ct}_1 \leftarrow \text{Enc}(pk_1, m)$
  - return $(\text{ct}_0, \text{ct}_1)$
- Dec$'((sk_0, sk_1), (\text{ct}_0, \text{ct}_1))$
  - return $\text{Dec}(sk_0, \text{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

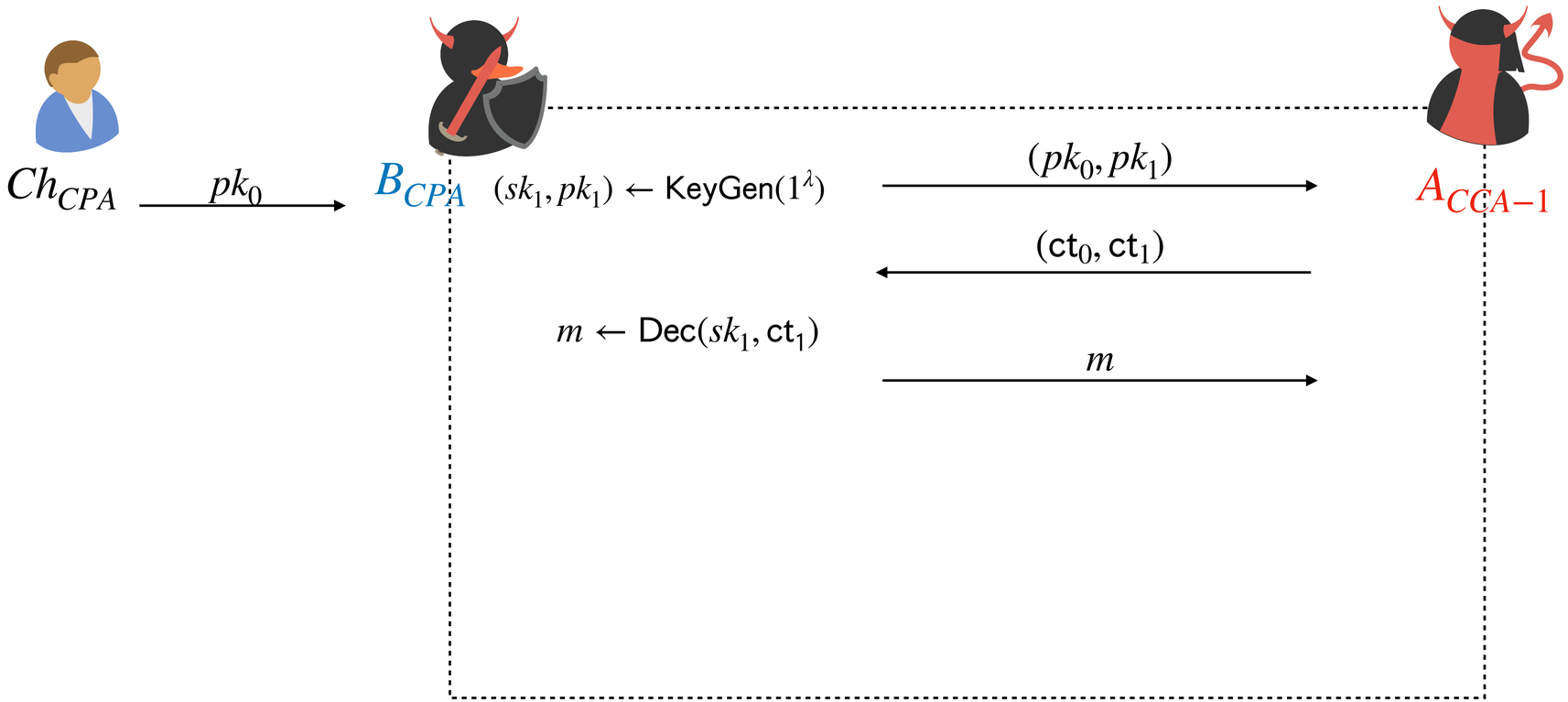
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

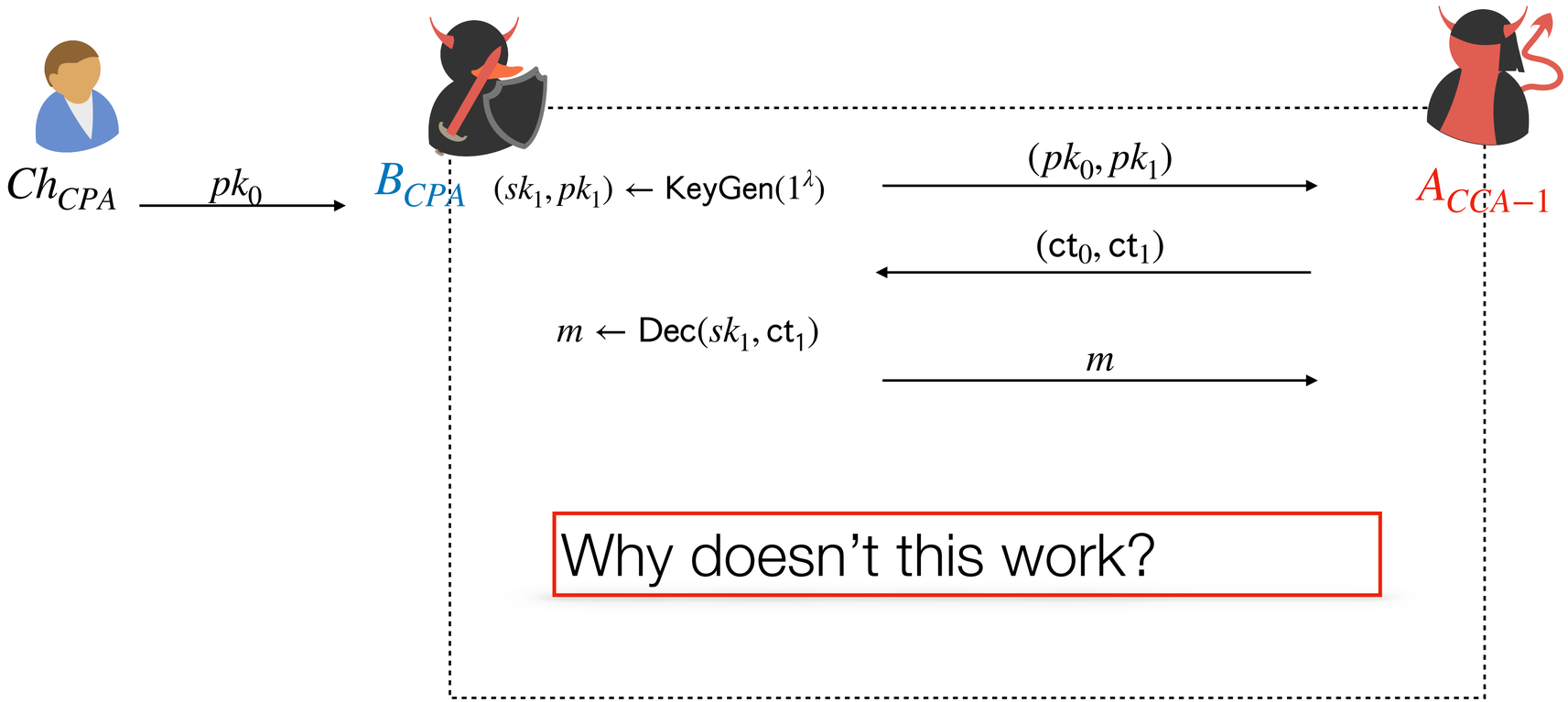
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

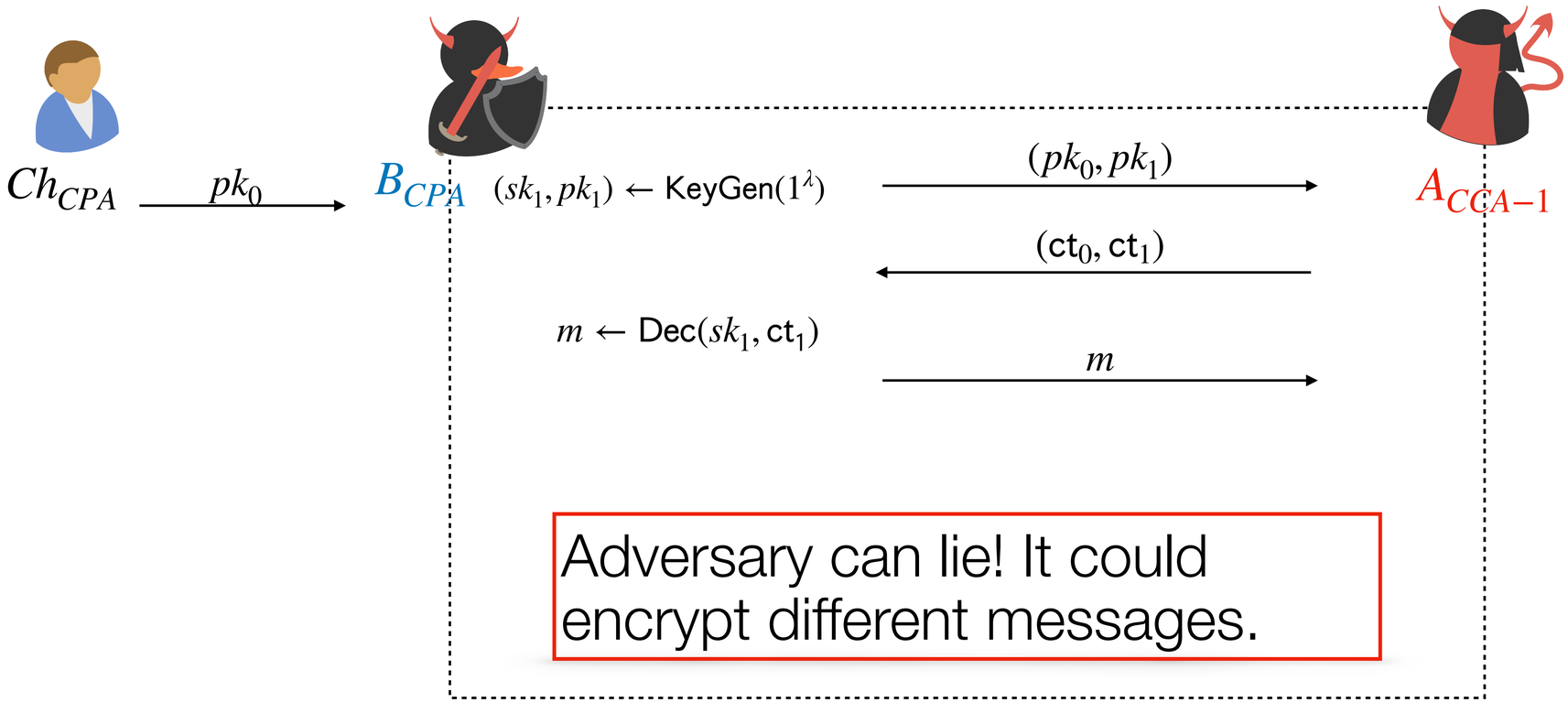
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

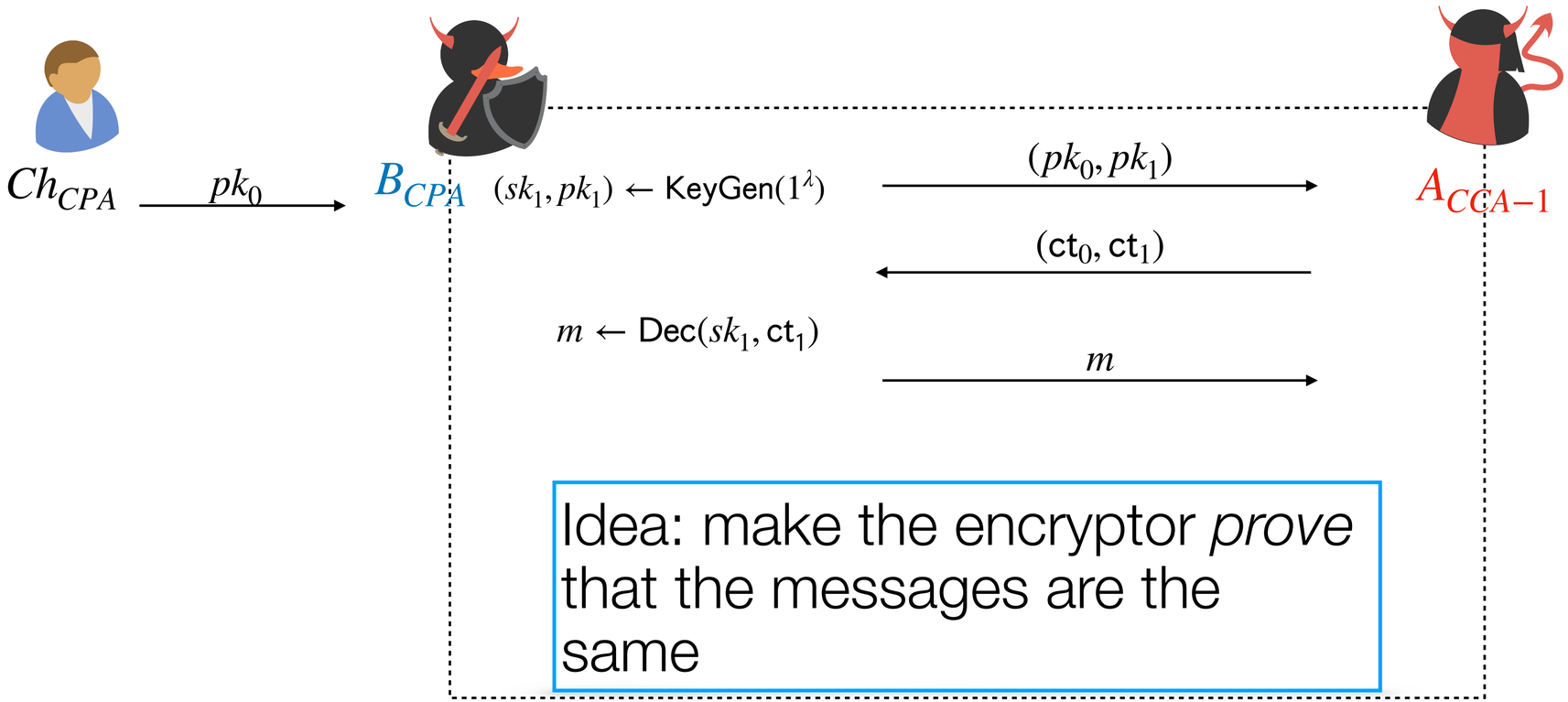
- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Attempted Proof: IND-CPA => IND-CCA-1

Our CCA-1 Scheme

- $\mathsf{KeyGen}'(1^\lambda)$
  - $pk_0 \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $pk_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}'((pk_0, pk_1), m)$
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1))$
  - return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$



Idea: make the encryptor *prove* that the messages are the same

Naor-Yung

Naor-Yung

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$

Naor-Yung

Naor-Yung

Let λ be the security parameter, $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - $\pi \leftarrow P((ct_0, ct_1, pk_0, pk_1), \mathsf{crs})$

Naor-Yung

Naor-Yung

Let λ be the security parameter, (KGen, Enc, Dec) be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - $\pi \leftarrow P((ct_0, ct_1, pk_0, pk_1), \mathsf{crs})$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1, \pi)$

Naor-Yung

Naor-Yung

Let λ be the security parameter, (KGen, Enc, Dec) be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - $\pi \leftarrow P((ct_0, ct_1, pk_0, pk_1), \mathsf{crs})$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1, \pi)$

Naor-Yung

Naor-Yung

Let λ be the security parameter, (KGen, Enc, Dec) be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - $\pi \leftarrow P((ct_0, ct_1, pk_0, pk_1), \mathsf{crs})$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1, \pi))$

Naor-Yung

Naor-Yung

Let λ be the security parameter, (KGen, Enc, Dec) be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - $\pi \leftarrow P((ct_0, ct_1, pk_0, pk_1), \mathsf{crs})$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1, \pi))$
  - if $V(\pi, \mathsf{crs}) = 0$ return $\perp$

Naor-Yung

Naor-Yung

Let λ be the security parameter, (KGen, Enc, Dec) be an IND-CPA secure public key encryption scheme, and let (P, V) be a NIZK for the following language:

$$L = \left\{ (\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1) \ : \ \exists k, m, r_0, r_1 \ \text{ s.t. } \begin{array}{c} \mathsf{ct}_0 = \mathsf{Enc}(pk_0, m; r_0) \wedge \\ \mathsf{ct}_1 = \mathsf{Enc}(pk_1, m; r_1) \end{array} \right\}$$

- $\mathsf{KGen}'(1^\lambda)$:
  - $(sk_0, pk_0) \leftarrow \mathsf{KGen}(1^\lambda)$
  - $(sk_1, pk_1) \leftarrow \mathsf{KGen}(1^\lambda)$
  - return $((sk_0, sk_1), (pk_0, pk_1))$
- $\mathsf{Enc}'((pk_0, pk_1), m)$:
  - $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(pk_0, m)$
  - $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(pk_1, m)$
  - $\pi \leftarrow P((ct_0, ct_1, pk_0, pk_1), \mathsf{crs})$
  - return $(\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
- $\mathsf{Dec}'((sk_0, sk_1), (\mathsf{ct}_0, \mathsf{ct}_1, \pi))$
  - if $V(\pi, \mathsf{crs}) = 0$ return $\perp$
  - else return $\mathsf{Dec}(sk_0, \mathsf{ct}_0)$

Proof of Security

Proof of Security

$\mathcal{G}_0$

Proof of Security

$\mathcal{G}_0$

Proof of Security

$\mathrm{crs} \leftarrow \mathrm{CRSGen}(1^\lambda)$

$\mathcal{G}_0$

Proof of Security

$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$

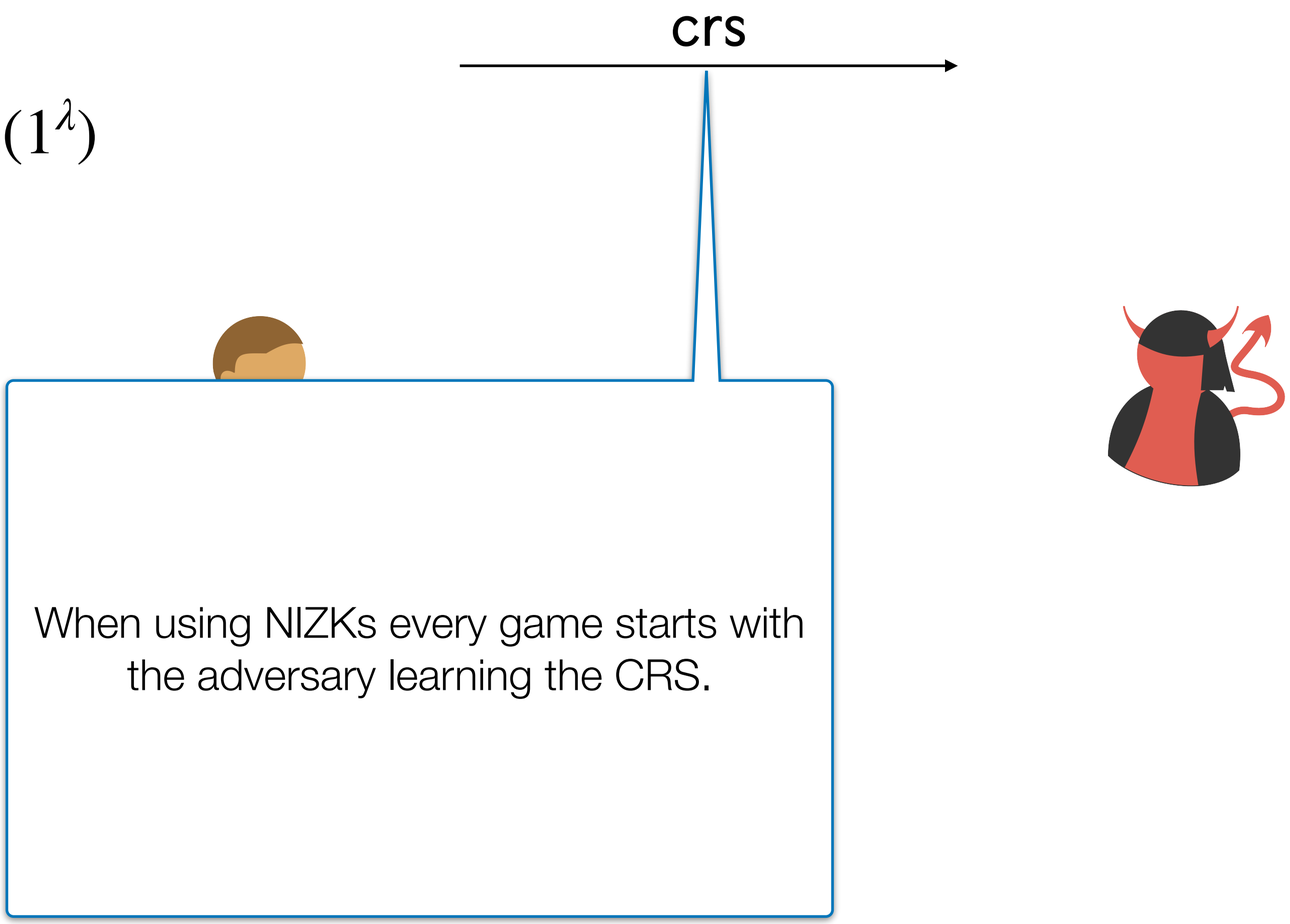
crs

$\mathcal{G}_0$

Proof of Security

$\mathrm{crs} \leftarrow \mathrm{CRSGen}(1^\lambda)$

$\mathcal{G}_0$



Proof of Security

$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$

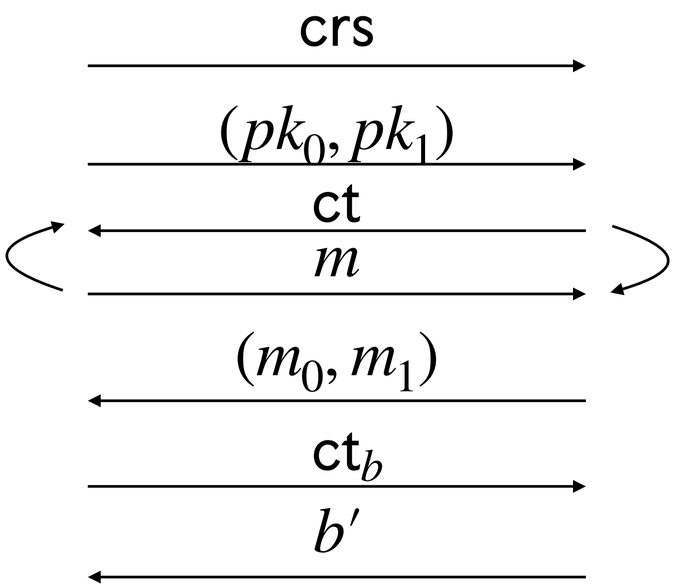
crs

$\mathcal{G}_0$

Proof of Security

$\mathcal{G}_0$

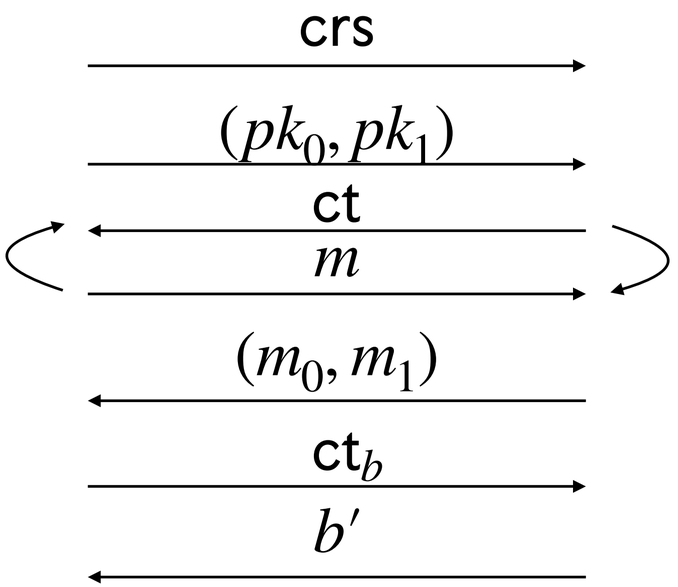
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_0$

$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$

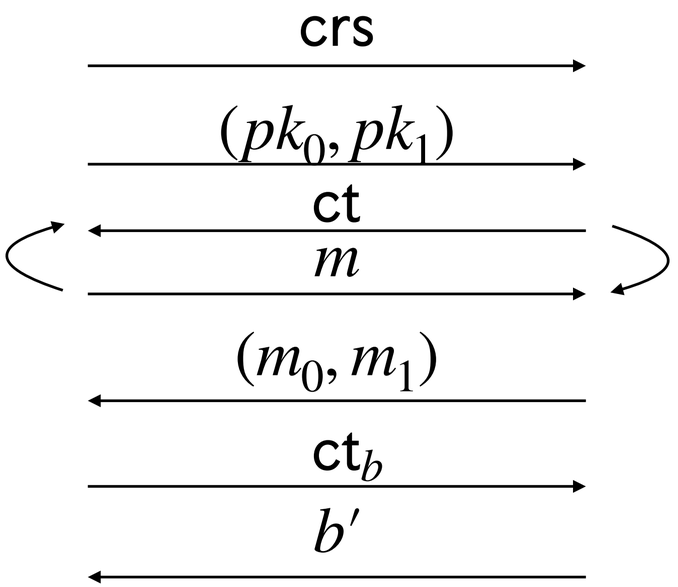


$\mathcal{G}_4$

Proof of Security

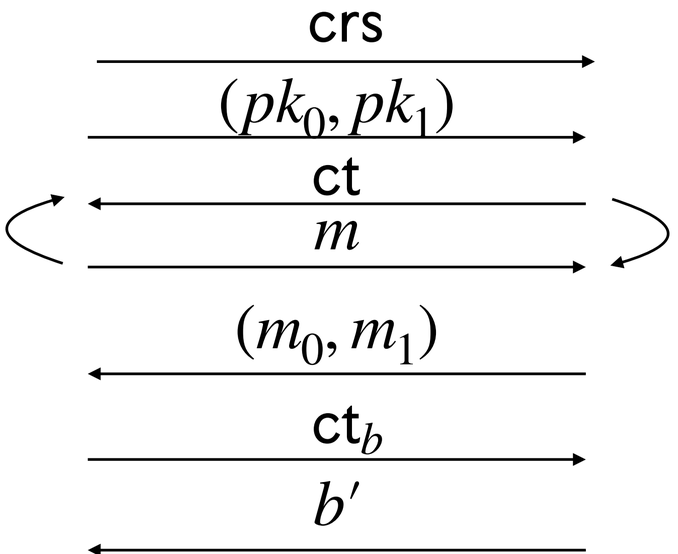
$\mathcal{G}_0$

$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$\mathcal{G}_4$

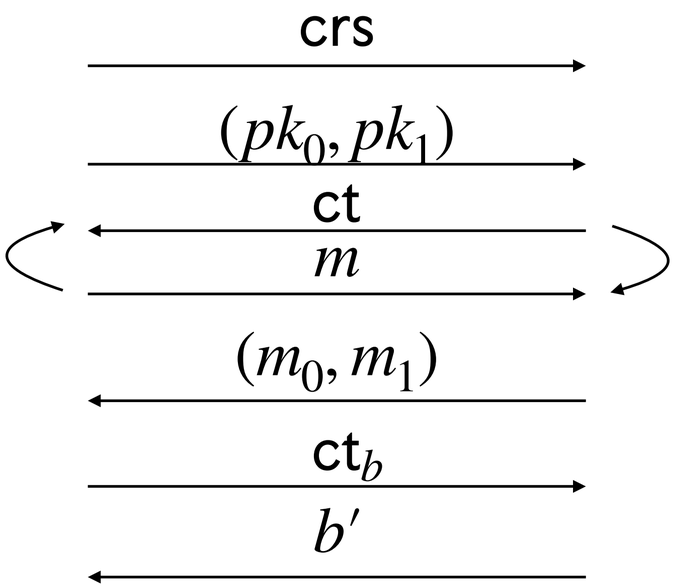
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_0$

$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$

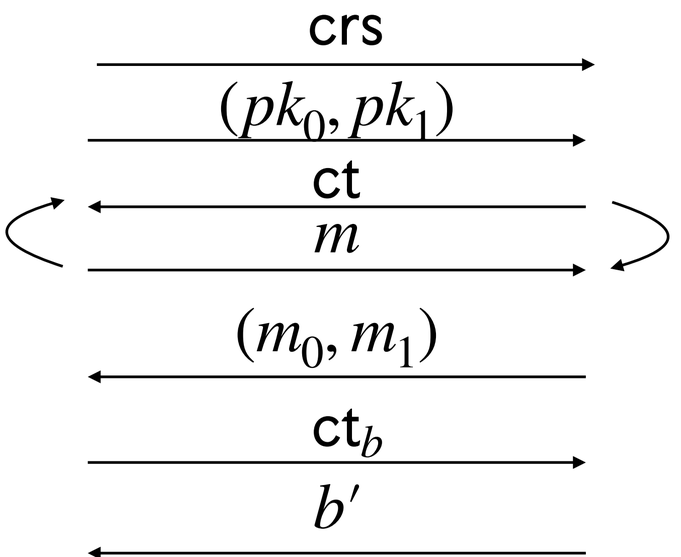


Goal:

$$\left|\Pr[\mathscr{A} \text{ outputs } 1 \text{ in } \mathcal{G}_0] - \Pr[\mathscr{A} \text{ outputs } 1 \text{ in } \mathcal{G}_4]\right| \leq \mathsf{negl}(\lambda)$$

$\mathcal{G}_4$

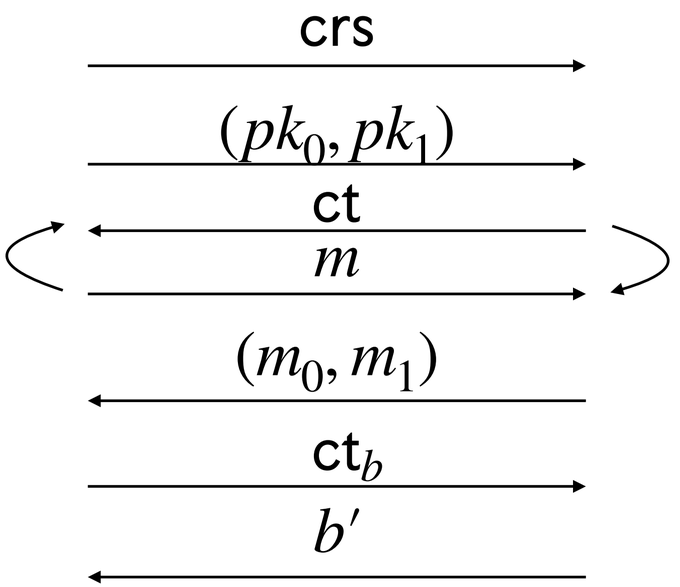
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_0$

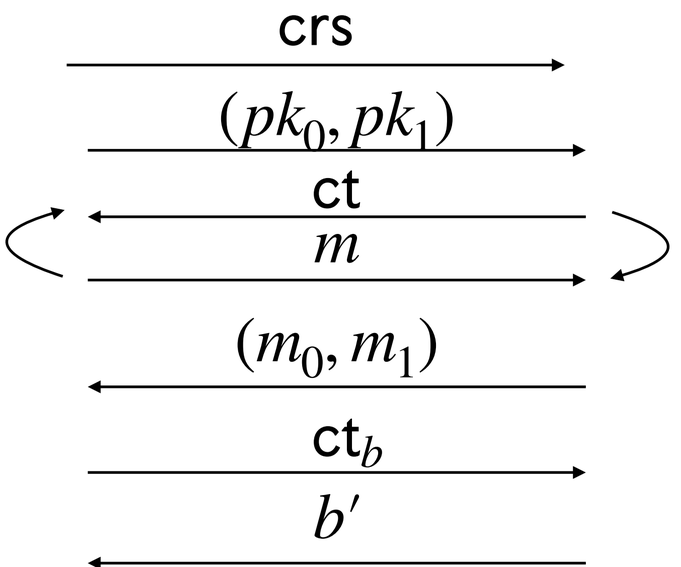
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Goal: $\left|\Pr[W_0] - \Pr[W_4]\right| \leq \mathsf{negl}(\lambda)$

$\mathcal{G}_4$

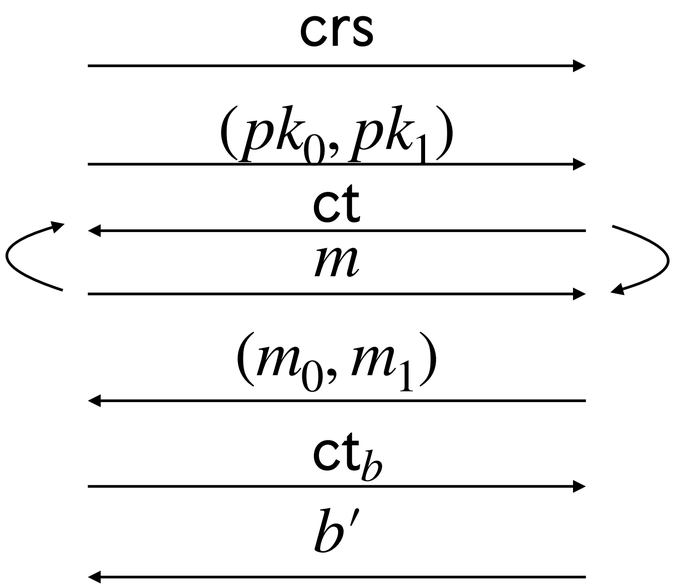
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_0$

$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$

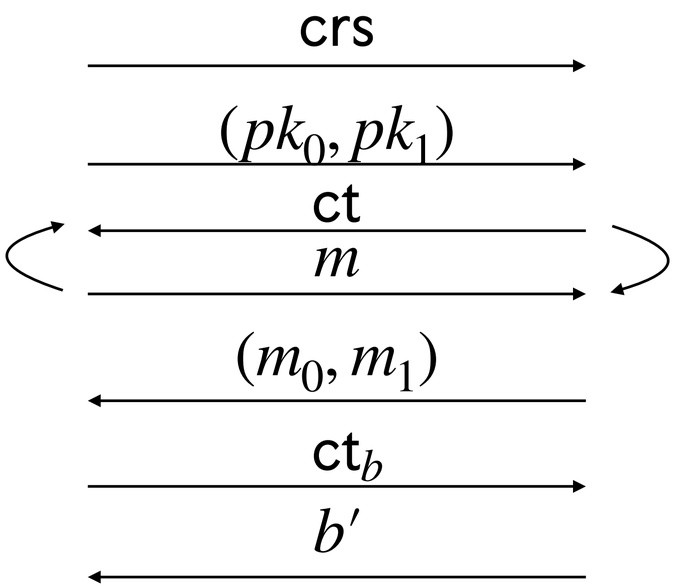


$\mathcal{G}_1$

Proof of Security

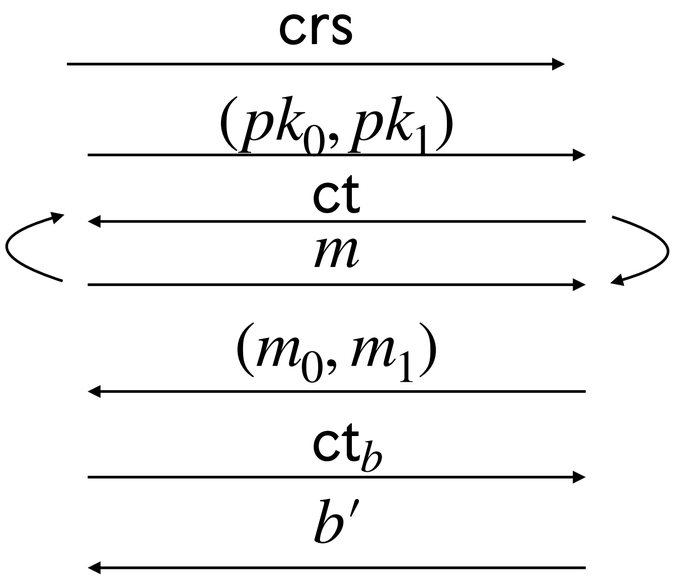
$\mathcal{G}_0$

$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$\mathcal{G}_1$

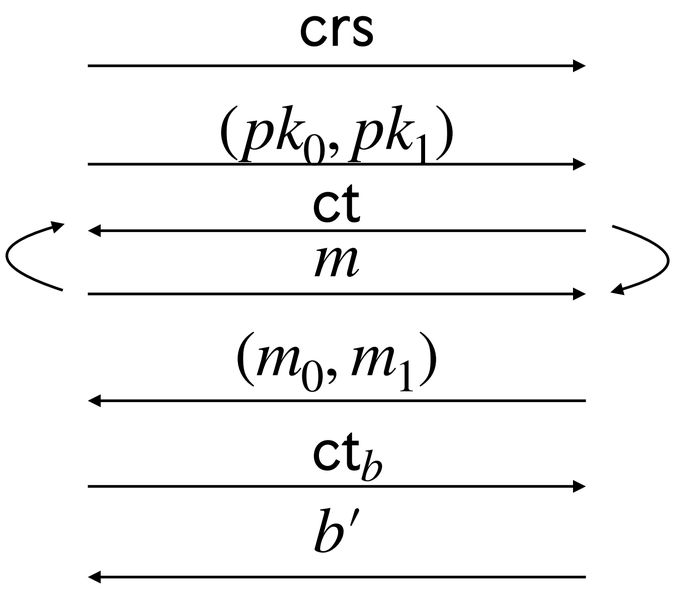
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_0$

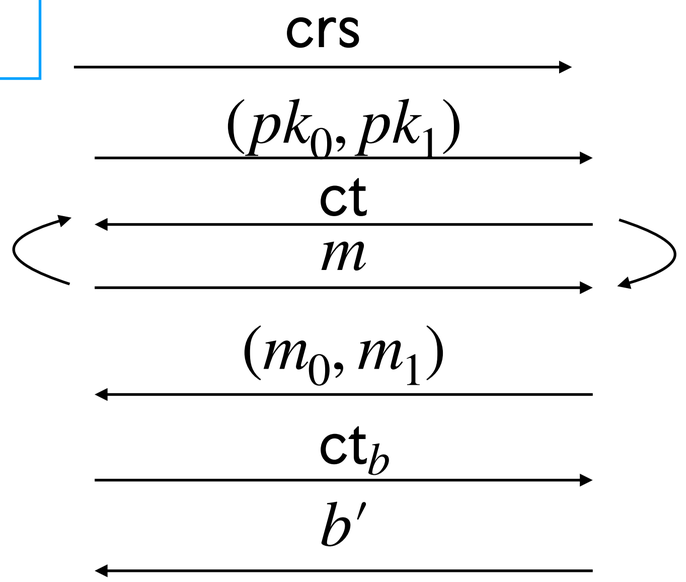
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Claim: $\left|\Pr[W_0] - \Pr[W_1]\right| \leq \mathsf{negl}(\lambda)$ by the ZK of (P, V)

$\mathcal{G}_1$

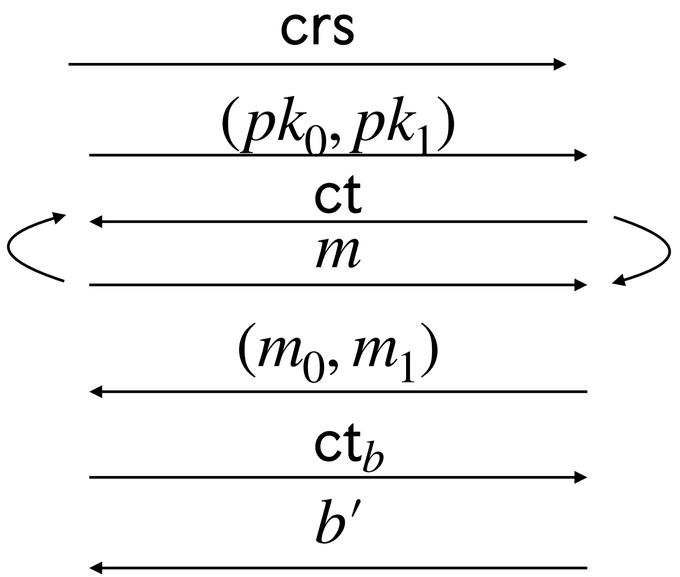
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_1$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$

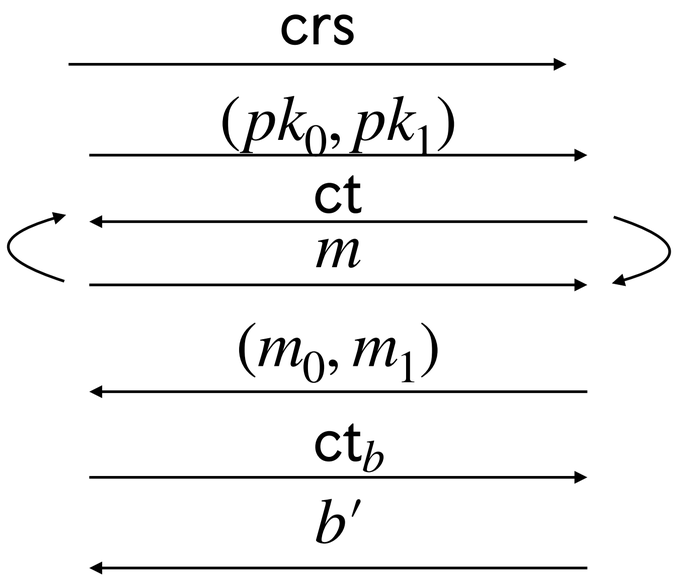


$\mathcal{G}_2$

Proof of Security

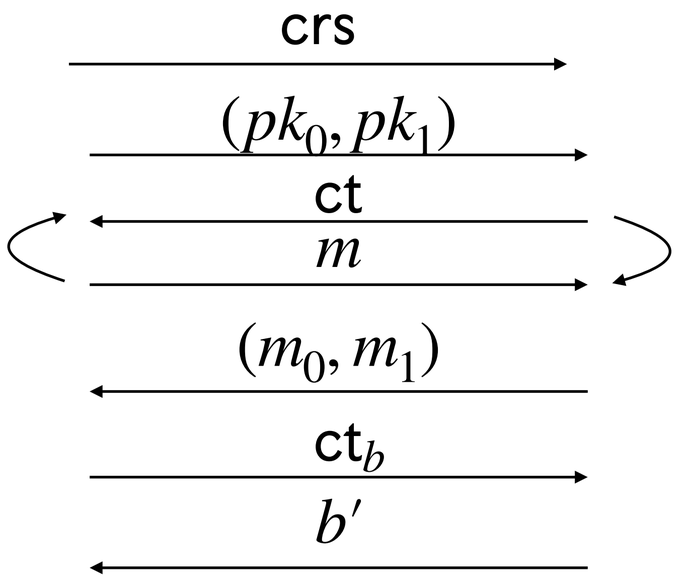
$\mathcal{G}_1$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$\mathcal{G}_2$

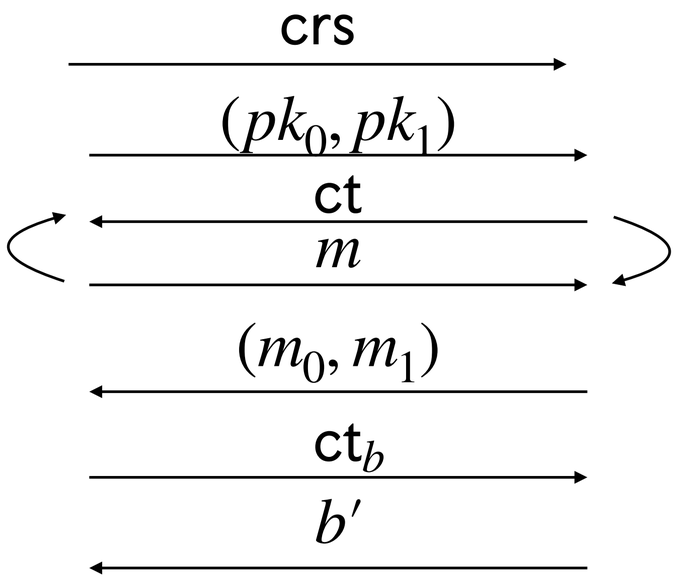
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_1$

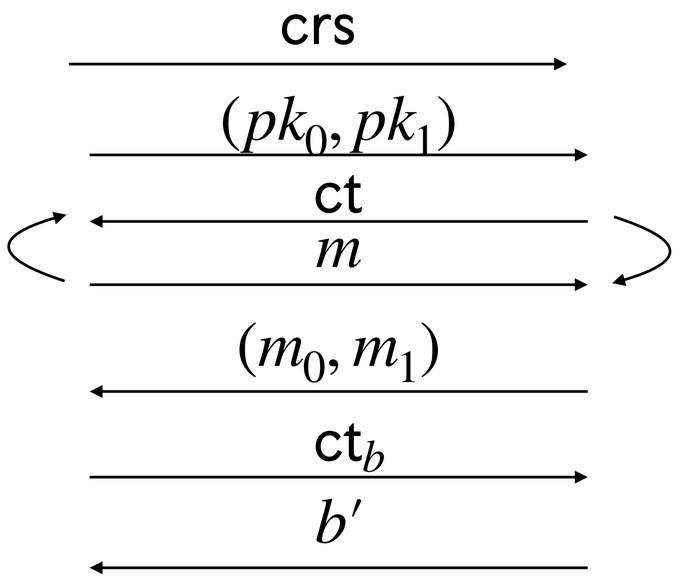
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA

$\mathcal{G}_2$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

Proof of Security

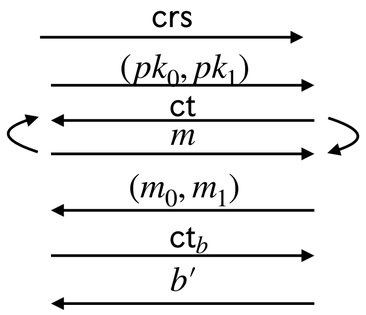
$\mathsf{crs} \leftarrow S_0(1^\lambda)$

$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$

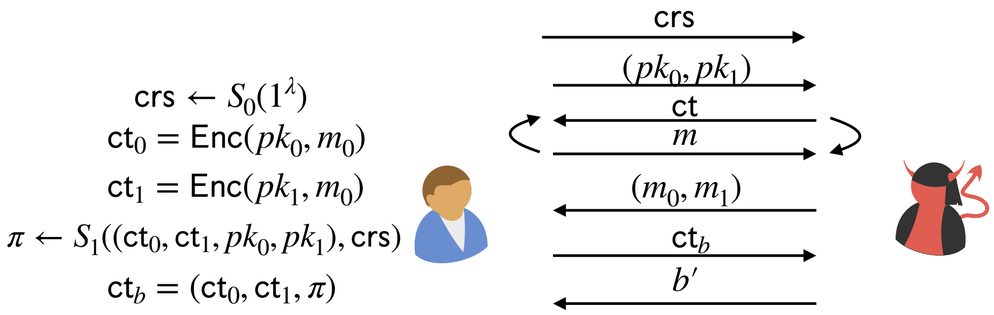
$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$

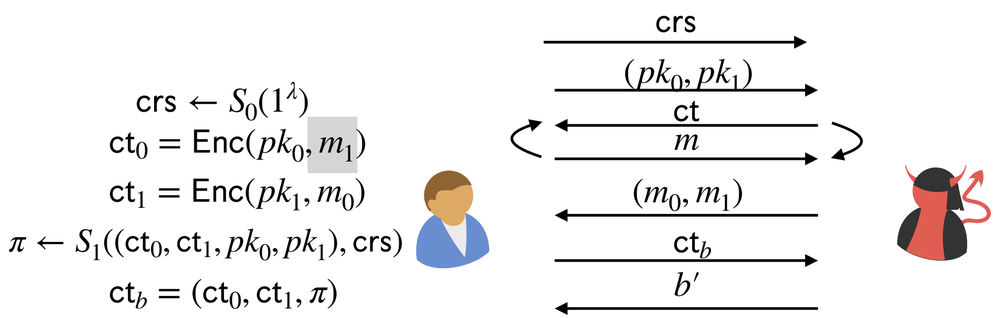
$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$

$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$



Proof of Security

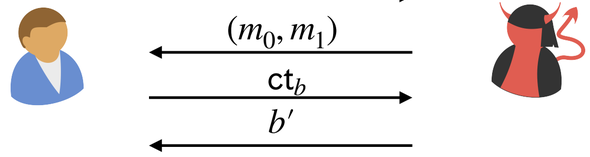




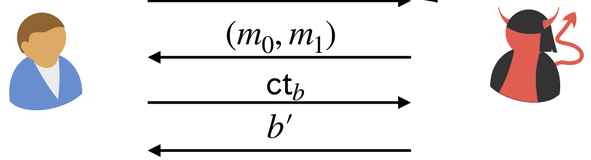
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathbf{negl}(\lambda)$ by IND-CPA

$\mathsf{crs} \leftarrow S_0(1^\lambda)$
$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_0)$
$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$
$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$
$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$

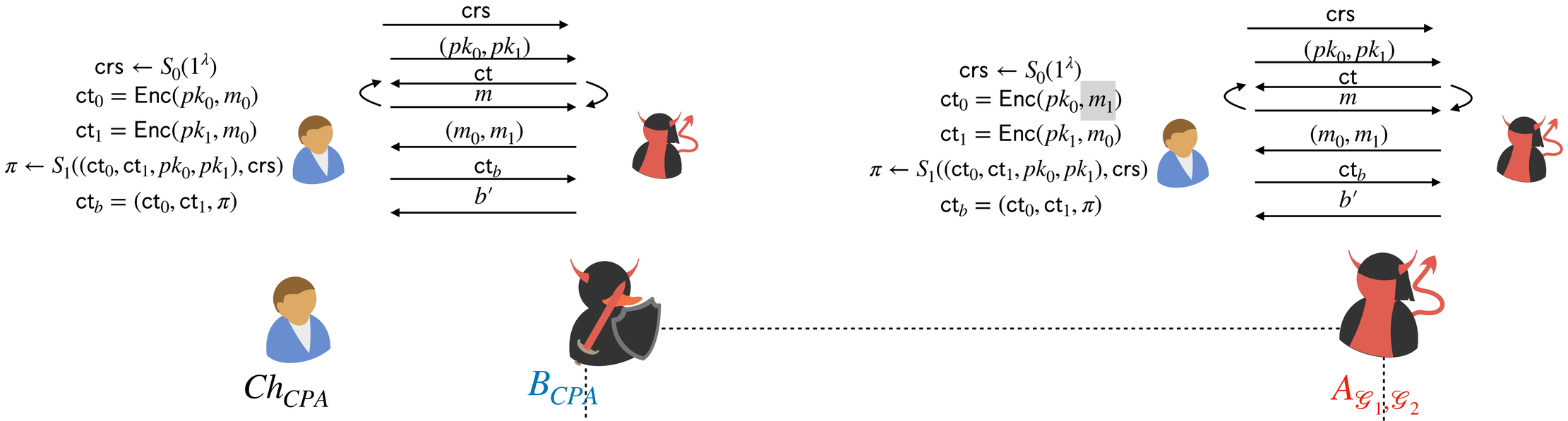


$\mathsf{crs} \leftarrow S_0(1^\lambda)$
$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$
$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$
$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$
$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$



Proof of Security

Claim: $\left|\mathsf{Pr}[W_1] - \mathsf{Pr}[W_2]\right| \leq \mathbf{negl}(\lambda)$ by IND-CPA



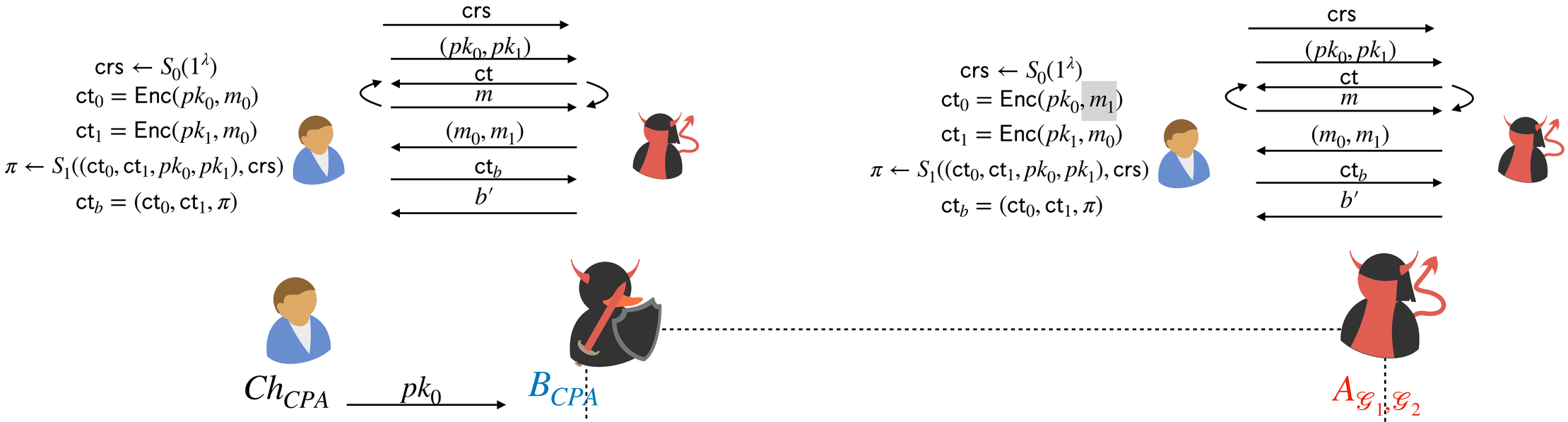
Ch_{CPA}

B_{CPA}

$A_{\mathcal{G}_1,\mathcal{G}_2}$

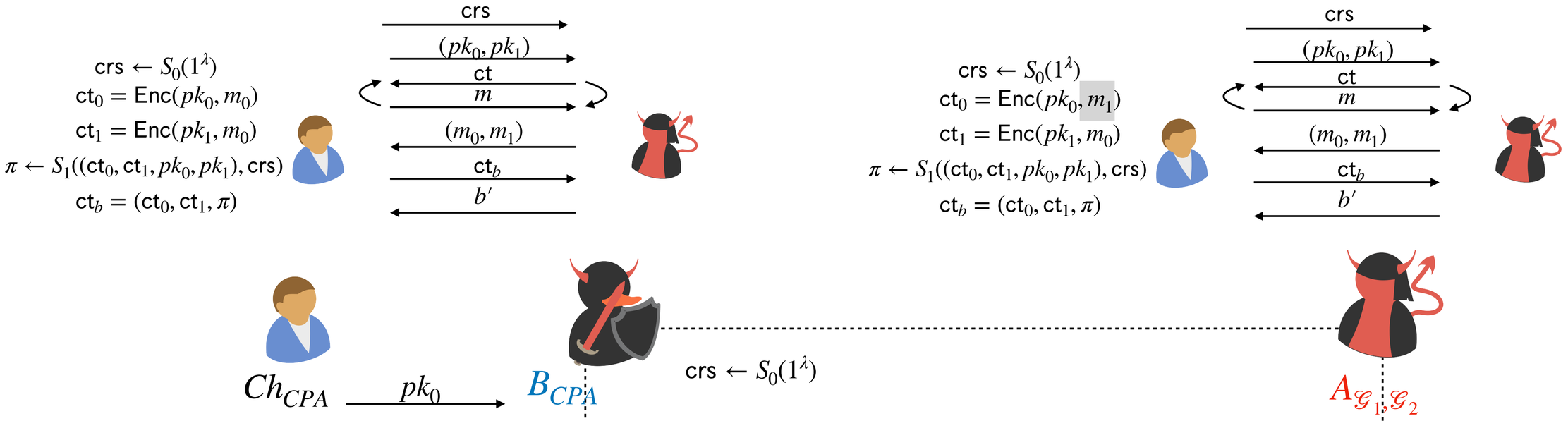
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



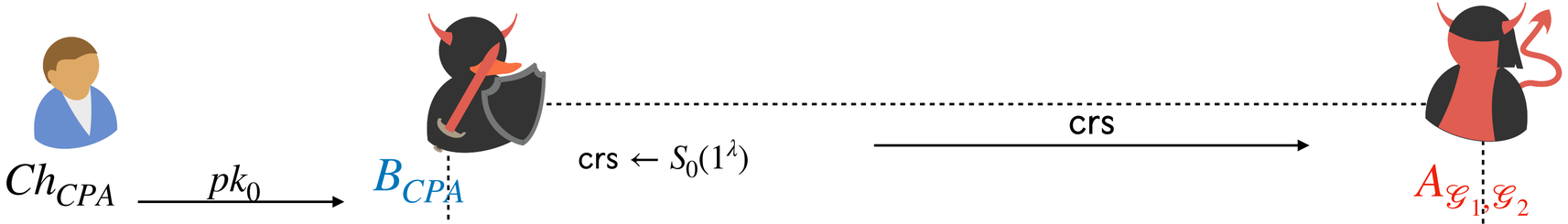
Proof of Security

Claim: $\left|\mathsf{Pr}[W_1] - \mathsf{Pr}[W_2]\right| \leq \mathbf{negl}(\lambda)$ by IND-CPA



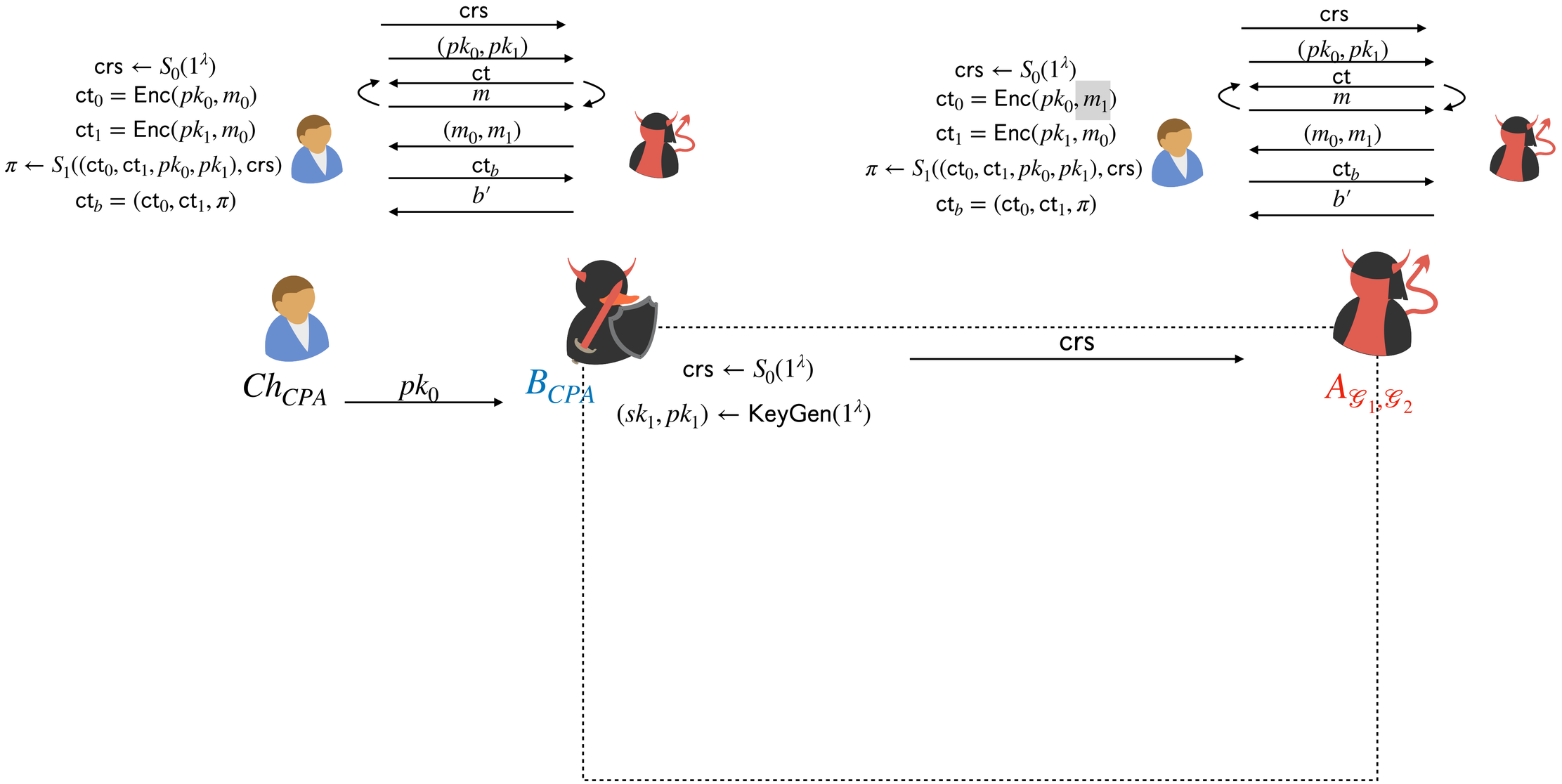
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



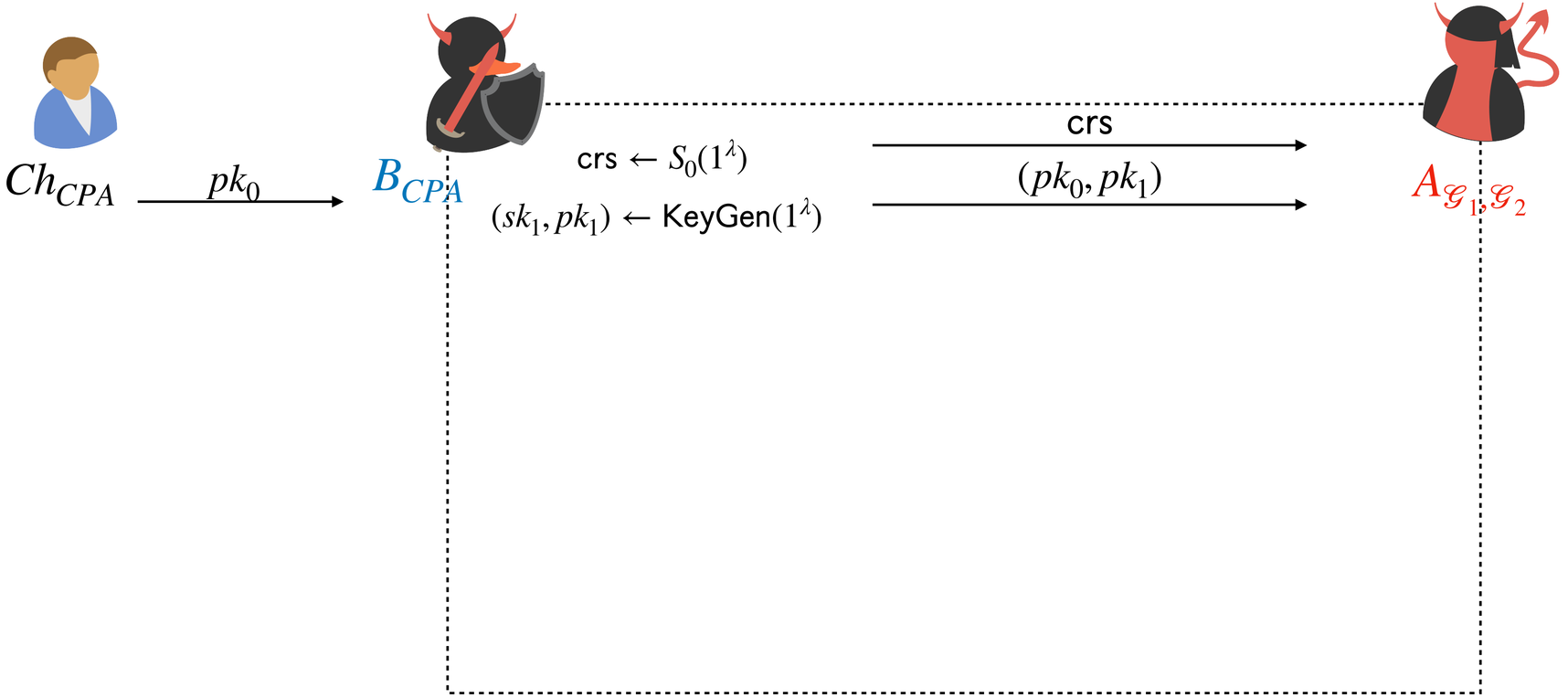
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



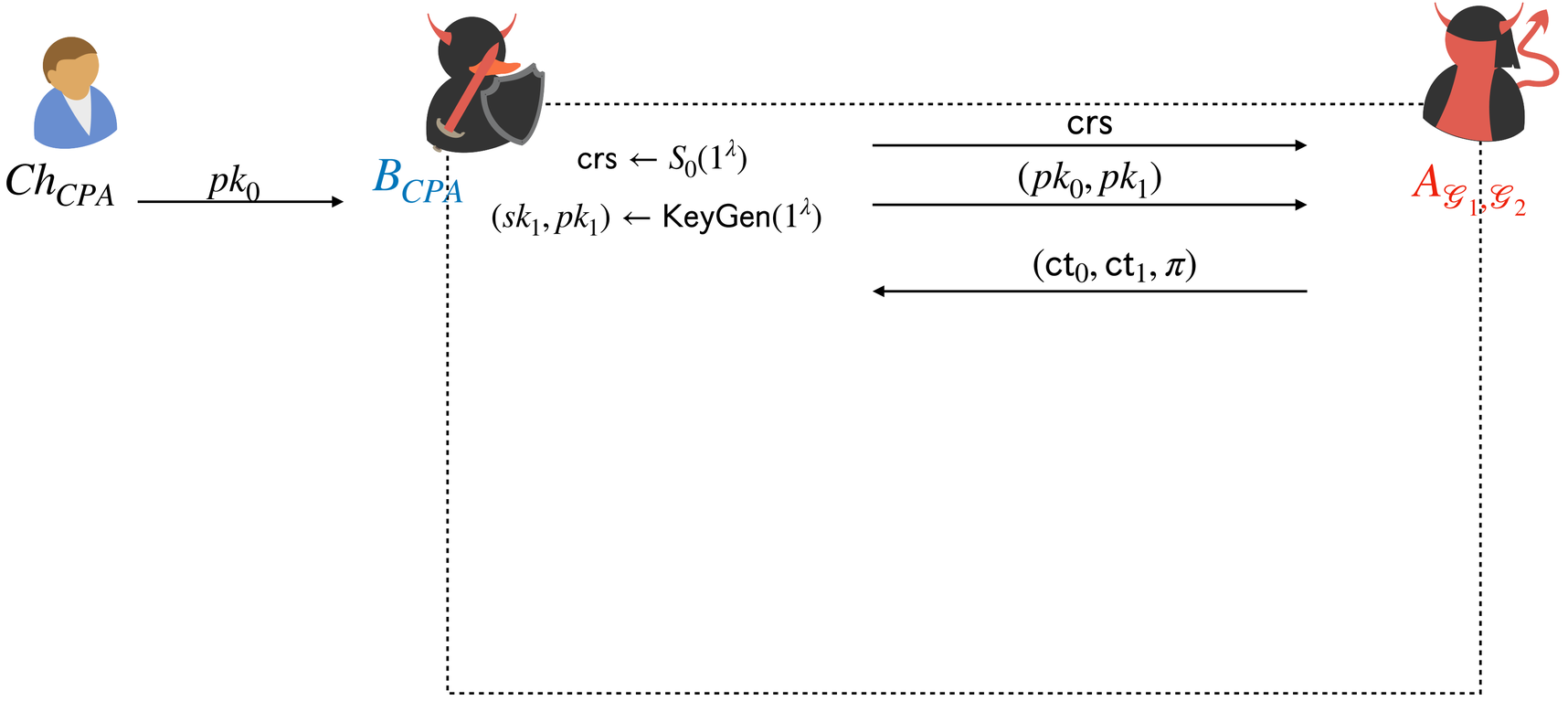
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



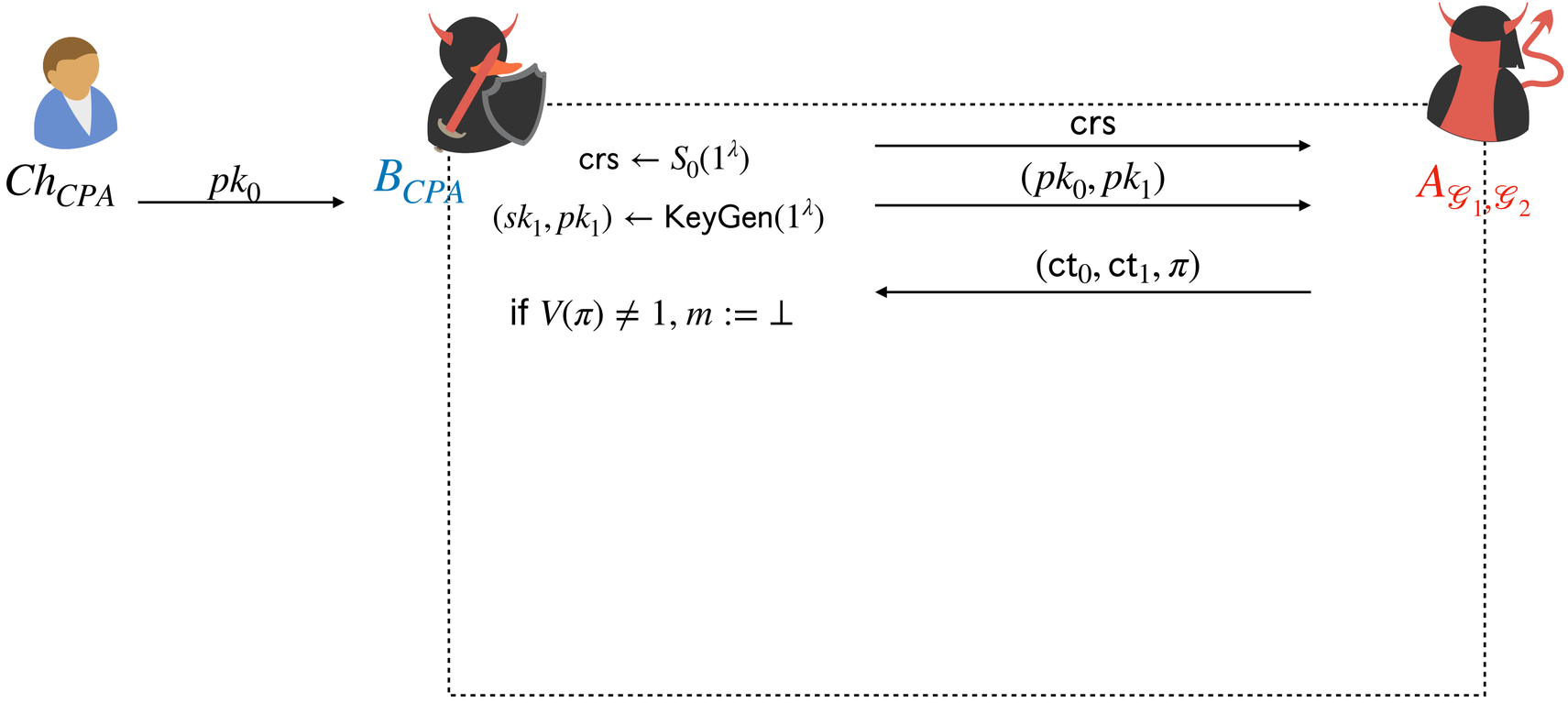
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



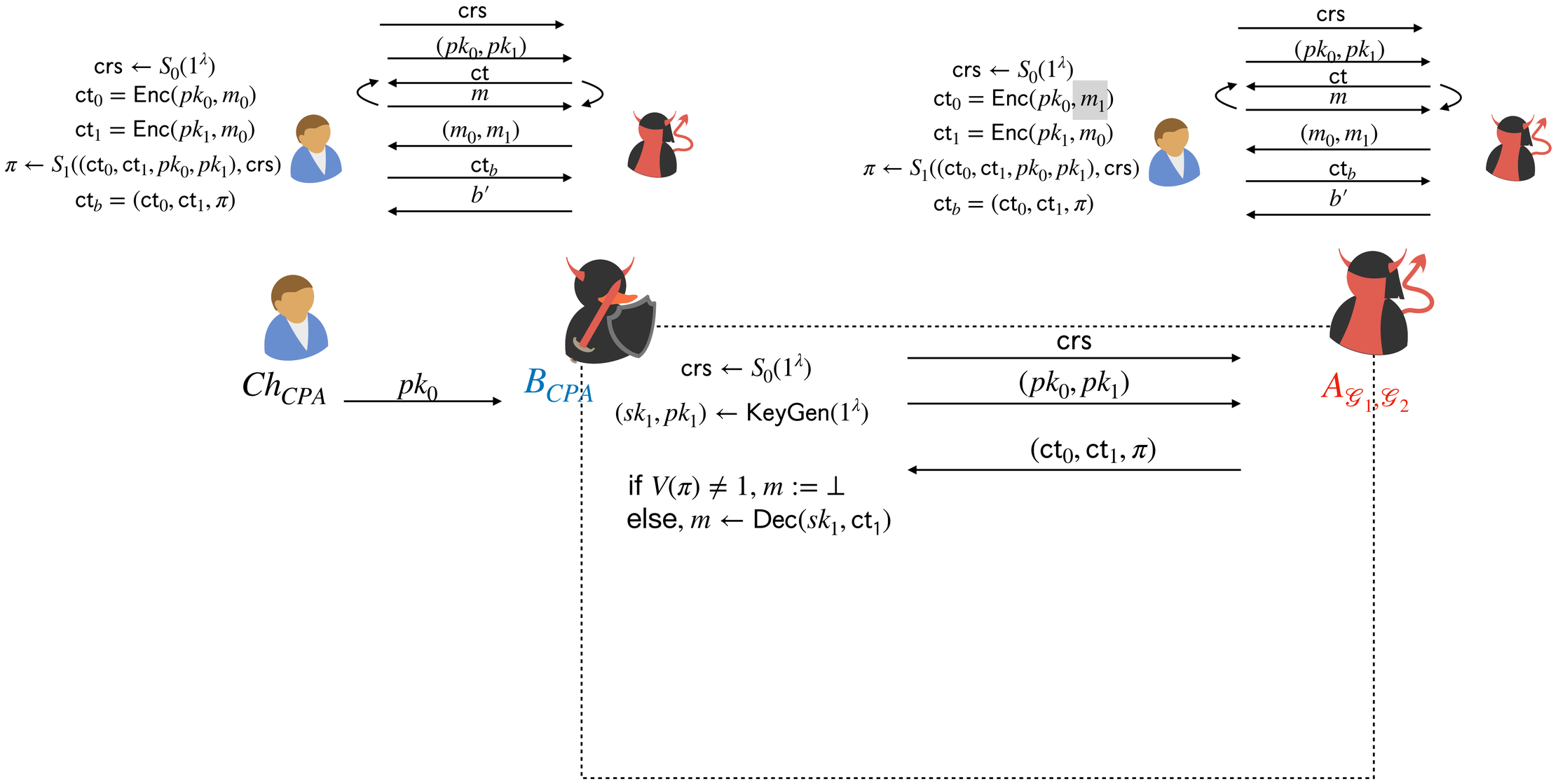
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



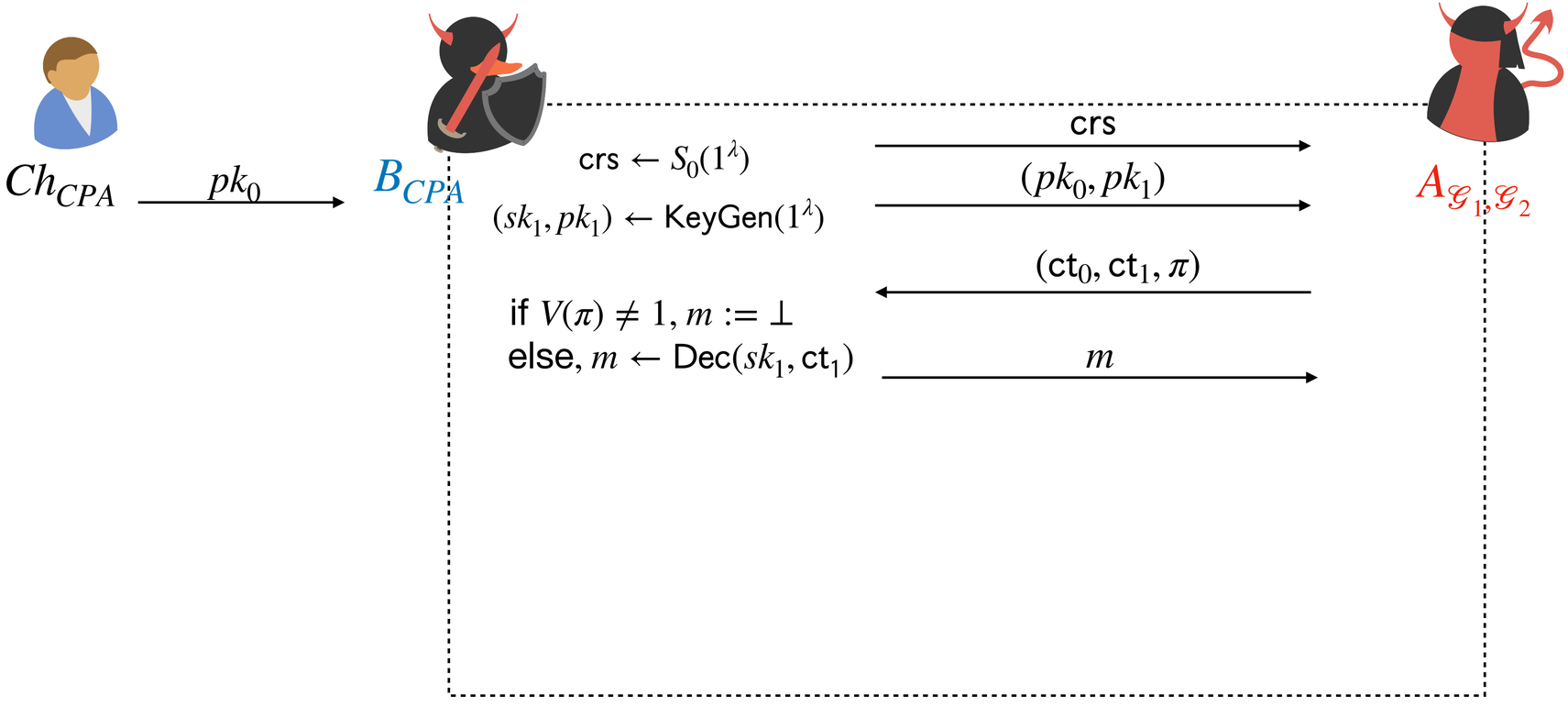
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



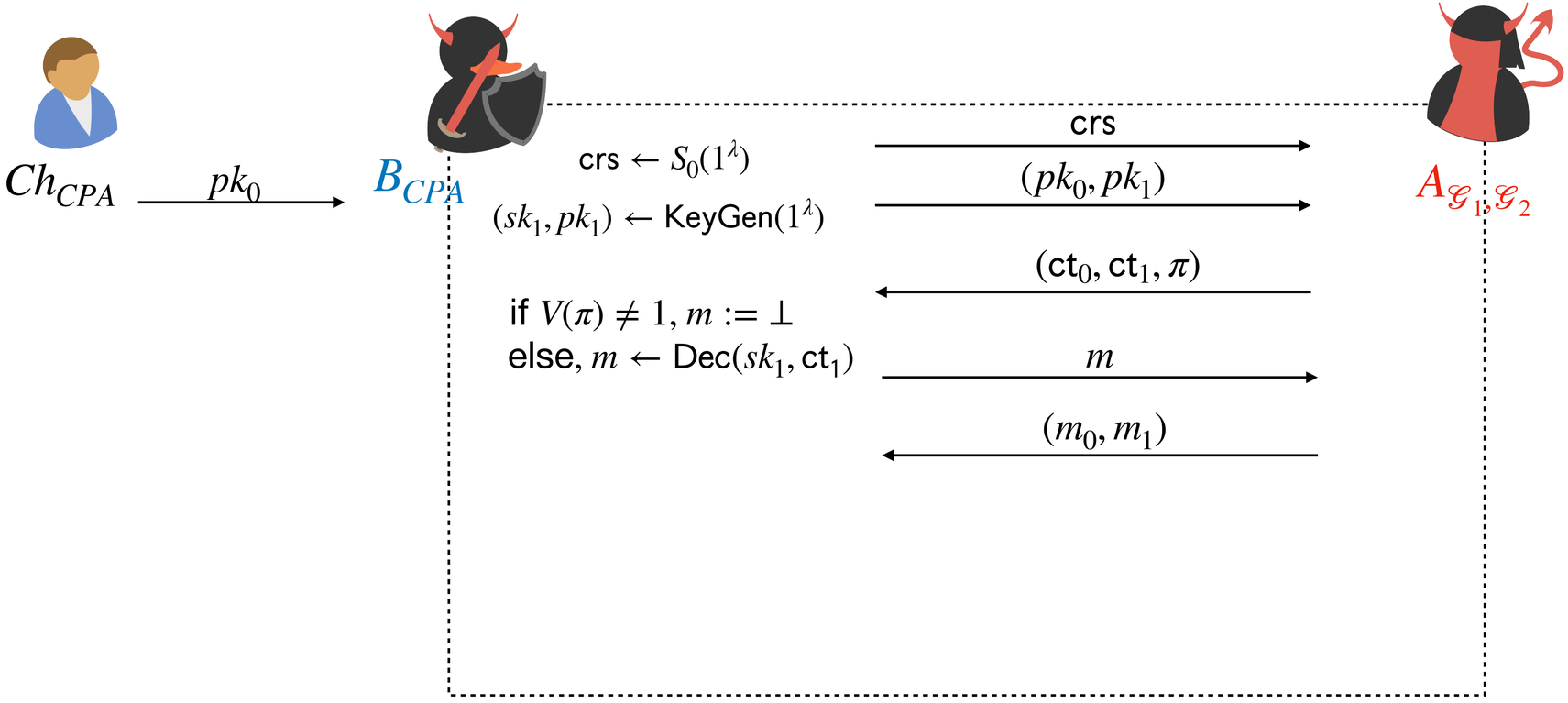
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



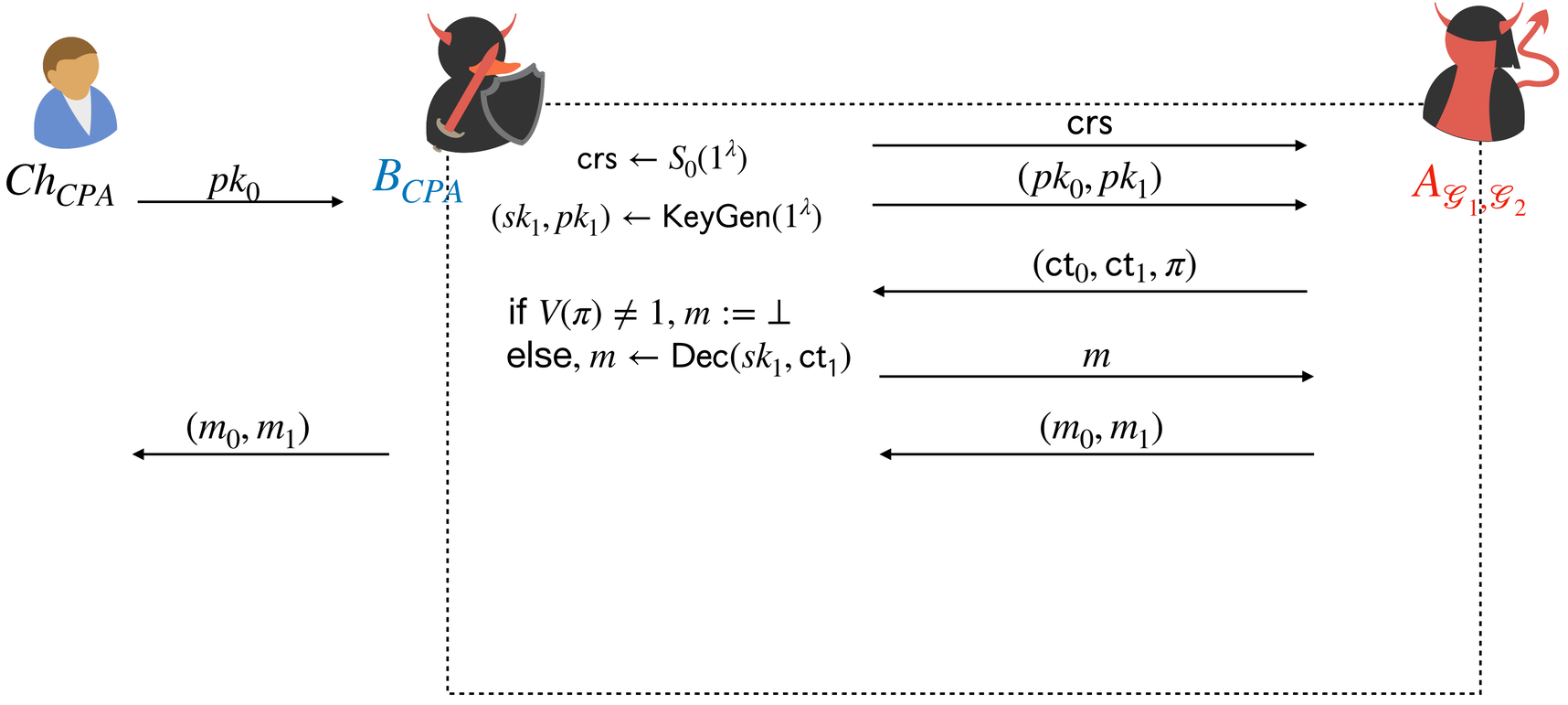
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



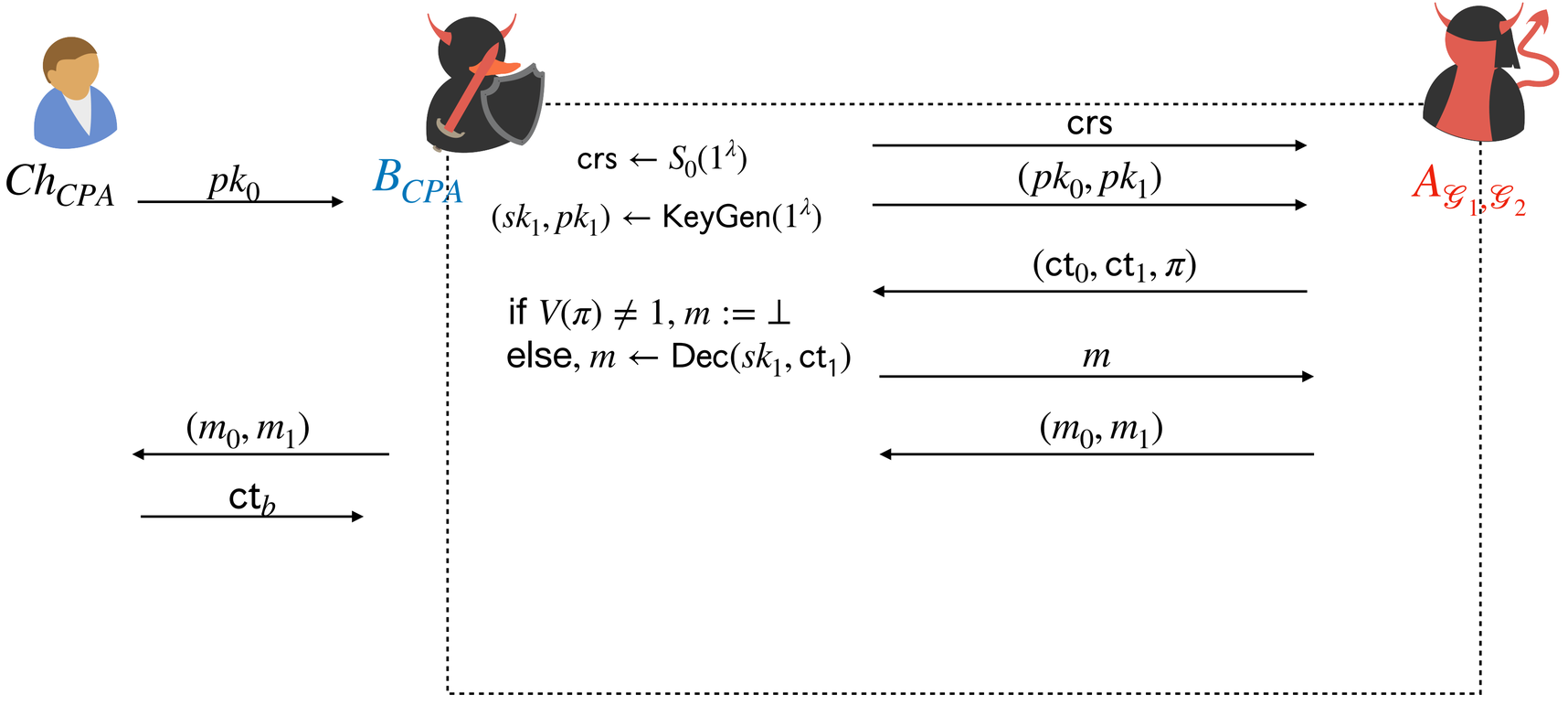
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



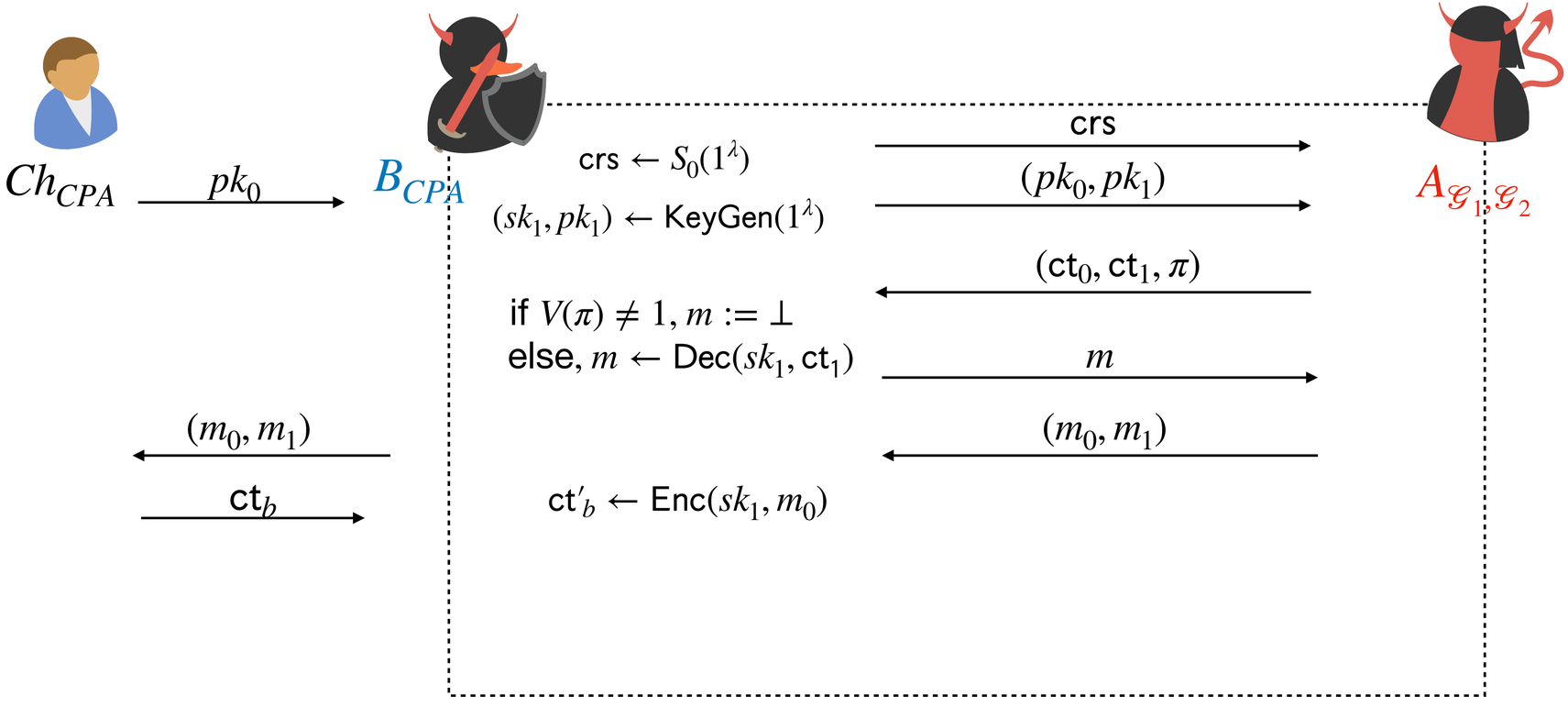
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



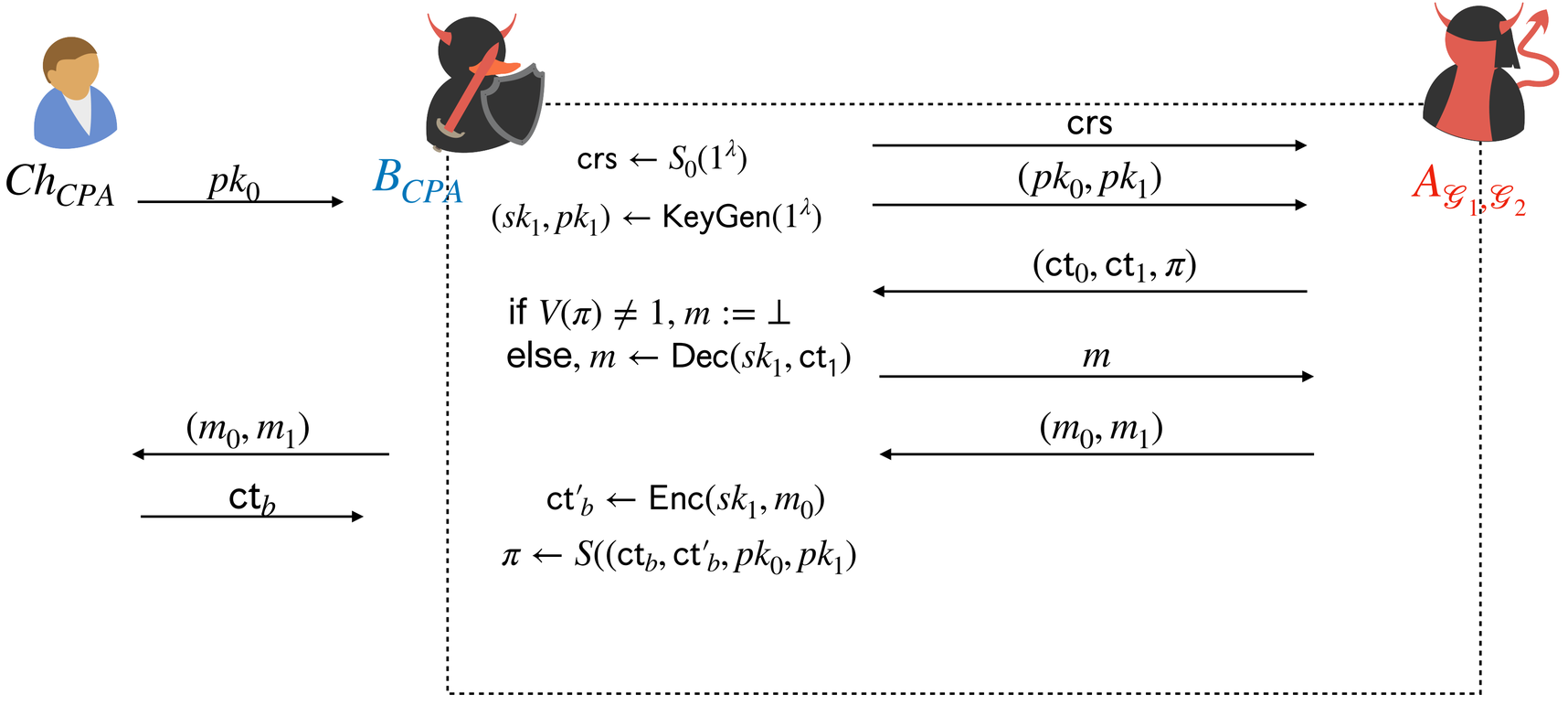
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



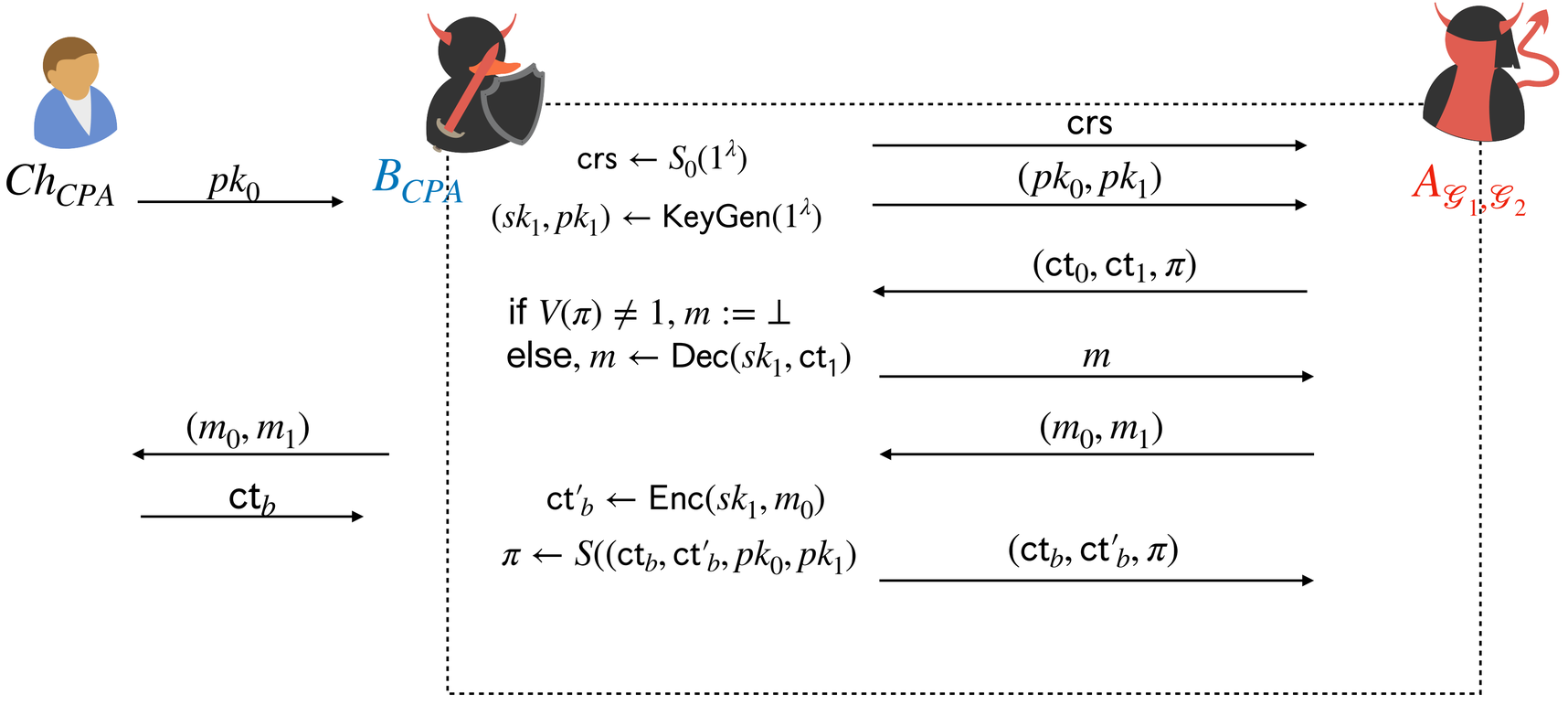
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



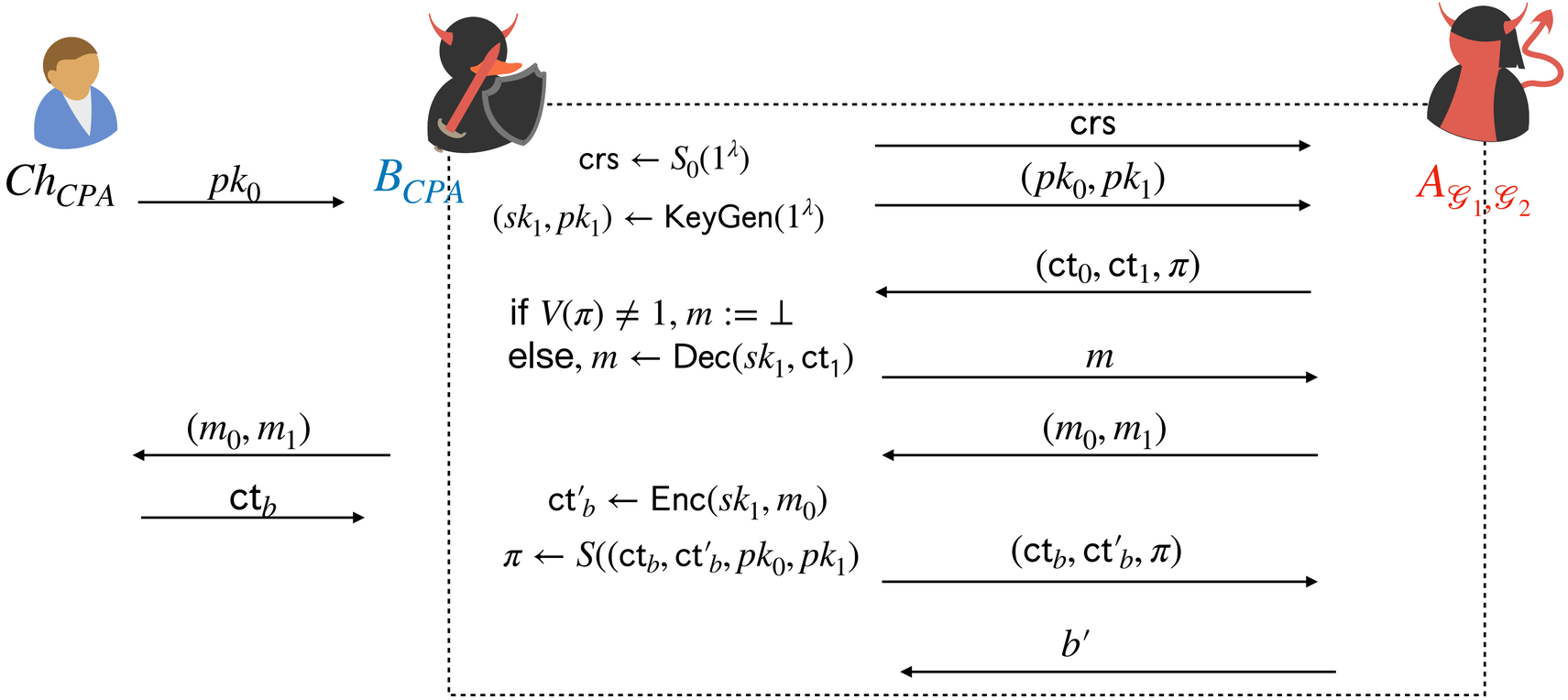
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



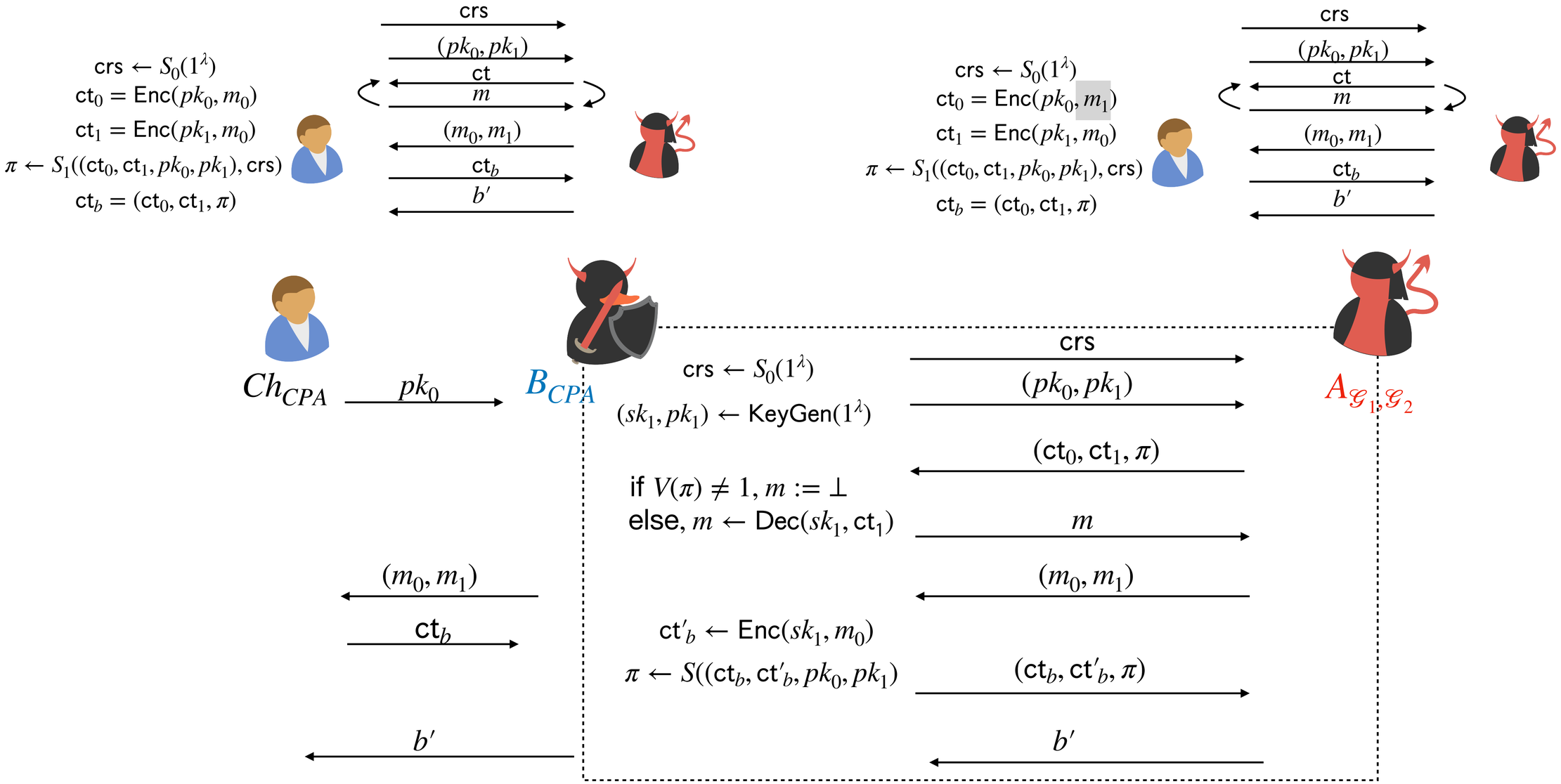
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



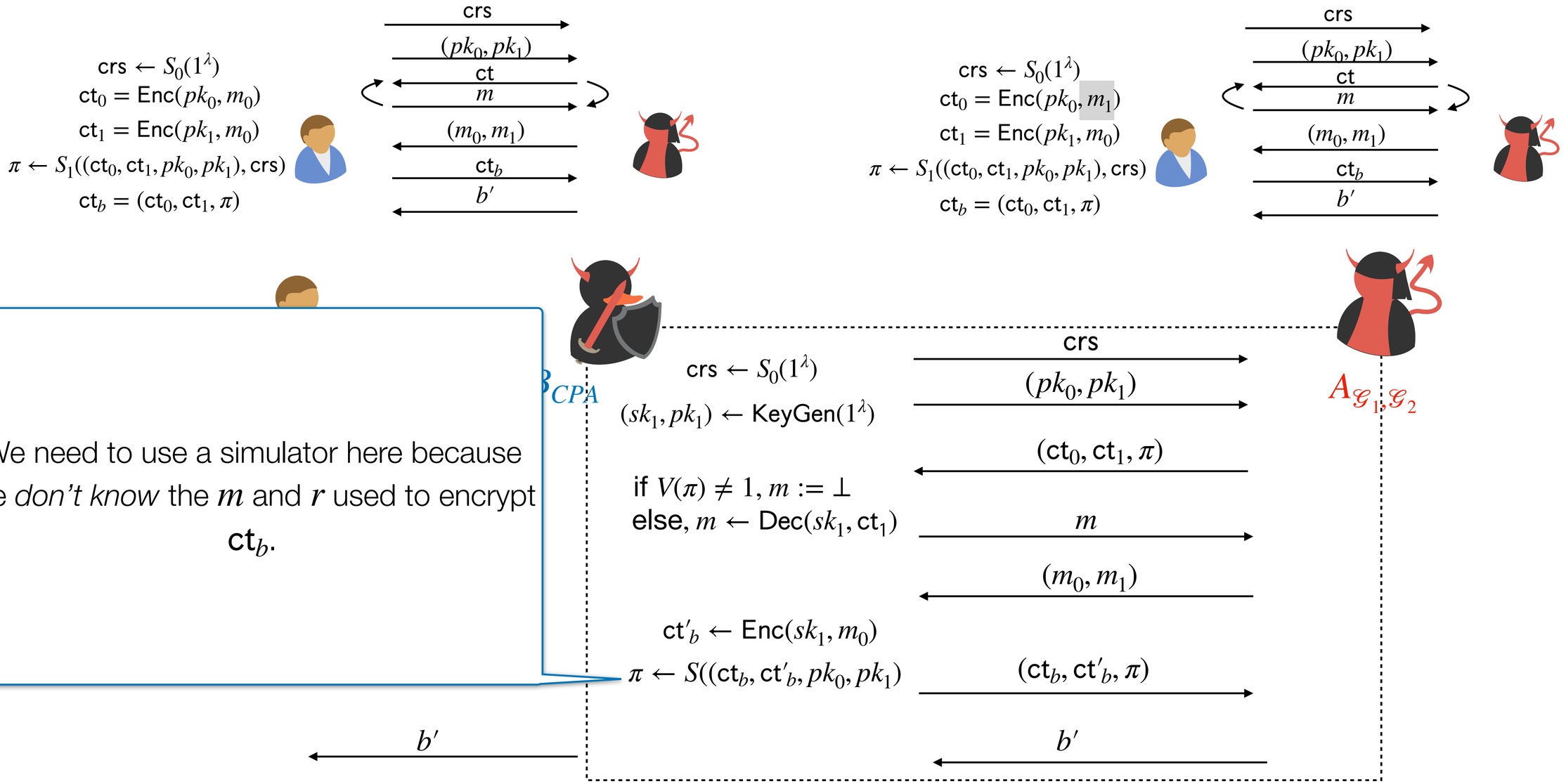
Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



Proof of Security

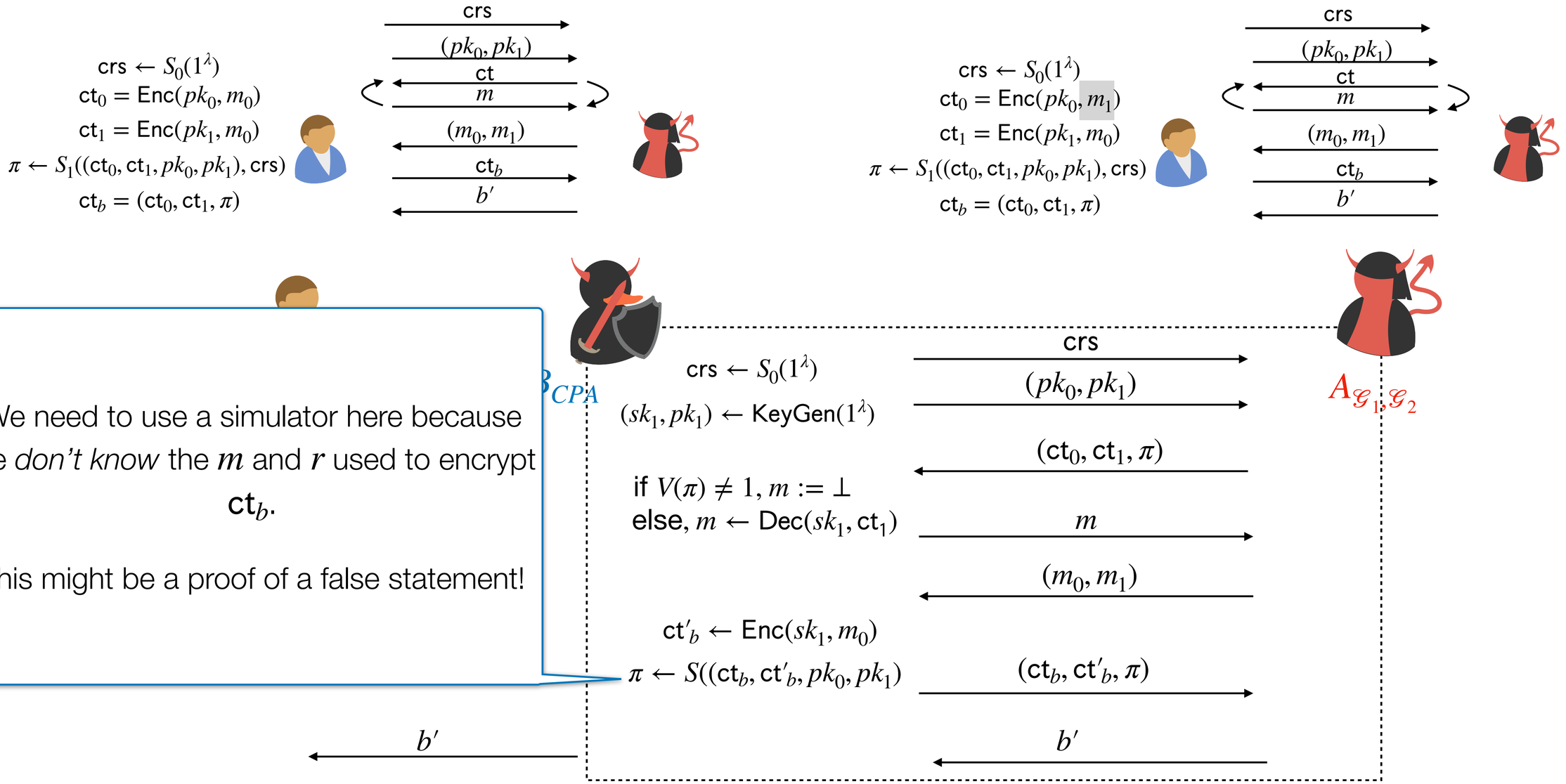
Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



We need to use a simulator here because we *don't know* the m and r used to encrypt ct_b.

Proof of Security

Claim: $\left|\Pr[W_1] - \Pr[W_2]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA



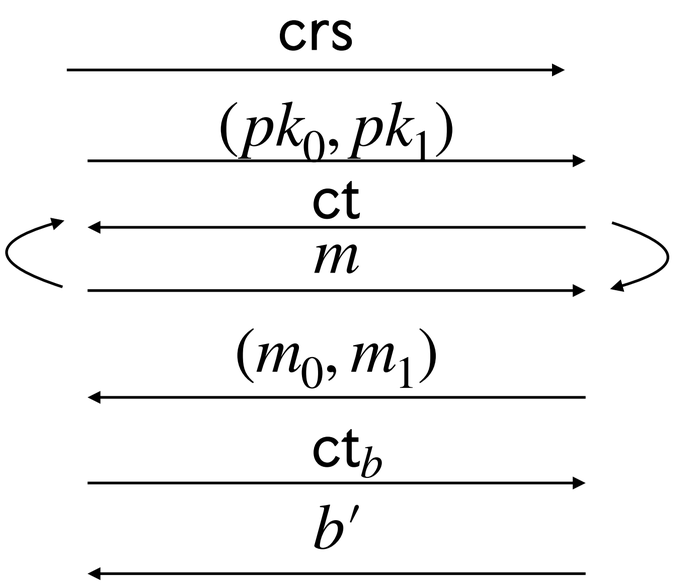
We need to use a simulator here because we *don't know* the m and r used to encrypt ct_b.

This might be a proof of a false statement!

Proof of Security

$\mathcal{G}_2$

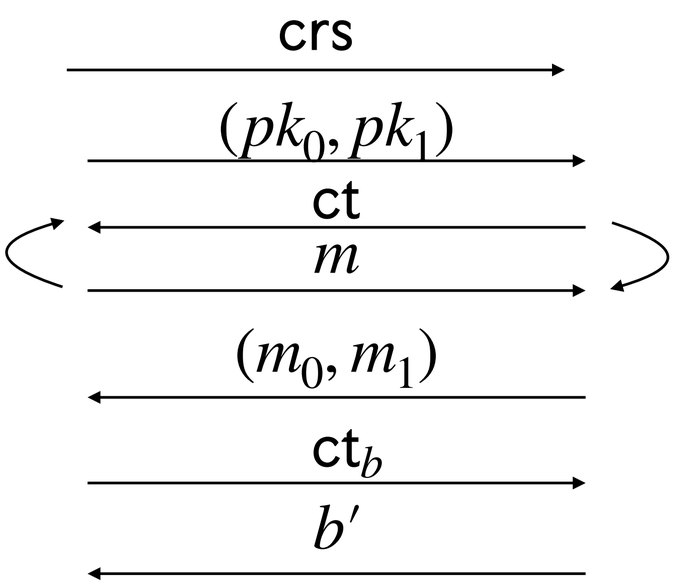
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_2$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$

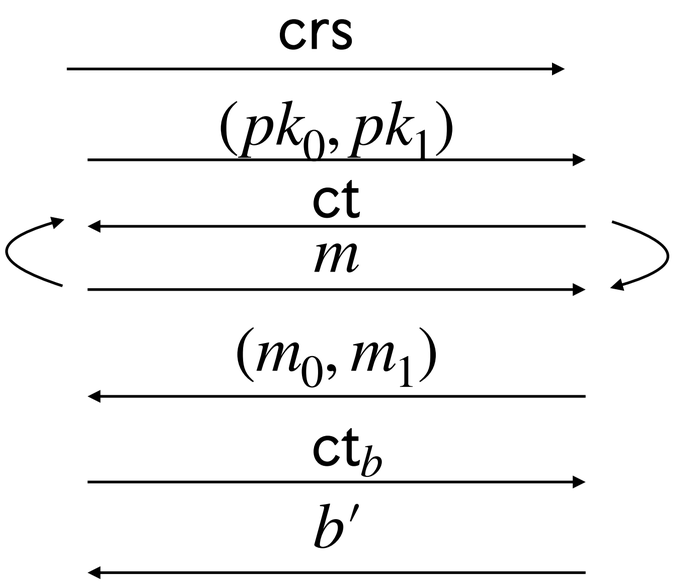


$\mathcal{G}_3$

Proof of Security

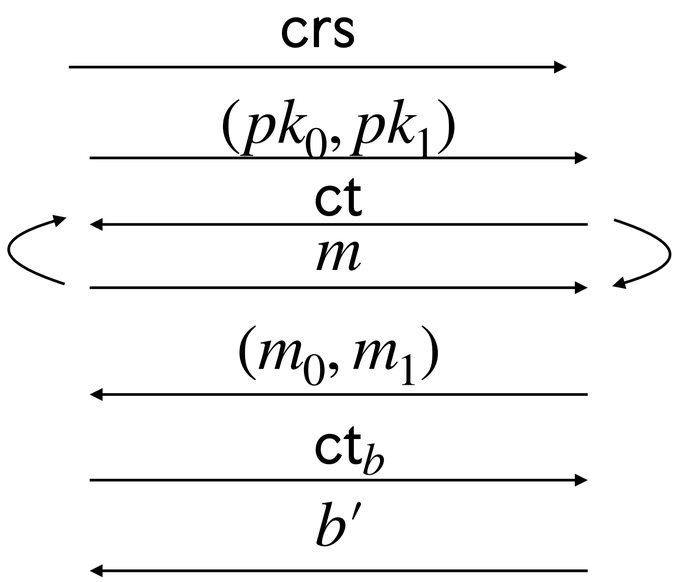
$\mathcal{G}_2$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$\mathcal{G}_3$

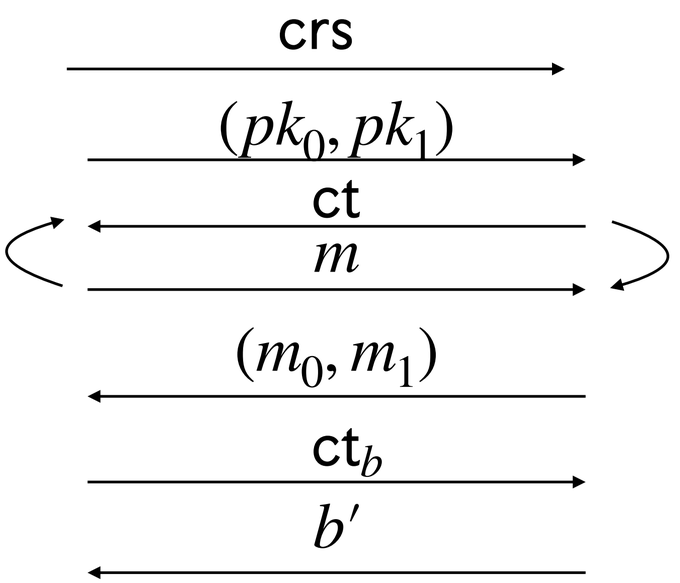
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_2$

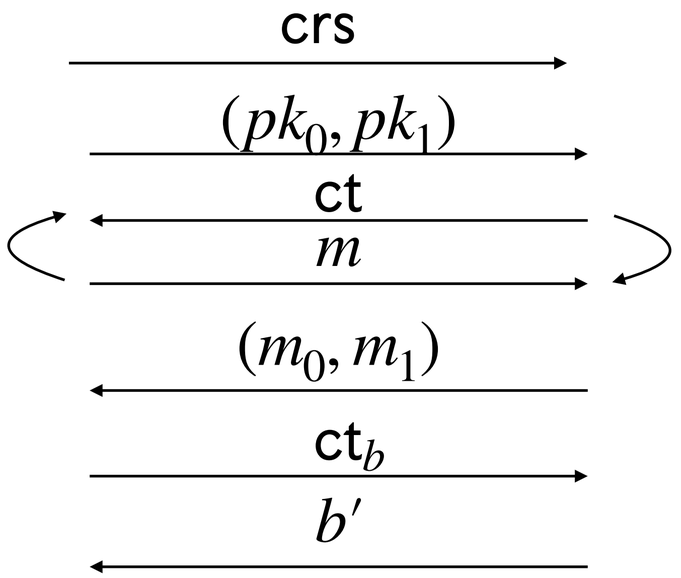
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_0)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$\left|\Pr[W_2] - \Pr[W_3]\right| \leq \mathsf{negl}(\lambda)$ by IND-CPA

$\mathcal{G}_3$

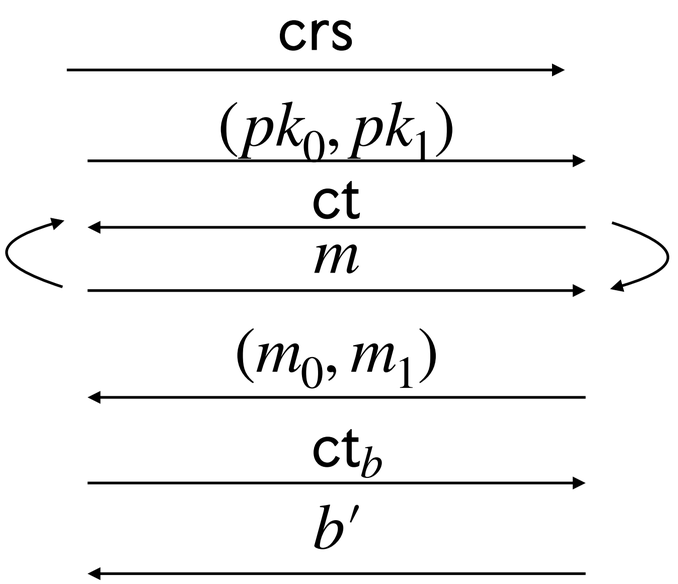
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_3$

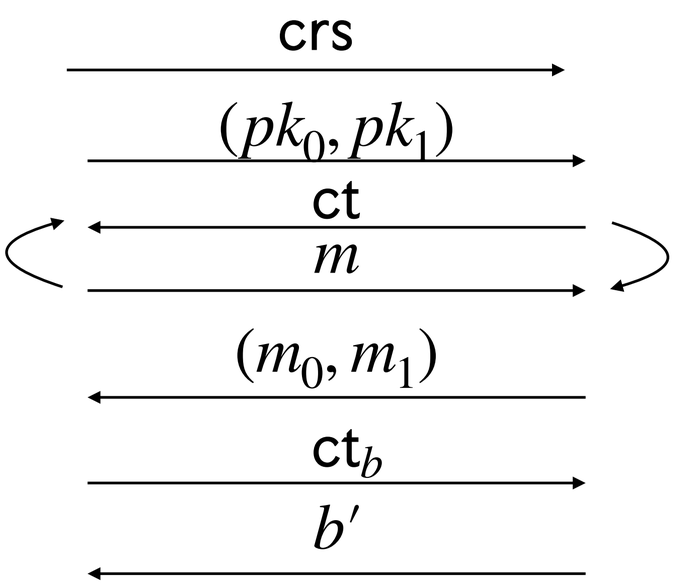
$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_3$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$

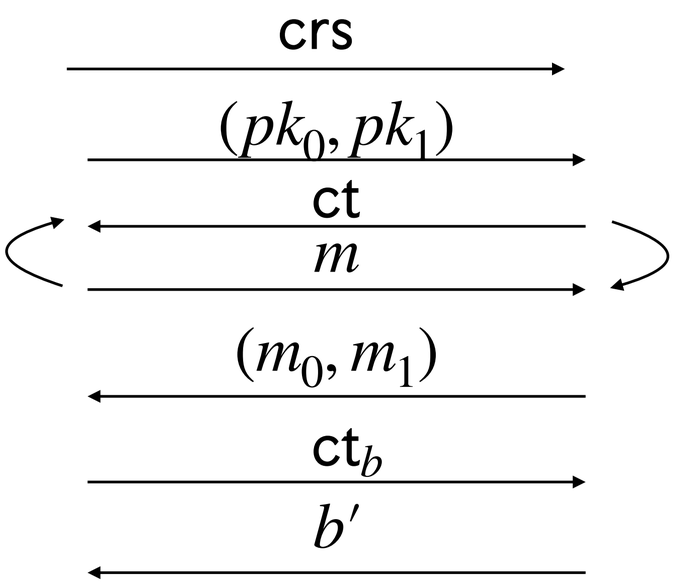


$\mathcal{G}_4$

Proof of Security

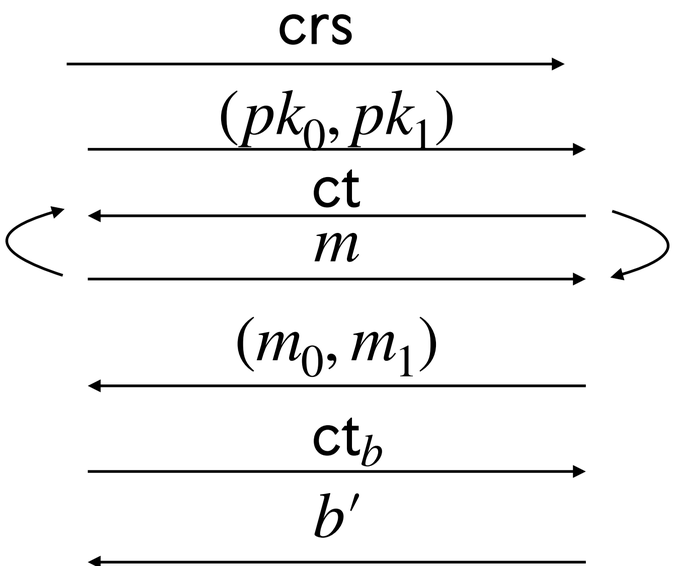
$\mathcal{G}_3$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$\mathcal{G}_4$

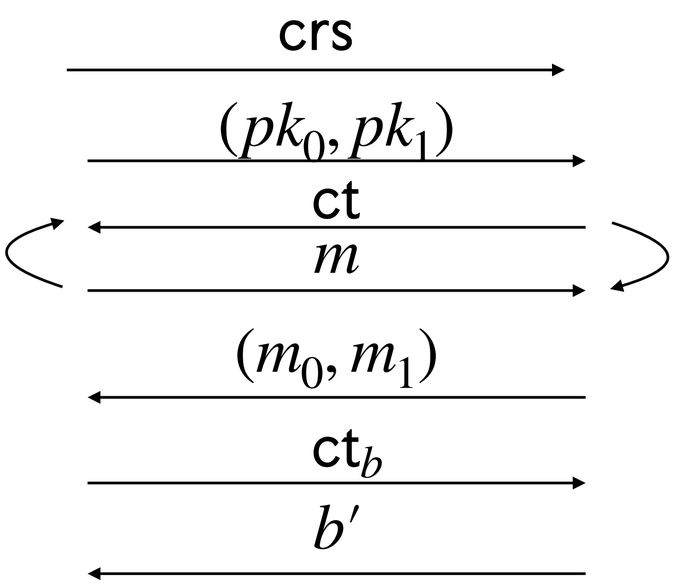
$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



Proof of Security

$\mathcal{G}_3$

$$\mathsf{crs} \leftarrow S_0(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow S_1((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$



$$\left|\Pr[W_3] - \Pr[W_4]\right| \leq \mathsf{negl}(\lambda) \text{ by ZK}$$

$\mathcal{G}_4$

$$\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda)$$
$$\mathsf{ct}_0 = \mathsf{Enc}(pk_0, m_1)$$
$$\mathsf{ct}_1 = \mathsf{Enc}(pk_1, m_1)$$
$$\pi \leftarrow P((\mathsf{ct}_0, \mathsf{ct}_1, pk_0, pk_1), \mathsf{crs})$$
$$\mathsf{ct}_b = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$$