CCA Security I

601.442/642 Modern Cryptography

9th April 2026

Logistics

Logistics

- Midterm II done, congratulations!

Logistics

- Midterm II done, congratulations!
  - Grades out soon

Logistics

- Midterm II done, congratulations!
  - Grades out soon
- HW 8 out today.

Logistics

- Midterm II done, congratulations!
  - Grades out soon
- HW 8 out today.
- Extra-credit HW out next week.

Agenda

Agenda

- Recap IND-CPA

Agenda

- Recap IND-CPA
- Discuss how it may not be sufficient in practice

Agenda

- Recap IND-CPA
- Discuss how it may not be sufficient in practice
- Introduce stronger IND-CCA security

Agenda

- Recap IND-CPA
- Discuss how it may not be sufficient in practice
- Introduce stronger IND-CCA security
- Describe an IND-CCA secure symmetric encryption scheme

Recap: IND-CPA Security

Recap: IND-CPA Security

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

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

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

(m_0, m_1)

ct_b

b'

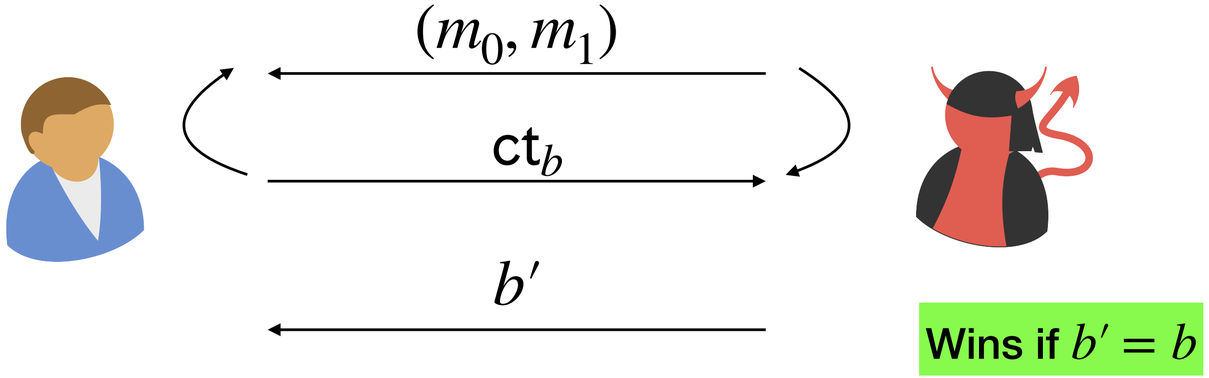
Wins if $b' = b$

Recap: IND-CPA Security

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

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

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



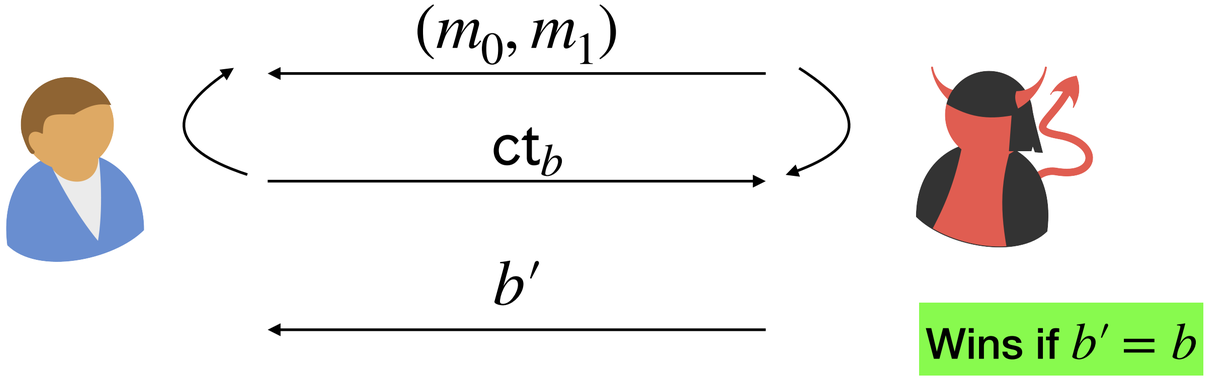
- Games capture the *capabilities* of the adversary

Recap: IND-CPA Security

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

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

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



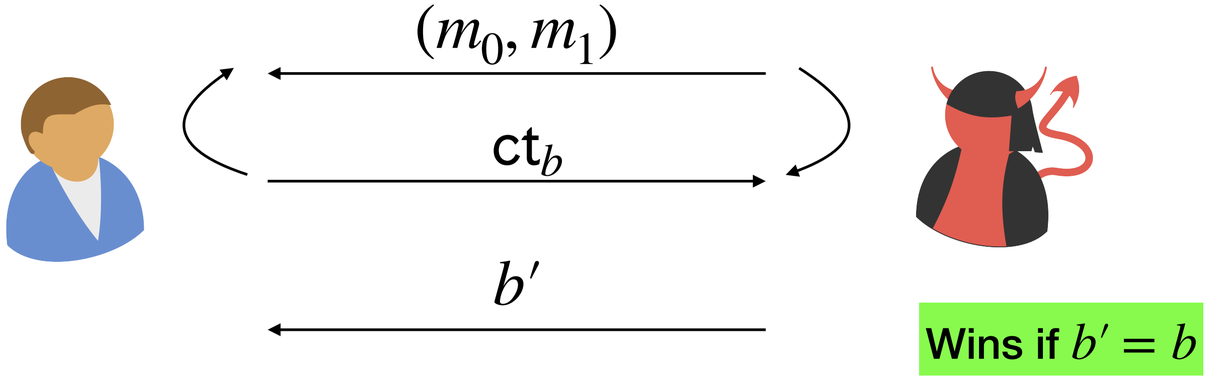
- Games capture the *capabilities* of the adversary
- In this game the adversary is *weak*. It can't learn anything about ciphertexts!

Recap: IND-CPA Security

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

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

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



- Games capture the *capabilities* of the adversary
- In this game the adversary is *weak*. It can't learn anything about ciphertexts!
- This may not be true in real life.

Padding Oracle Attack

Padding Oracle Attack

- Say our encryption scheme only works on 256-bit messages.

Padding Oracle Attack

- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.

Padding Oracle Attack

- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.

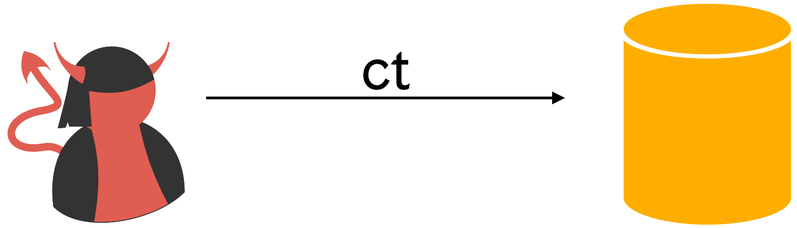
Padding Oracle Attack

- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.

Padding Oracle Attack

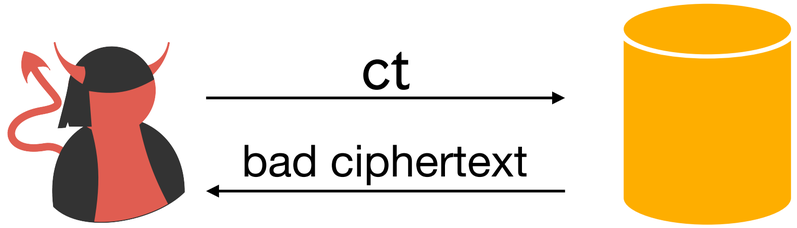
- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.

Padding Oracle Attack



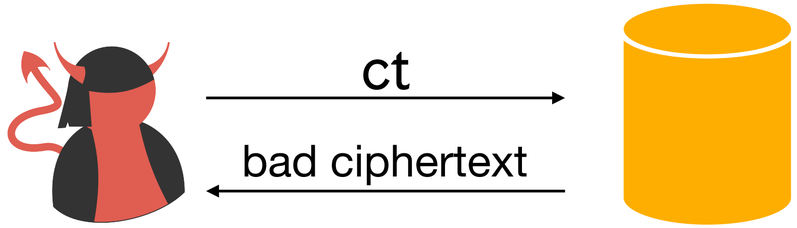
- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.

Padding Oracle Attack



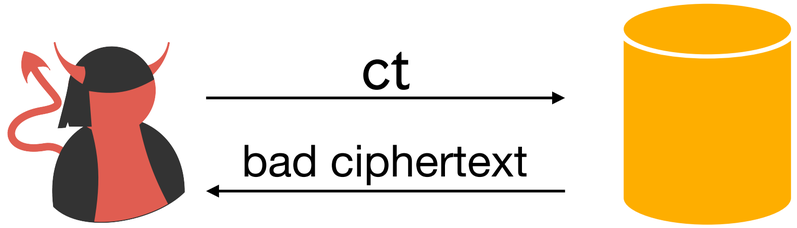
- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.

Padding Oracle Attack



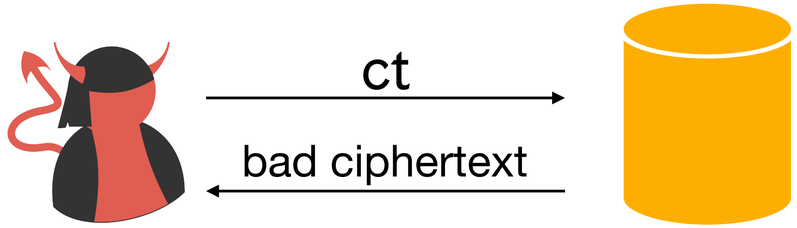
- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.
- This leaks information about ciphertexts!

Padding Oracle Attack



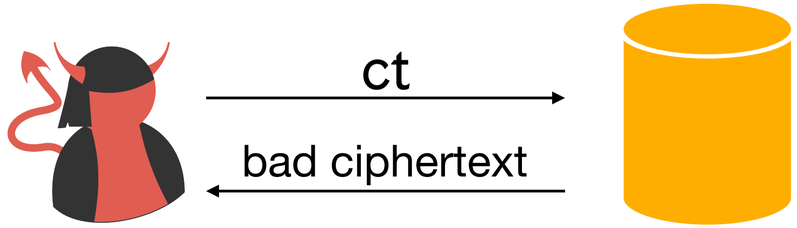
- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.
- This leaks information about ciphertexts!
  - Adversaries can learn if ciphertexts are properly padded, even if they don't know what message they encrypt.

Padding Oracle Attack



- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.
- This leaks information about ciphertexts!
  - Adversaries can learn if ciphertexts are properly padded, even if they don't know what message they encrypt.
  - This has been used to completely break schemes in real life!

Padding Oracle Attack



- Say our encryption scheme only works on 256-bit messages.
- To encrypt, we *pad* the message, and then encrypt, e.g. $m||030303$.
- To decrypt, we decrypt the ciphertext, then check the padding. If it is properly formatted, we accept, and otherwise we reject.
- This leaks information about ciphertexts!
  - Adversaries can learn if ciphertexts are properly padded, even if they don't know what message they encrypt.
  - This has been used to completely break schemes in real life!
  - The problem was these schemes were proved to be secure *only* if the adversary can't learn anything about the ciphertexts.

Updating IND-CPA

Updating IND-CPA

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

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

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

(m_0, m_1)

ct_b

b'

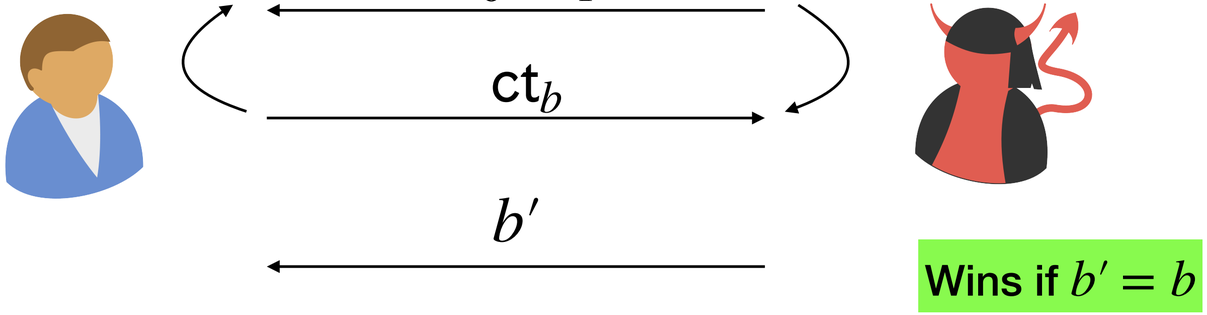
Wins if $b' = b$

Updating IND-CPA

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

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

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



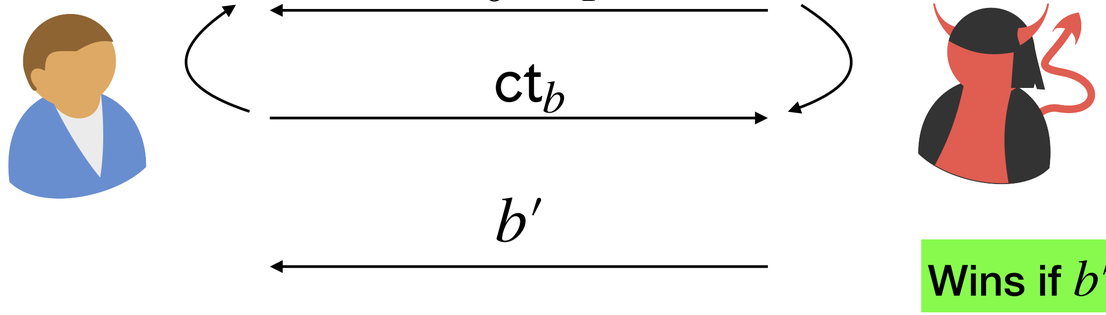
- We want to change the IND-CPA game so that adversaries can learn some things about ciphertexts. What should the adversary be able to learn?

Updating IND-CPA

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

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

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



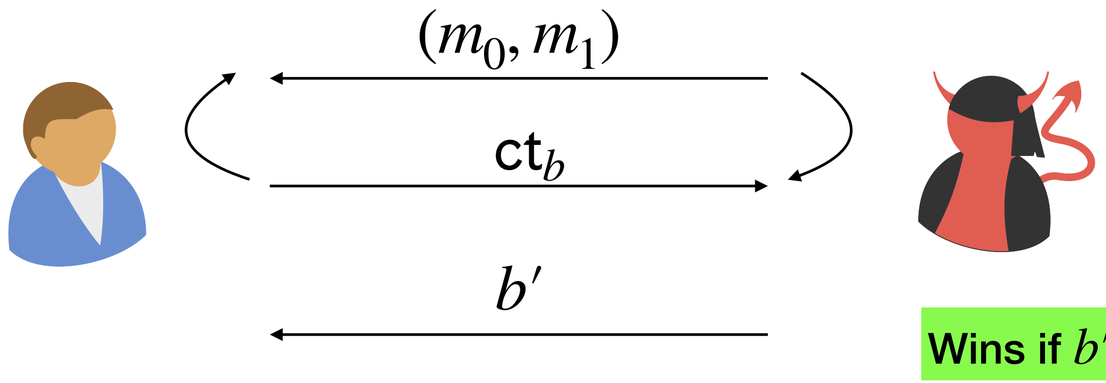
- We want to change the IND-CPA game so that adversaries can learn some things about ciphertexts. What should the adversary be able to learn?
- The *stronger* the adversary is, i.e. the more it can learn, the more secure our scheme is when we prove that it satisfies the definition.

Updating IND-CPA

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

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

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



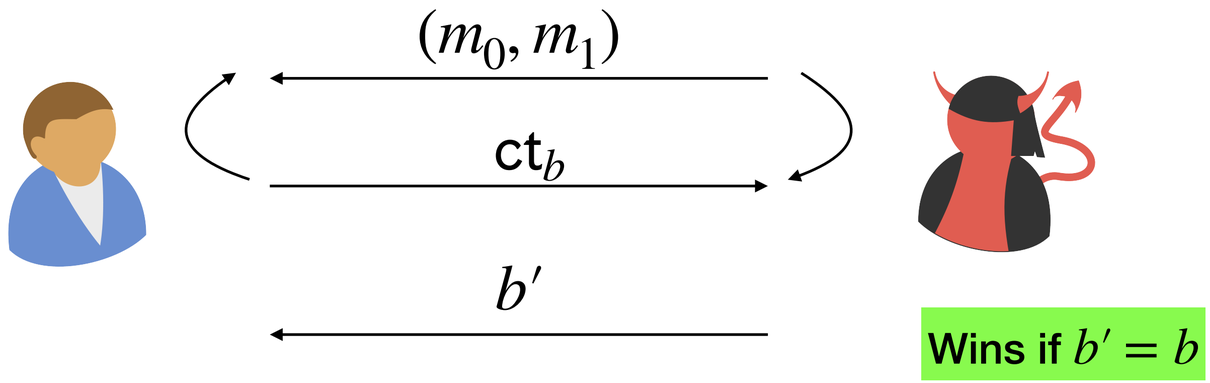
- We want to change the IND-CPA game so that adversaries can learn some things about ciphertexts. What should the adversary be able to learn?
- The *stronger* the adversary is, i.e. the more it can learn, the more secure our scheme is when we prove that it satisfies the definition.
- What is the most information an adversary can learn about a ciphertext?

Updating IND-CPA

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

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

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



- We want to change the IND-CPA game so that adversaries can learn some things about ciphertexts. What should the adversary be able to learn?
- The *stronger* the adversary is, i.e. the more it can learn, the more secure our scheme is when we prove that it satisfies the definition.
- What is the most information an adversary can learn about a ciphertext?
- The plaintext it encrypts!

IND-CCA-1 Security

IND-CCA-1 Security

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

ct ←

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

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

ct ←

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

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

ct

m

IND-CCA-1 Security

IND-CCA-1 Security

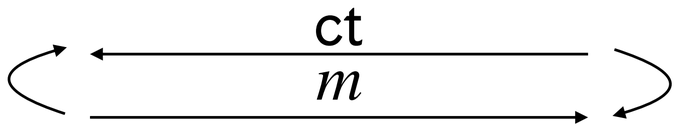
An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack (1)* (IND-CCA-1) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

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

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

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

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



IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

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

ct

m

(m_0, m_1)

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

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

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

ct

m

(m_0, m_1)

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

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

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

ct

m

(m_0, m_1)

ct_b

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

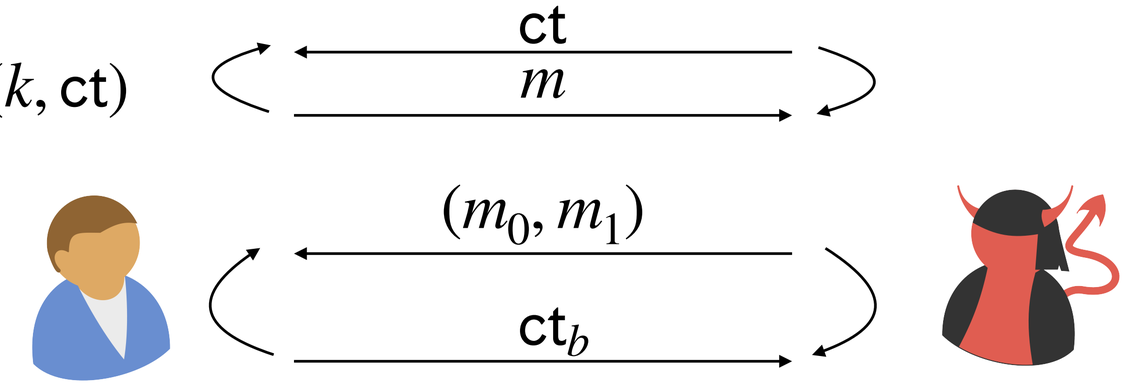
$$\Pr[\mathscr{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

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

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



IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

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

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

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

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

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

ct

m

(m_0, m_1)

ct_b

b'

IND-CCA-1 Security

IND-CCA-1 Security

An 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 $\forall \lambda \in \mathbb{N}$

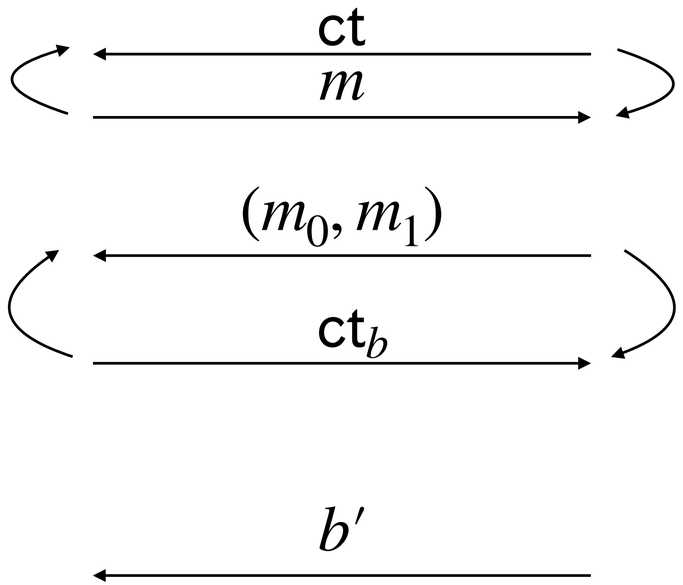
$$\Pr[\mathscr{A} \text{ wins CCA1Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

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

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



Wins if $b' = b$

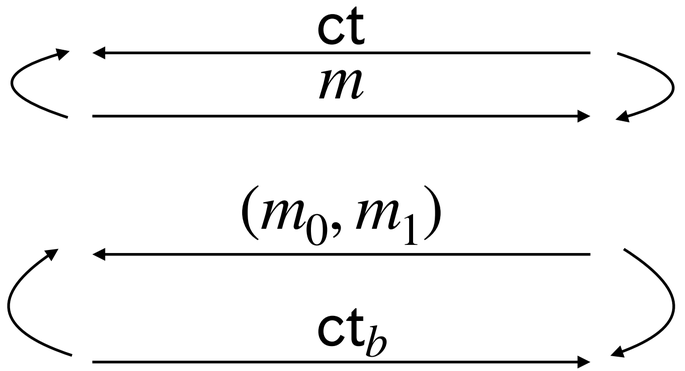
IND-CCA-1 Security

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

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

$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

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



Wins if $b' = b$

b'

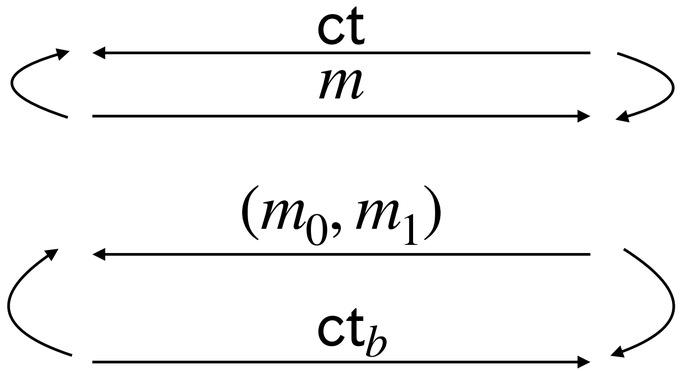
IND-CCA-1 Security

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

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

$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

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



Wins if $b' = b$

b'

- This definition seems arbitrary.

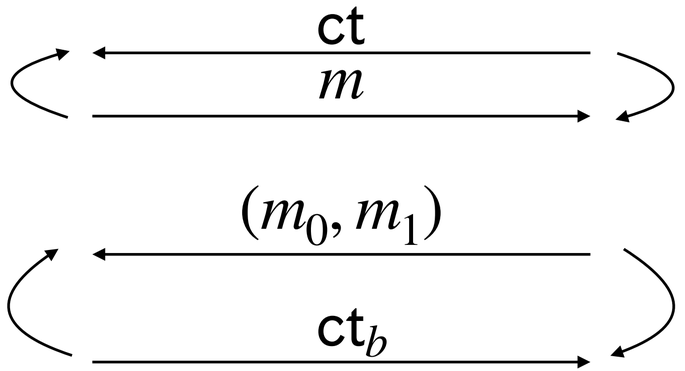
IND-CCA-1 Security

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

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

$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

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



Wins if $b' = b$

- This definition seems arbitrary.
- Why can the adversary only get decryptions before seeing the challenge ciphertext?

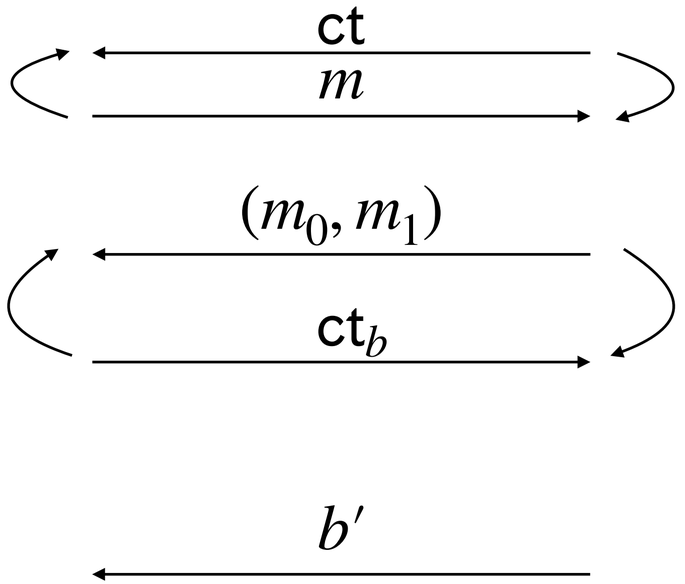
IND-CCA-1 Security

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

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

$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

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



Wins if $b' = b$

- This definition seems arbitrary.
- Why can the adversary only get decryptions before seeing the challenge ciphertext?
- For *real* security, we should let the adversary get decryptions *after* seeing the challenge also.

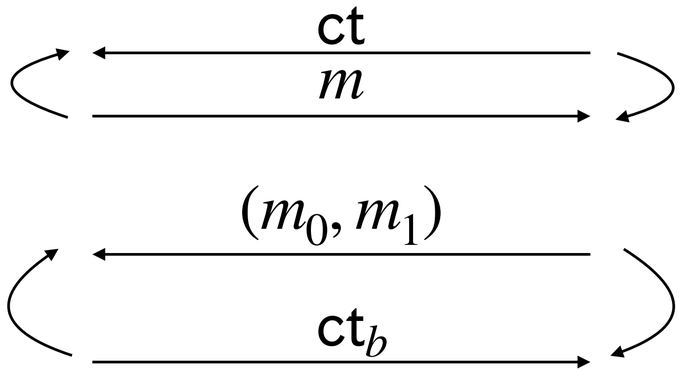
IND-CCA-1 Security

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

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

$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

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



Wins if $b' = b$

- This definition seems arbitrary.
- Why can the adversary only get decryptions before seeing the challenge ciphertext?
- For *real* security, we should let the adversary get decryptions *after* seeing the challenge also.
- But also need to make sure they don't just decrypt $\mathbf{ct}_b$!

IND-CCA-2 Security

IND-CCA-2 Security

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

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

IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

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

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

IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

IND-CCA-2 Security

IND-CCA-2 Security

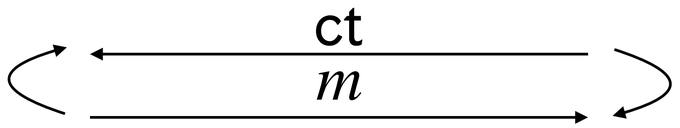
An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

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



IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

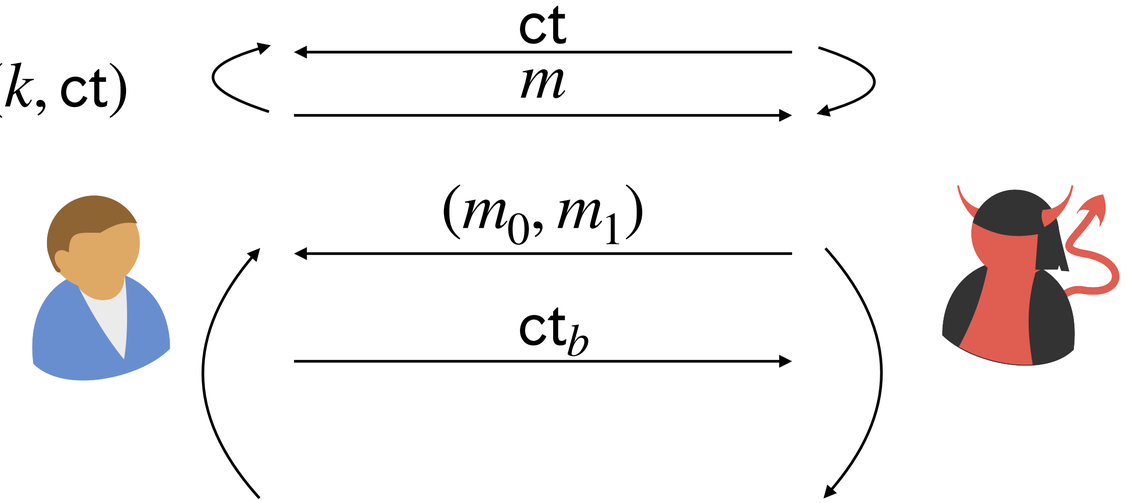
$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

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

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



IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

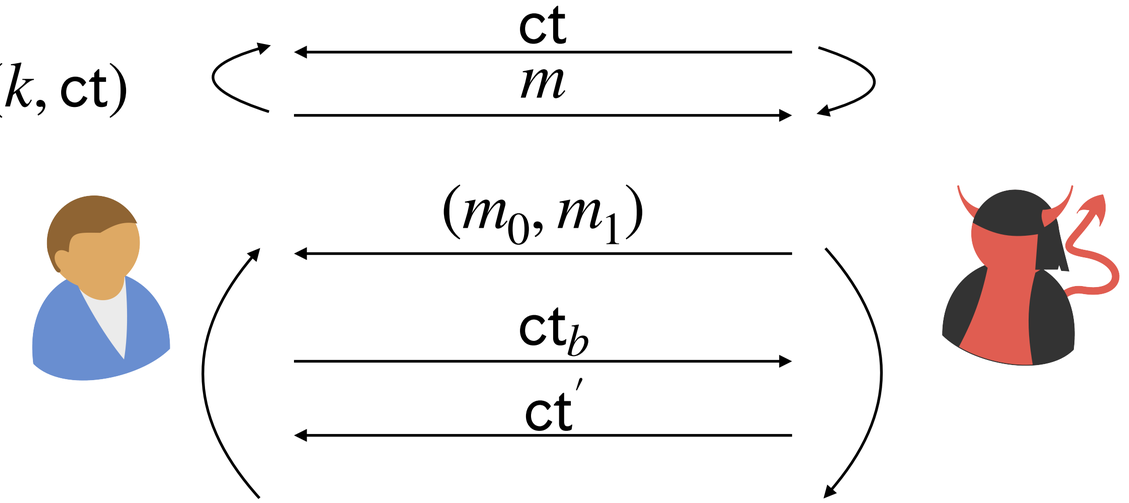
$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

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

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

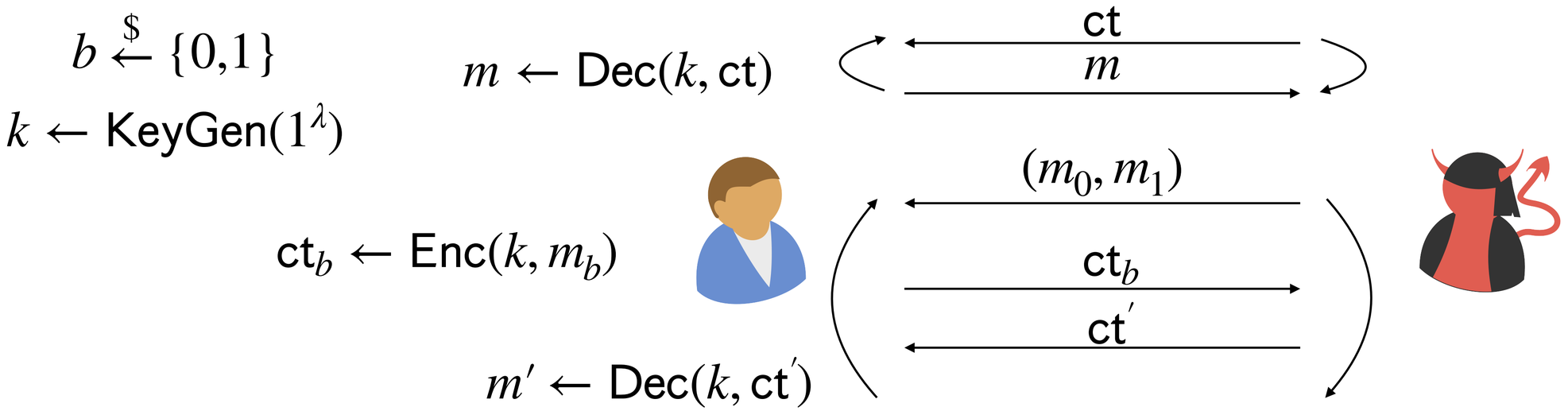


IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

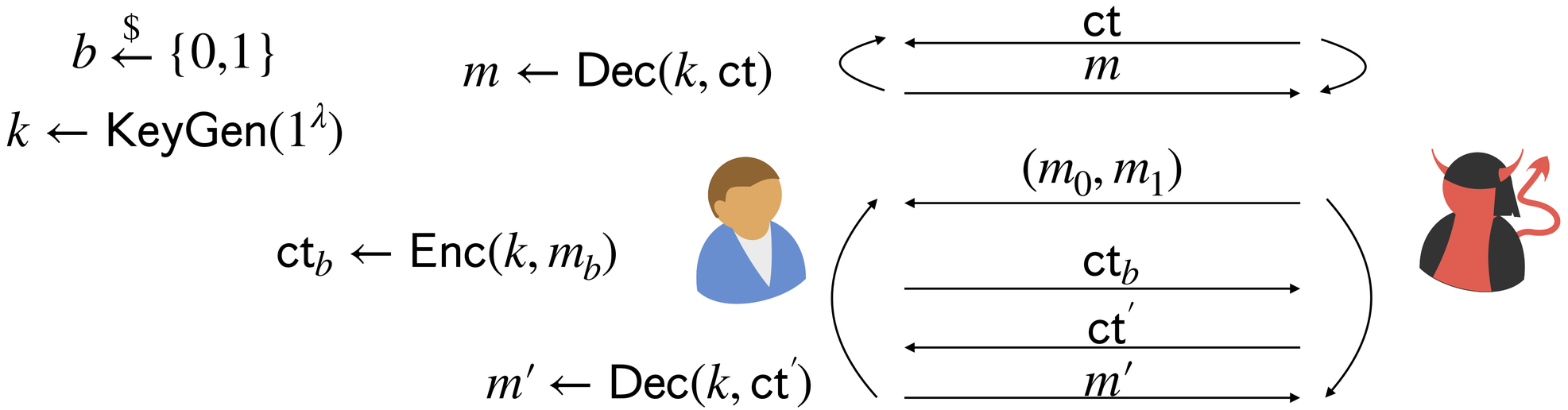


IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

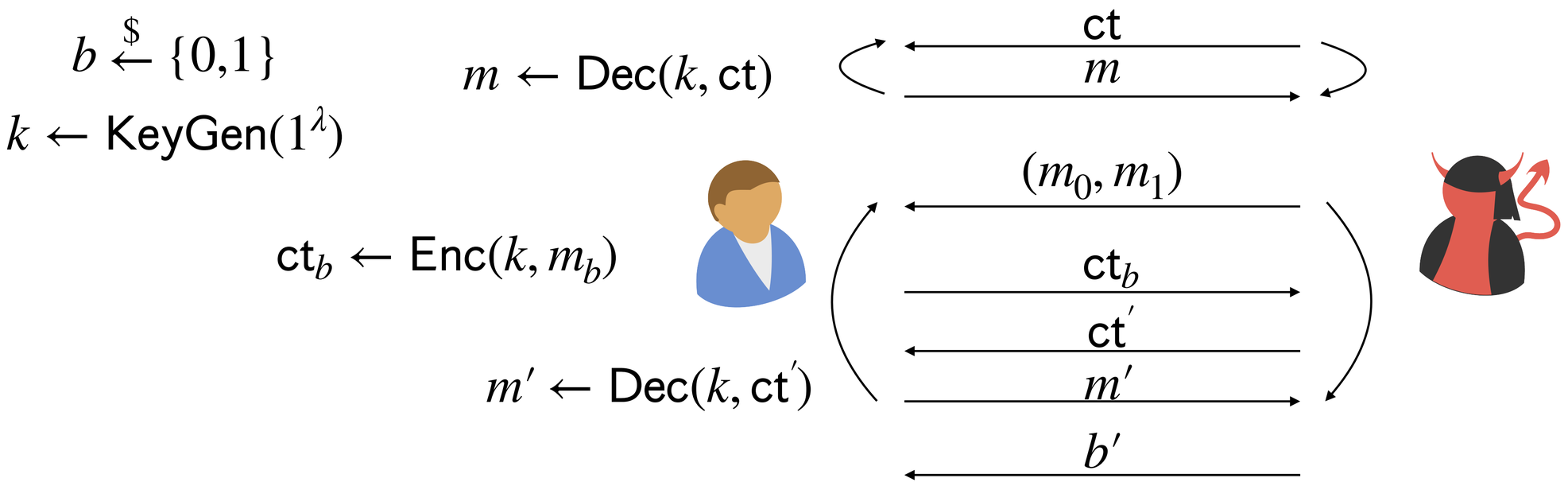


IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

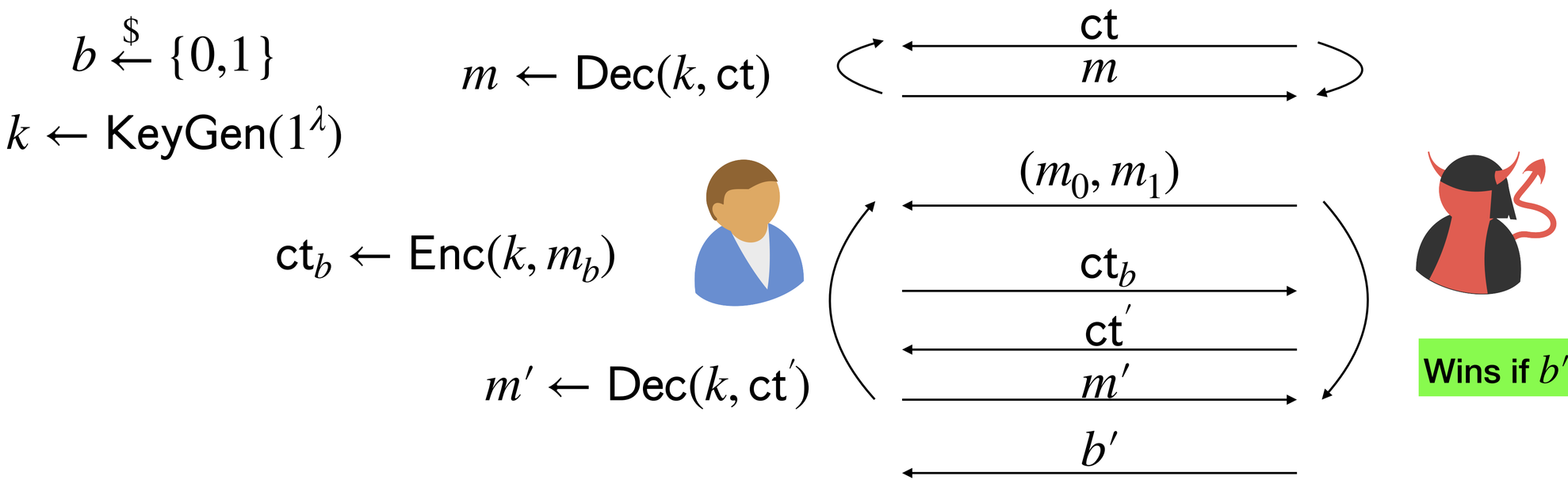


IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$



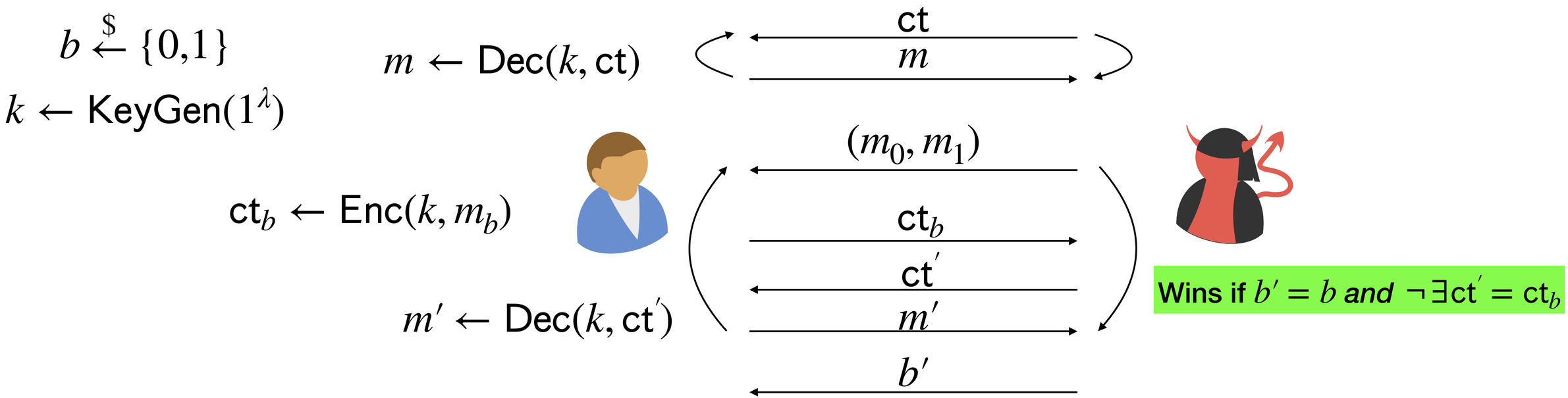
Wins if $b' = b$

IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

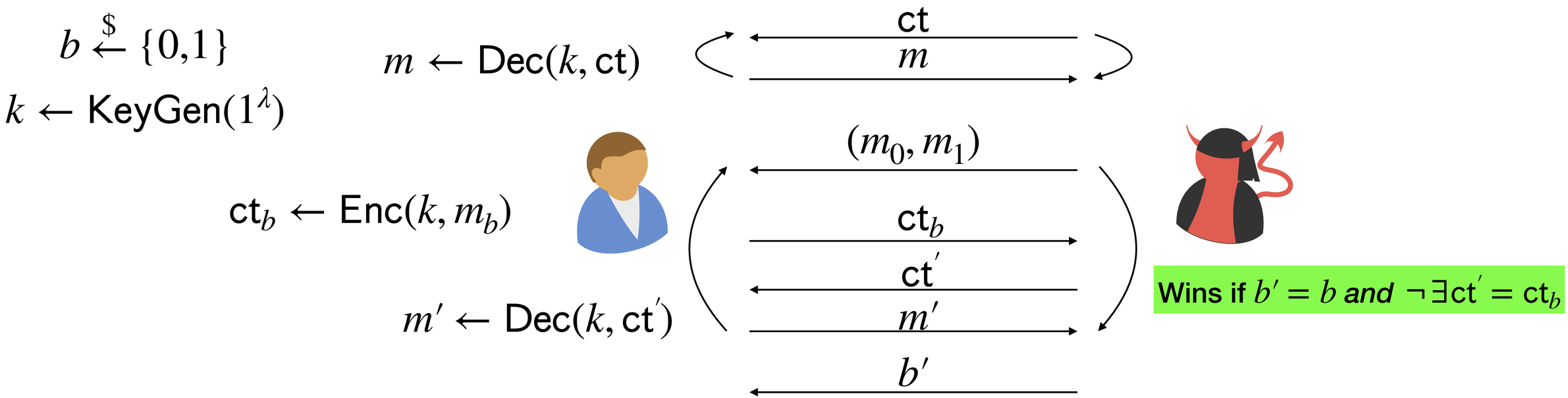


IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

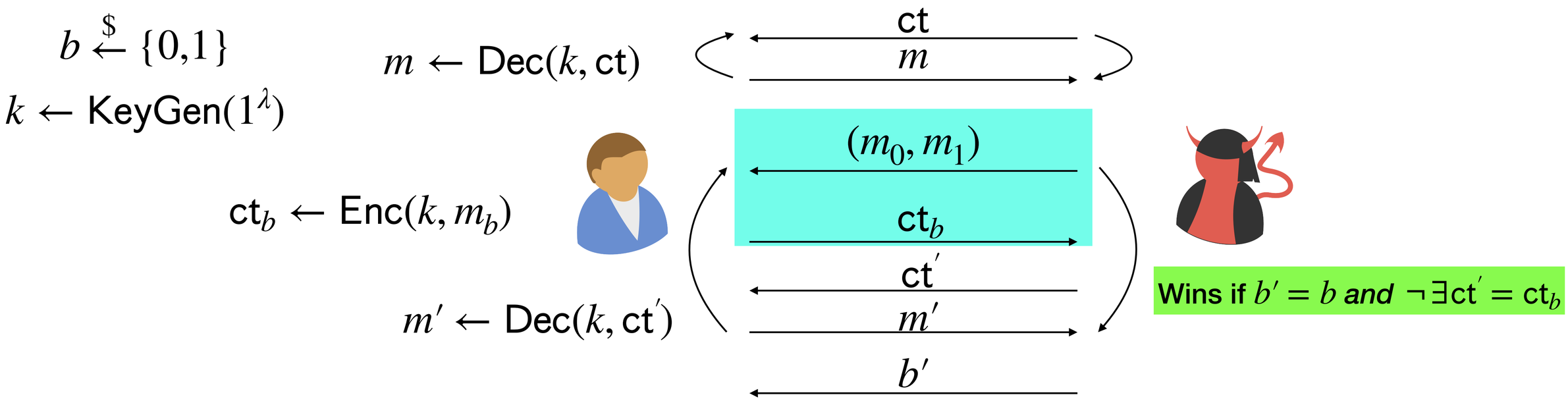


IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathscr{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathscr{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$



IND-CCA-2 Security

Encryption Oracle
$O_E^b(k, \cdot, \cdot)$

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

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

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

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

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

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

ct
m
(m_0, m_1)
ct_b
ct'
m'
b'

Wins if $b' = b$ *and* $\neg \exists \text{ct}' = \text{ct}_b$

IND-CCA-2 Security

Encryption Oracle
$O_E^b(k, \cdot, \cdot)$

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$

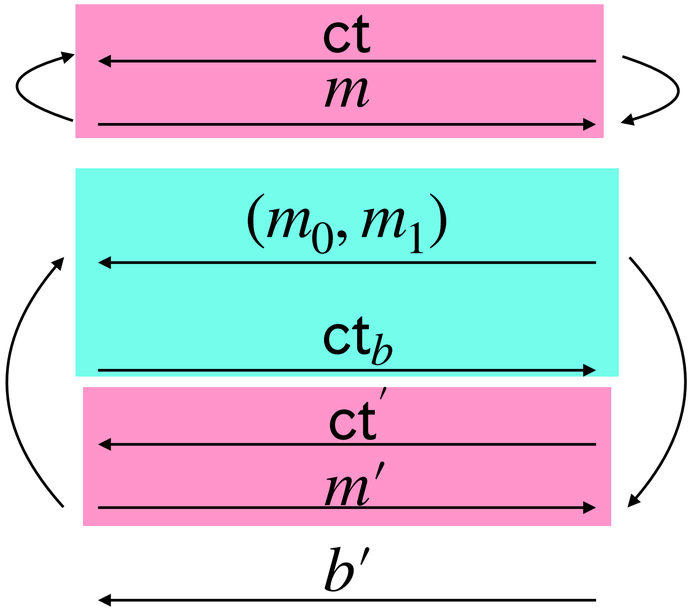
$b \xleftarrow{\$} \{0,1\}$

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

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

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

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



Wins if $b' = b$ *and* $\neg \exists \text{ct}' = \text{ct}_b$

IND-CCA-2 Security

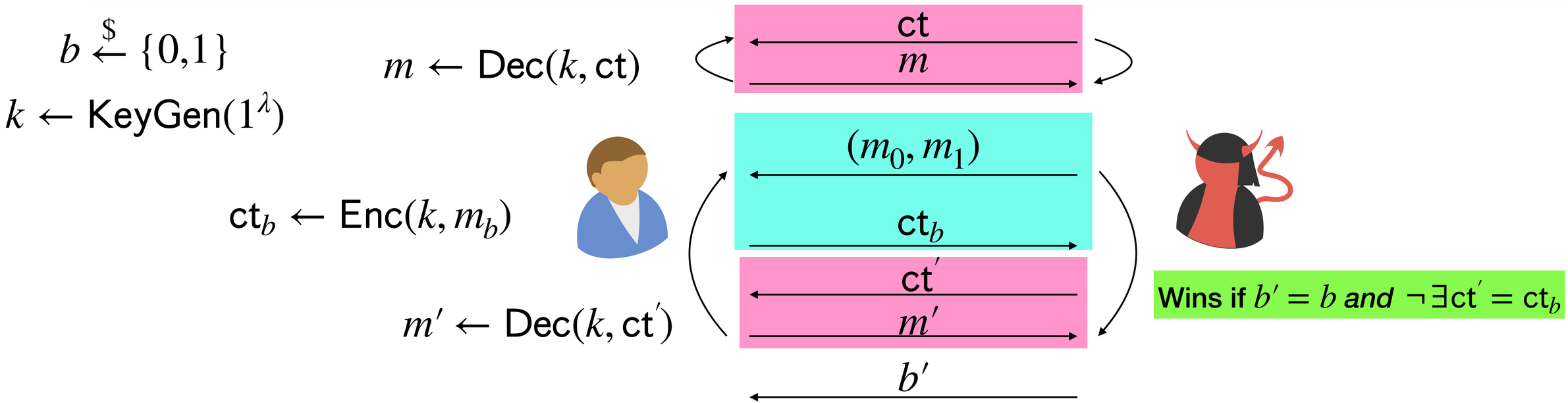
Encryption Oracle $O_E^b(k, \cdot, \cdot)$

Decryption Oracle $O_D(k)$

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr[\mathcal{A} \text{ wins CCA2Game}] \leq \frac{1}{2} + \nu(\lambda)$$



IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

IND-CCA-2 Security

IND-CCA-2 Security

An encryption scheme (KeyGen, Enc, Dec) satisfies *indistinguishability under chosen-ciphertext attack* (IND-CCA-2) if for all NUPPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that $\forall \lambda \in \mathbb{N}$

$$\Pr\left[\begin{array}{c} b' = b \wedge \\ \mathcal{A} \text{ never queries an output of } O_E^b \text{ to } O_D \end{array} : \begin{array}{c} k \leftarrow \text{KeyGen}(1^\lambda) \\ b \xleftarrow{\$} \{0,1\} \\ b' \leftarrow \mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda) \end{array}\right] \leq \frac{1}{2} + \nu(\lambda)$$

Challenges to Achieving CCA-2

$$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$$

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathcal{A}$ to query the decryption oracle on a new ciphertext, but still win the game!

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathcal{A}$ to query the decryption oracle on a new ciphertext, but still win the game!
- In the reduction: We will want to use an IND-CPA scheme as a building block to make a CCA-2 scheme

Challenges to Achieving CCA-2

$\mathscr{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathscr{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathscr{A}$ to query the decryption oracle on a new ciphertext, but still win the game!
- In the reduction: We will want to use an IND-CPA scheme as a building block to make a CCA-2 scheme
  - That means we have to reduce to the CPA security of the underlying scheme

Challenges to Achieving CCA-2

$$\mathscr{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$$

- Practically: $\mathscr{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathscr{A}$ to query the decryption oracle on a new ciphertext, but still win the game!
- In the reduction: We will want to use an IND-CPA scheme as a building block to make a CCA-2 scheme
  - That means we have to reduce to the CPA security of the underlying scheme
  - When our internal adversary makes decryption queries how can we answer them? We are playing the CPA game and so *don't have* a decryption oracle.

Challenges to Achieving CCA-2

$$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$$

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathcal{A}$ to query the decryption oracle on a new ciphertext, but still win the game!

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

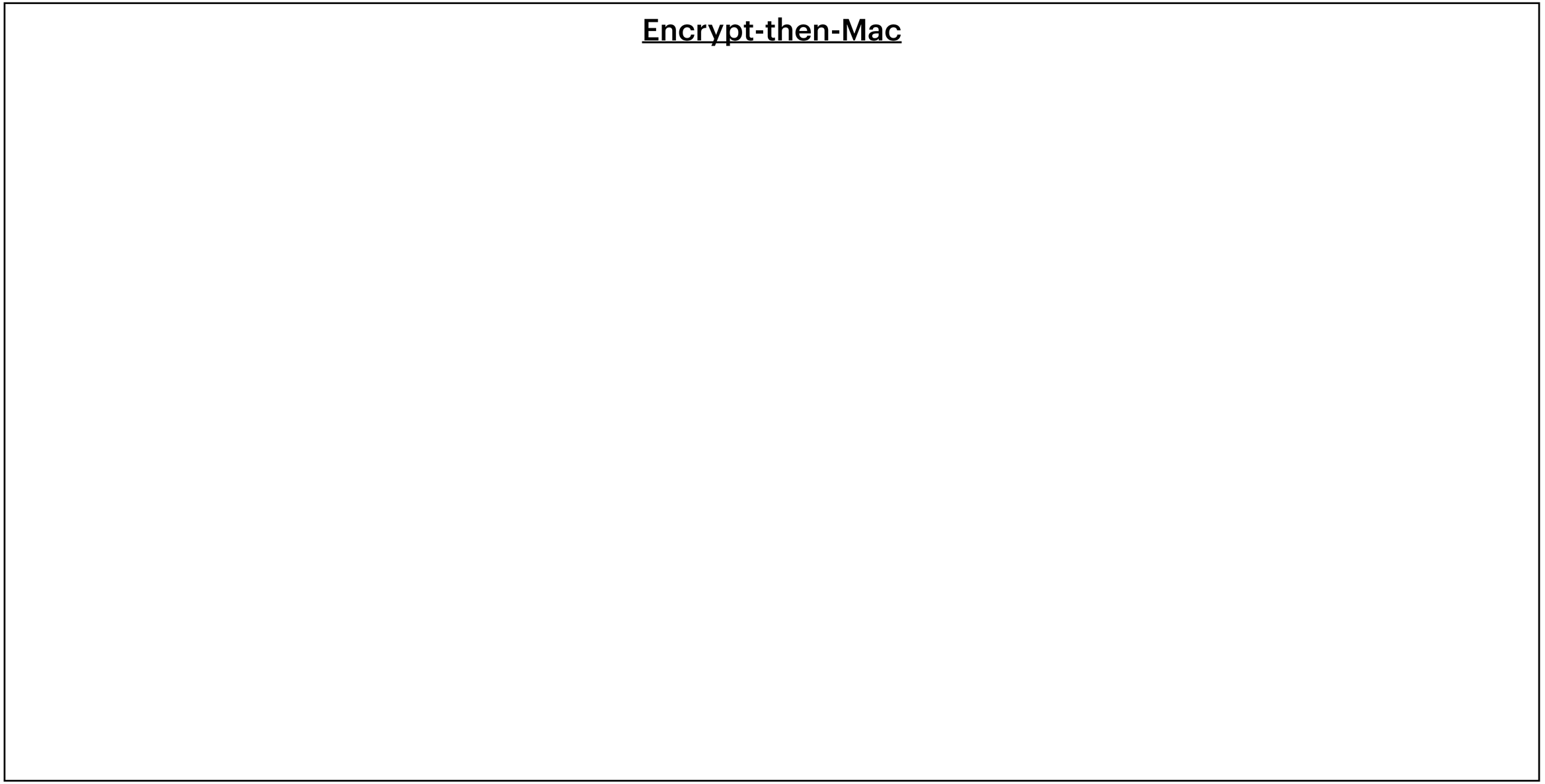
- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathcal{A}$ to query the decryption oracle on a new ciphertext, but still win the game!
- How do we fix this? Prevent $\mathcal{A}$ from be able to make new *valid* ciphertexts.

Challenges to Achieving CCA-2

$\mathcal{A}^{O_E^b(k,\cdot,\cdot),O_D(k,\cdot)}(1^\lambda)$

- Practically: $\mathcal{A}$ can *transform* ciphertexts into *related ciphertexts.*
  - For example: given a ciphertext $\mathsf{ct} = \mathsf{Enc}(m)$, can create $\mathsf{ct}' = \mathsf{Enc}(m+1)$
  - This would allow $\mathcal{A}$ to query the decryption oracle on a new ciphertext, but still win the game!
- How do we fix this? Prevent $\mathcal{A}$ from be able to make new *valid* ciphertexts.
- If **Dec** just rejects ciphertexts that weren't output by **Enc**, the decryption oracle is useless!

Encrypt-then-Mac



Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure encryption scheme, and let $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ be a *strong* UF-CMA secure MAC scheme.

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure encryption scheme, and let $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ be a *strong* UF-CMA secure MAC scheme.

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

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure encryption scheme, and let $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure encryption scheme, and let $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Ver})$ be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- KGen$'(1^\lambda)$:
  - $k_{MAC} \leftarrow$ KeyGen(1^λ)
  - $k_{Enc} \leftarrow$ KGen(1^λ)
  - return (k_{MAC}, k_{Enc})

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- KGen$'(1^\lambda)$:
  - $k_{MAC} \leftarrow$ KeyGen(1^λ)
  - $k_{Enc} \leftarrow$ KGen(1^λ)
  - return (k_{MAC}, k_{Enc})
- Enc$'((k_{MAC}, k_{Enc}), m)$:

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$
  - return (k_{MAC}, k_{Enc})
- $\mathsf{Enc}'((k_{MAC}, k_{Enc}), m)$:
  - $\mathsf{ct} \leftarrow \mathsf{Enc}(k_{Enc}, m)$

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$
  - return (k_{MAC}, k_{Enc})
- $\mathsf{Enc}'((k_{MAC}, k_{Enc}), m)$:
  - $\mathsf{ct} \leftarrow \mathsf{Enc}(k_{Enc}, m)$
  - return $(\mathsf{ct}, \mathsf{Tag}(k_{MAC}, \mathsf{ct}))$

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$
  - return (k_{MAC}, k_{Enc})
- $\mathsf{Enc}'((k_{MAC}, k_{Enc}), m)$:
  - $\mathsf{ct} \leftarrow \mathsf{Enc}(k_{Enc}, m)$
  - return $(\mathsf{ct}, \mathsf{Tag}(k_{MAC}, \mathsf{ct}))$

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$
  - return (k_{MAC}, k_{Enc})
- $\mathsf{Enc}'((k_{MAC}, k_{Enc}), m)$:
  - $\mathsf{ct} \leftarrow \mathsf{Enc}(k_{Enc}, m)$
  - return $(\mathsf{ct}, \mathsf{Tag}(k_{MAC}, \mathsf{ct}))$
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$
  - return (k_{MAC}, k_{Enc})
- $\mathsf{Enc}'((k_{MAC}, k_{Enc}), m)$:
  - $\mathsf{ct} \leftarrow \mathsf{Enc}(k_{Enc}, m)$
  - return $(\mathsf{ct}, \mathsf{Tag}(k_{MAC}, \mathsf{ct}))$
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$

Encrypt-then-Mac

Encrypt-then-Mac

Let λ be the security parameter, $\ell(\lambda)$ be a polynomial , (KGen, Enc, Dec) be an IND-CPA secure encryption scheme, and let (KeyGen, Tag, Ver) be a *strong* UF-CMA secure MAC scheme.

- $\mathsf{KGen}'(1^\lambda)$:
  - $k_{MAC} \leftarrow \mathsf{KeyGen}(1^\lambda)$
  - $k_{Enc} \leftarrow \mathsf{KGen}(1^\lambda)$
  - return (k_{MAC}, k_{Enc})
- $\mathsf{Enc}'((k_{MAC}, k_{Enc}), m)$:
  - $\mathsf{ct} \leftarrow \mathsf{Enc}(k_{Enc}, m)$
  - return $(\mathsf{ct}, \mathsf{Tag}(k_{MAC}, \mathsf{ct}))$
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

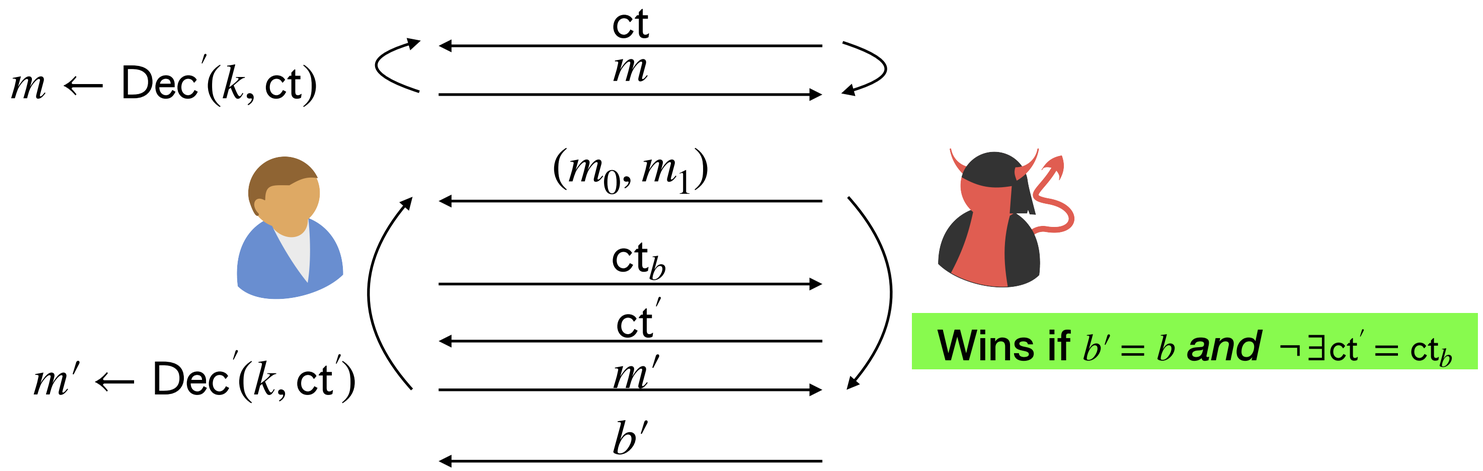
Proof of Security

Proof of Security

Game 0

Proof of Security

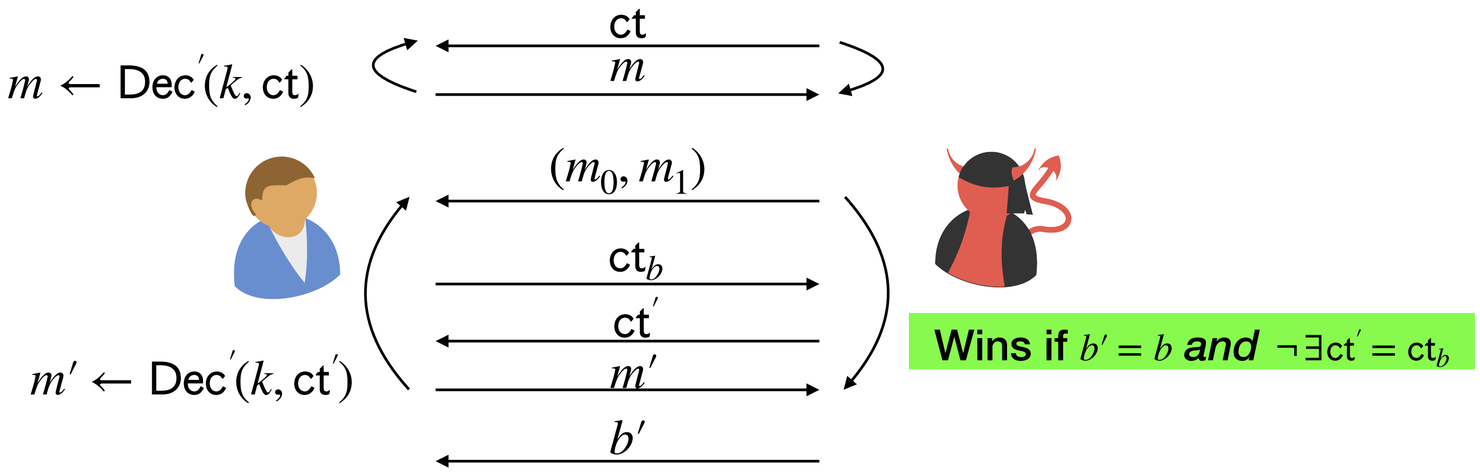
Game 0



Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

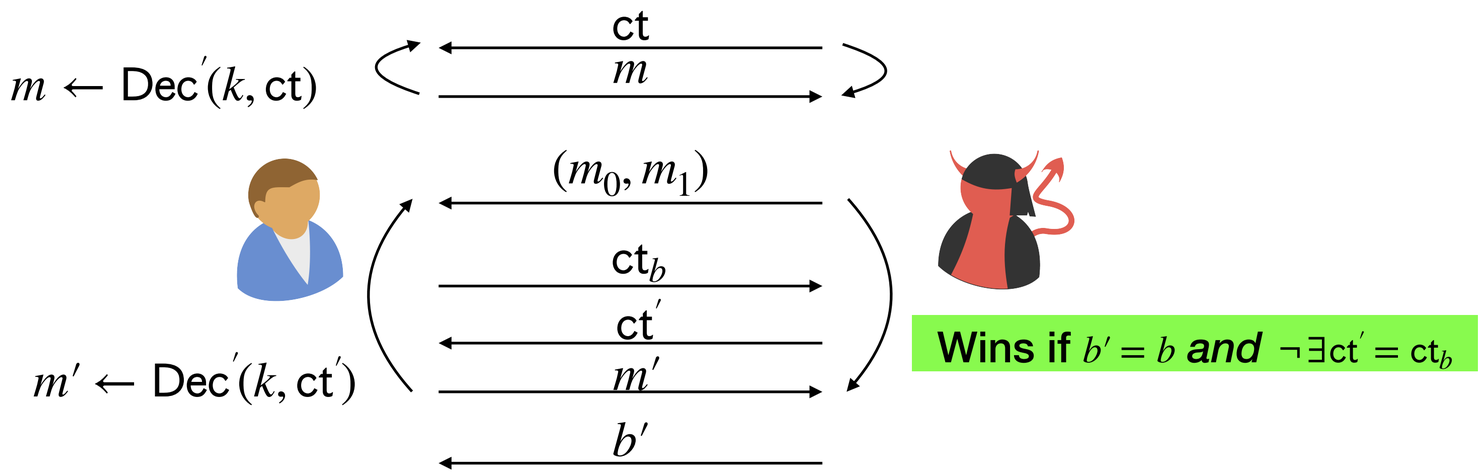
Game 0



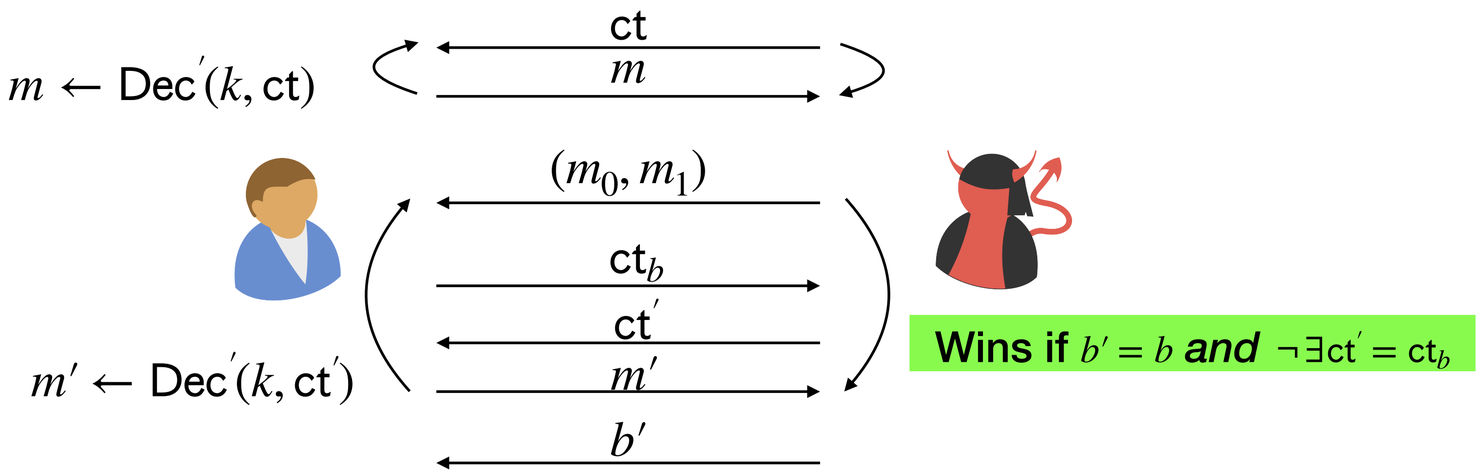
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

Game 0



Game 1

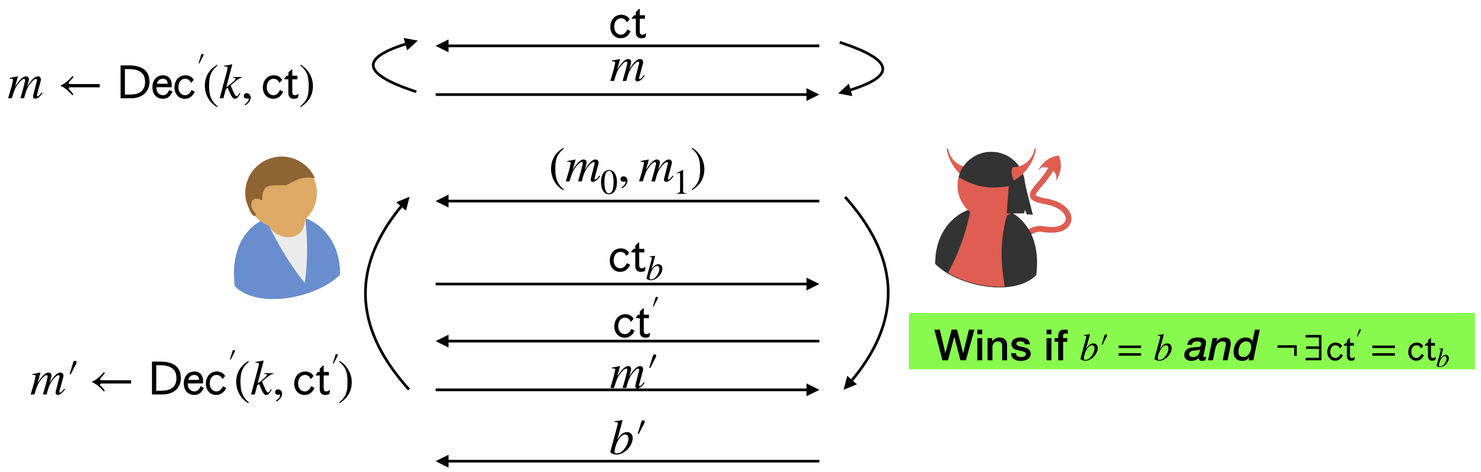


Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

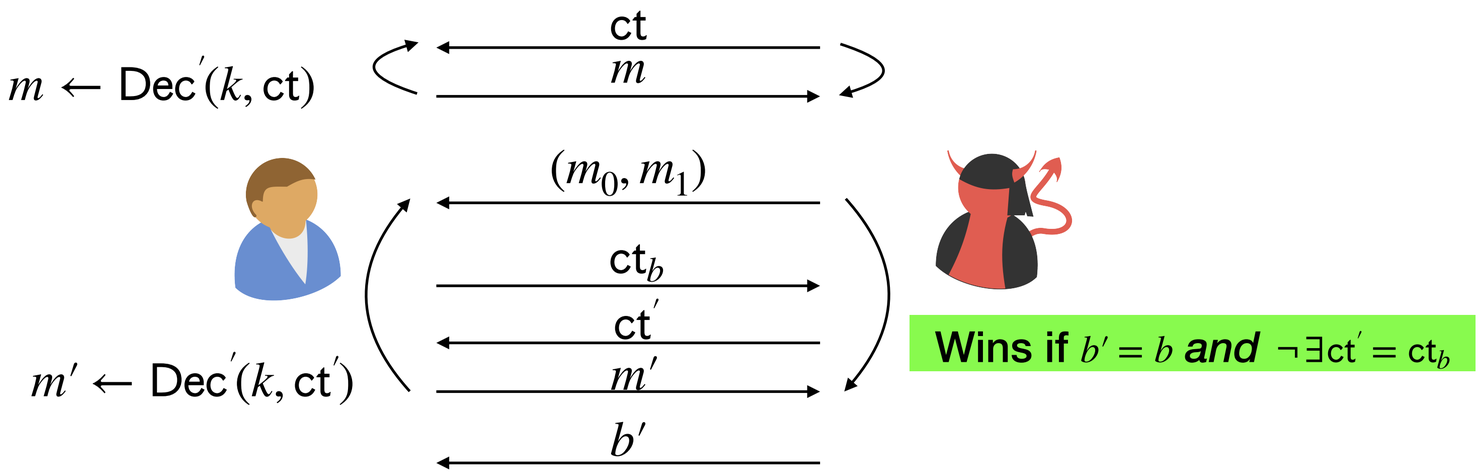
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Game 0



Wins if $b' = b$ *and* $\neg \exists \mathsf{ct}' = \mathsf{ct}_b$

Game 1



Wins if $b' = b$ *and* $\neg \exists \mathsf{ct}' = \mathsf{ct}_b$

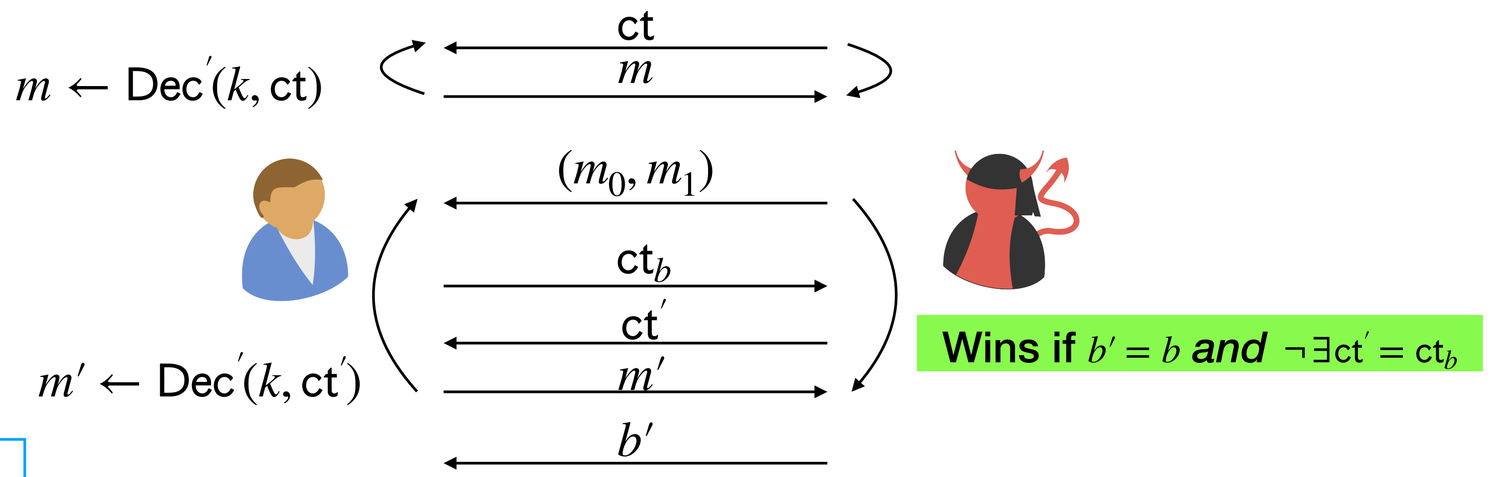
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

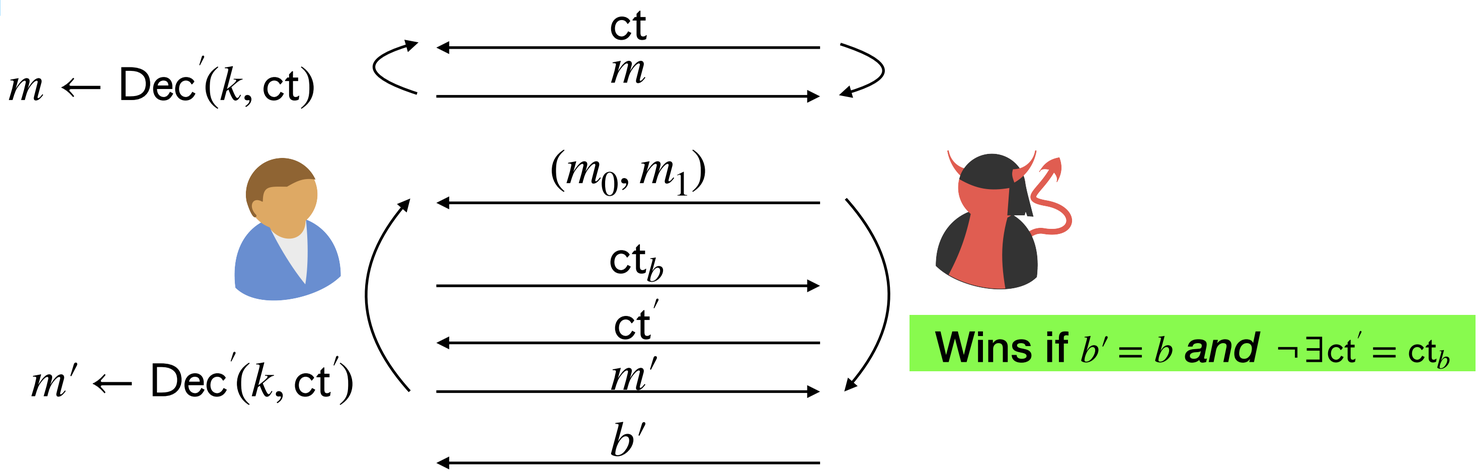
Claim:
$|\Pr[\mathcal{A} \text{ wins } \mathsf{Game}_0] - \Pr[\mathcal{A} \text{ wins } \mathsf{Game}_1]| \leq \mathsf{negl}(\lambda)$

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Game 0



Game 1



Proof of Security

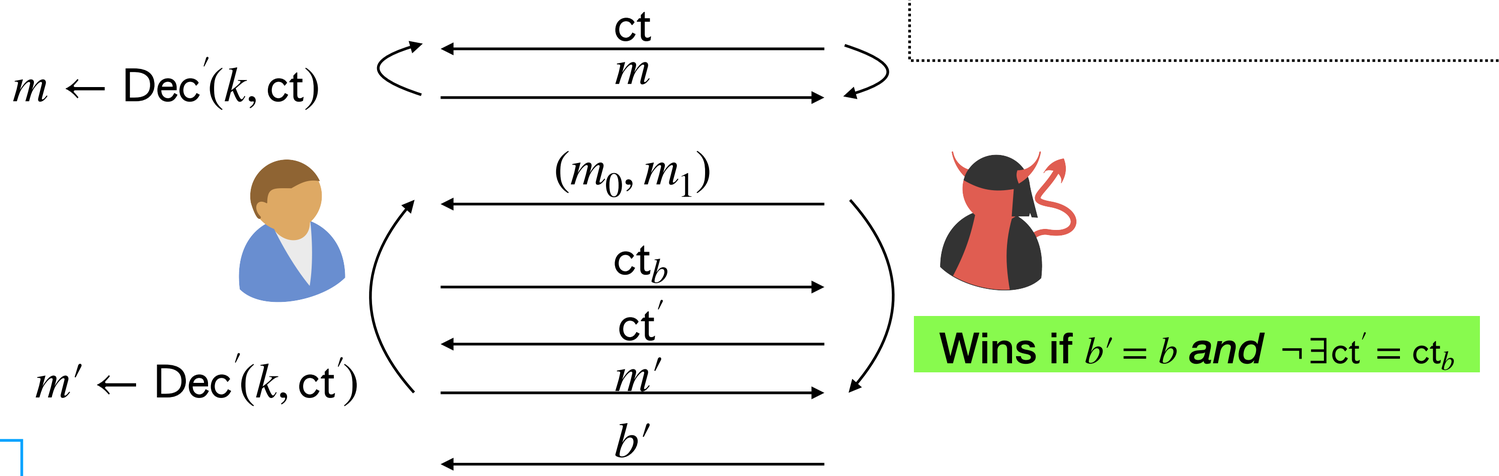
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

Claim:
$|\Pr[\mathscr{A} \text{ wins } \mathsf{Game}_0] - \Pr[\mathscr{A} \text{ wins } \mathsf{Game}_1]| \leq \mathsf{negl}(\lambda)$

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

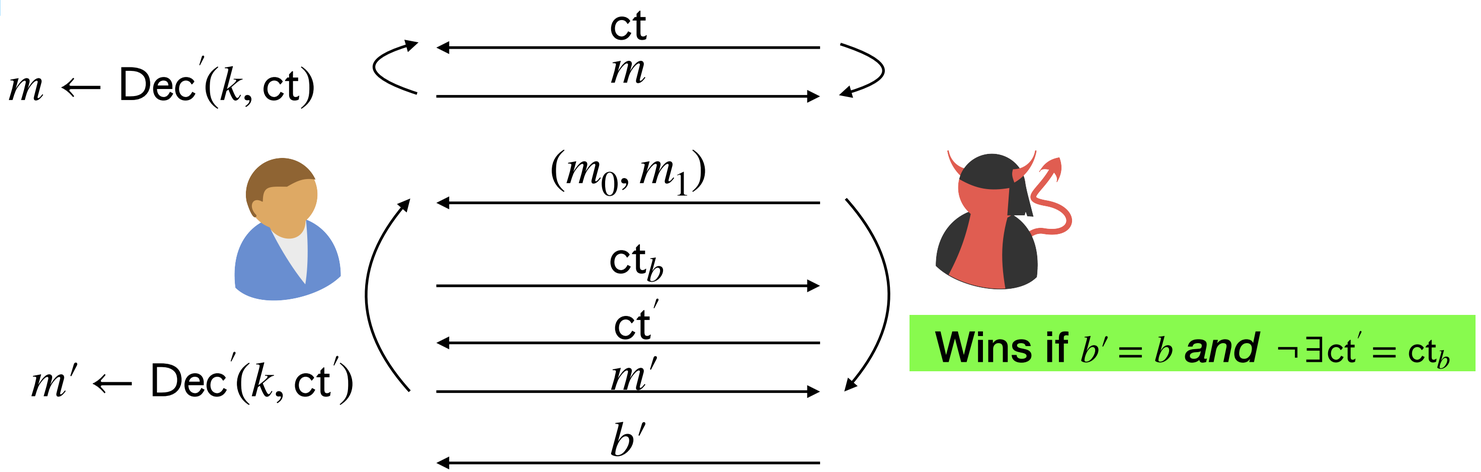
These games are only different to a winning adversary if it can create new signatures.

Game 0



Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

Game 1



Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

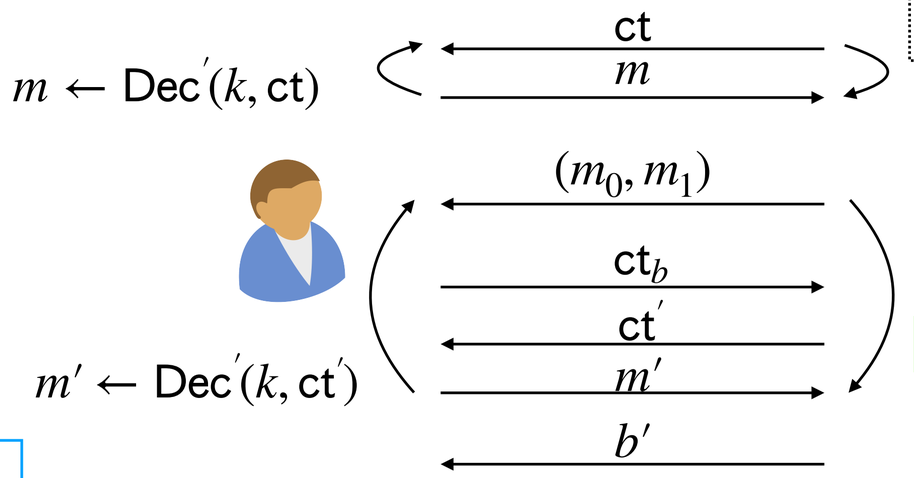
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

Claim:
$|\Pr[\mathscr{A} \text{ wins } \mathsf{Game}_0] - \Pr[\mathscr{A} \text{ wins } \mathsf{Game}_1]| \leq \mathsf{negl}(\lambda)$

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

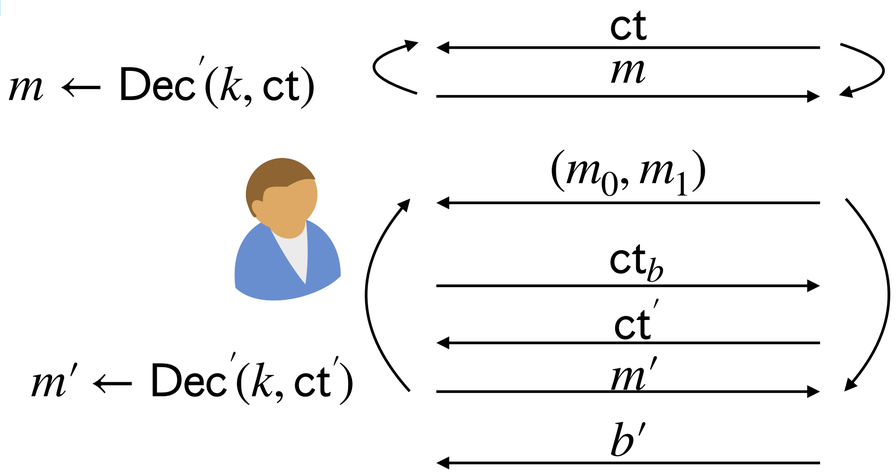
Game 0



But, by strong UF-CMA security it can't!

Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

Game 1



Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

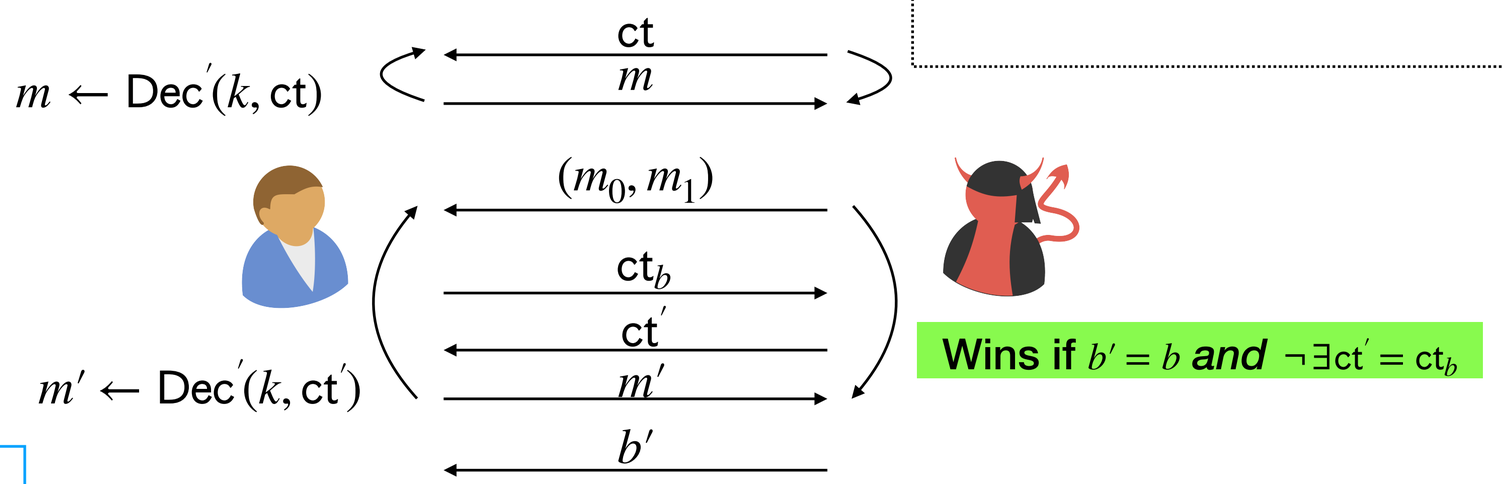
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - If $\mathsf{Ver}(k_{MAC}, \mathsf{ct}, \sigma) \neq 1$ then return $\perp$
  - Else return $\mathsf{Dec}(k_{Enc}, \mathsf{ct})$

Claim:
$|\Pr[\mathscr{A} \text{ wins } \mathsf{Game}_0] - \Pr[\mathscr{A} \text{ wins } \mathsf{Game}_1]| \leq \mathsf{negl}(\lambda)$

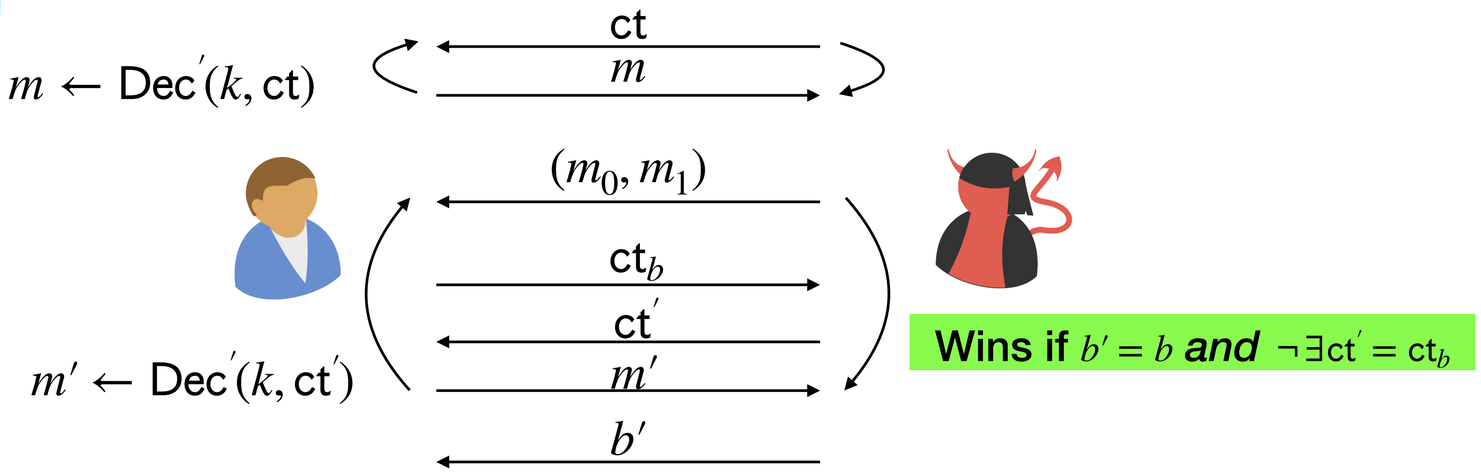
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Game 0



Wins if $b' = b$ ***and*** $\neg \exists \mathsf{ct}' = \mathsf{ct}_b$

Game 1



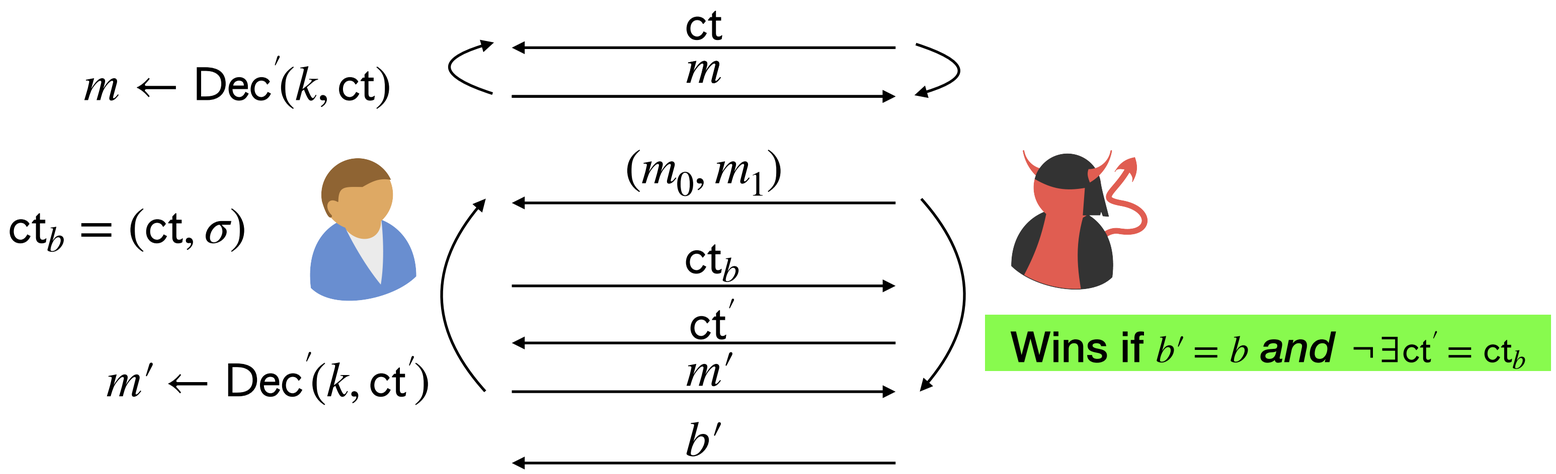
Wins if $b' = b$ ***and*** $\neg \exists \mathsf{ct}' = \mathsf{ct}_b$

So
$|\Pr[\mathscr{A} \text{ wins } \mathsf{Game}_0] - \Pr[\mathscr{A} \text{ wins } \mathsf{Game}_1]| \leq \mathsf{negl}(\lambda)$
By the strong UF-CMA security of MAC

Proof of Security

Proof of Security

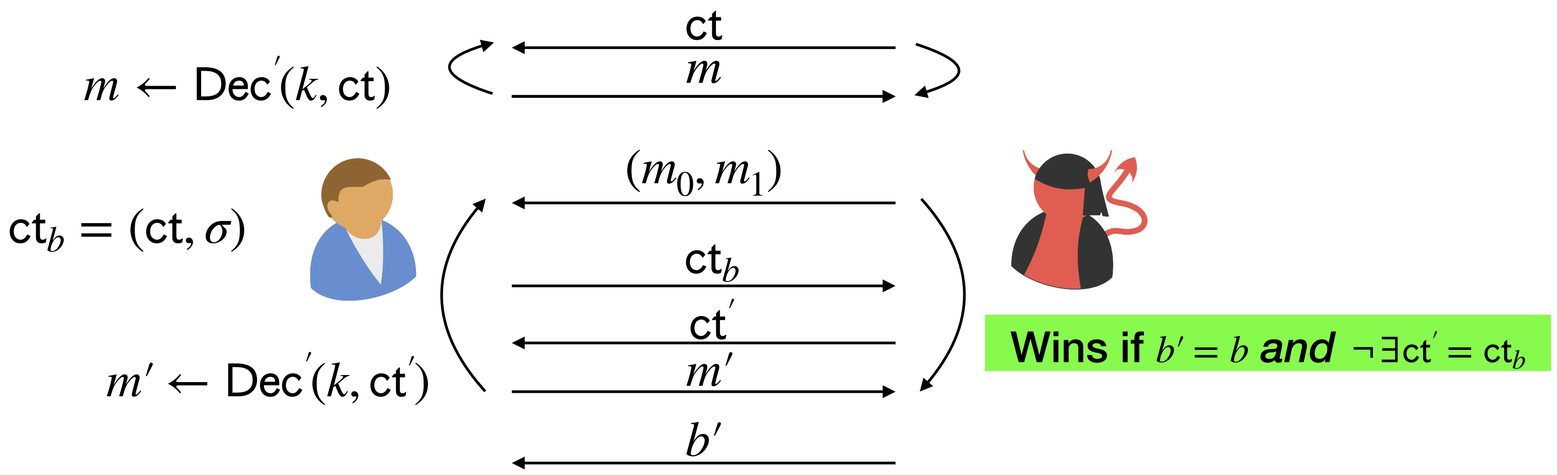
- $\mathrm{Dec}'((k_{MAC}, k_{Enc}), (\mathrm{ct}, \sigma))$:
  - return $\perp$



Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$



Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

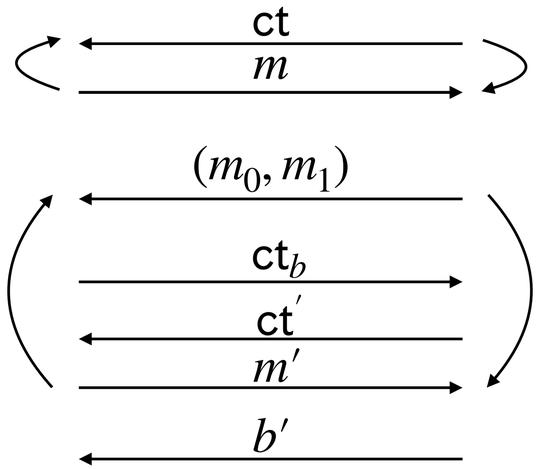
Ch_{CPA}

Game 1

$m \leftarrow \mathsf{Dec}'(k, \mathsf{ct})$

$\mathsf{ct}_b = (\mathsf{ct}, \sigma)$

$m' \leftarrow \mathsf{Dec}'(k, \mathsf{ct}')$



Wins if $b' = b$ ***and*** $\neg \exists \mathsf{ct}' = \mathsf{ct}_b$

B_{CPA}

A_{Game_1}

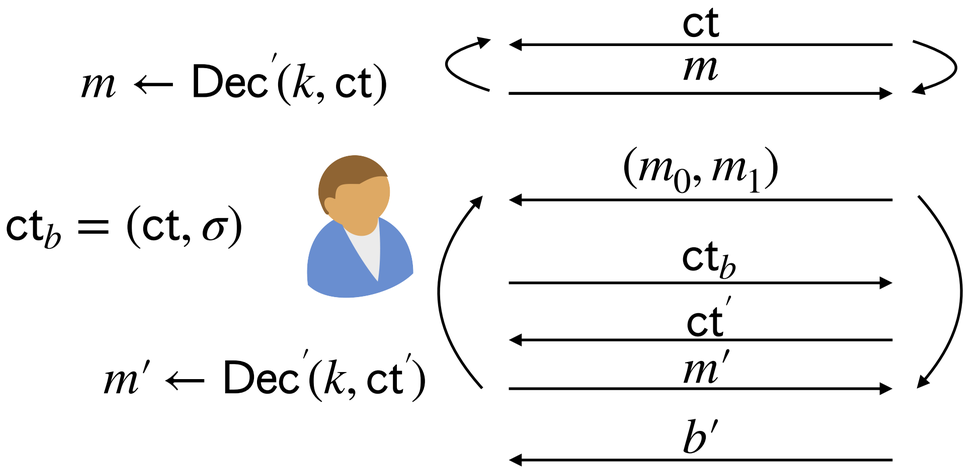
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Ch_{CPA}

Game 1



Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

B_{CPA}

ct

A_{Game_1}

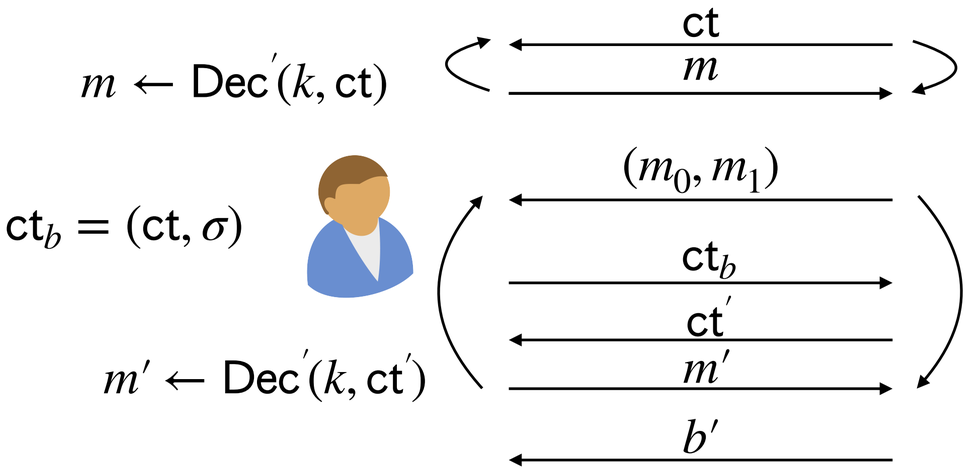
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Ch_{CPA}

Game 1



Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

B_{CPA}

A_{Game_1}

ct

$\perp$

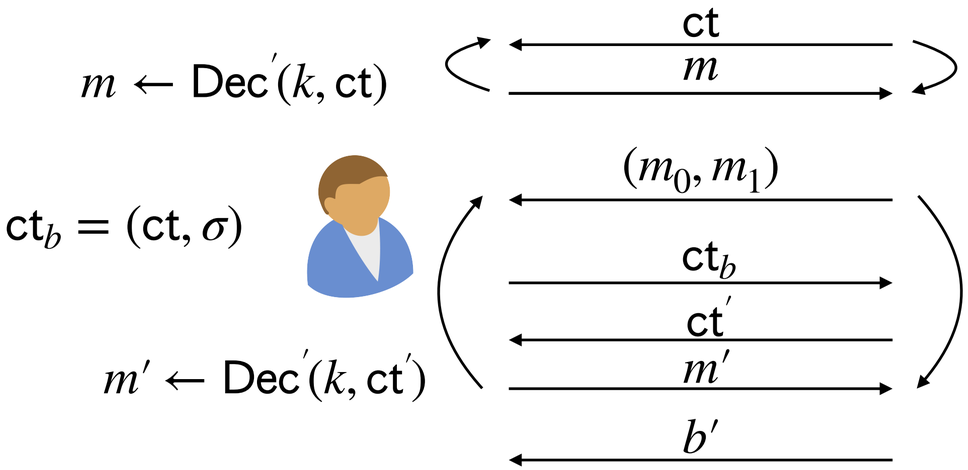
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Ch_{CPA}

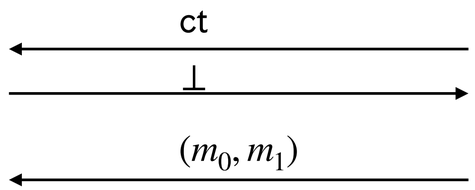
Game 1



Wins if $b' = b$ *and* $\neg\exists \mathsf{ct}' = \mathsf{ct}_b$

B_{CPA}

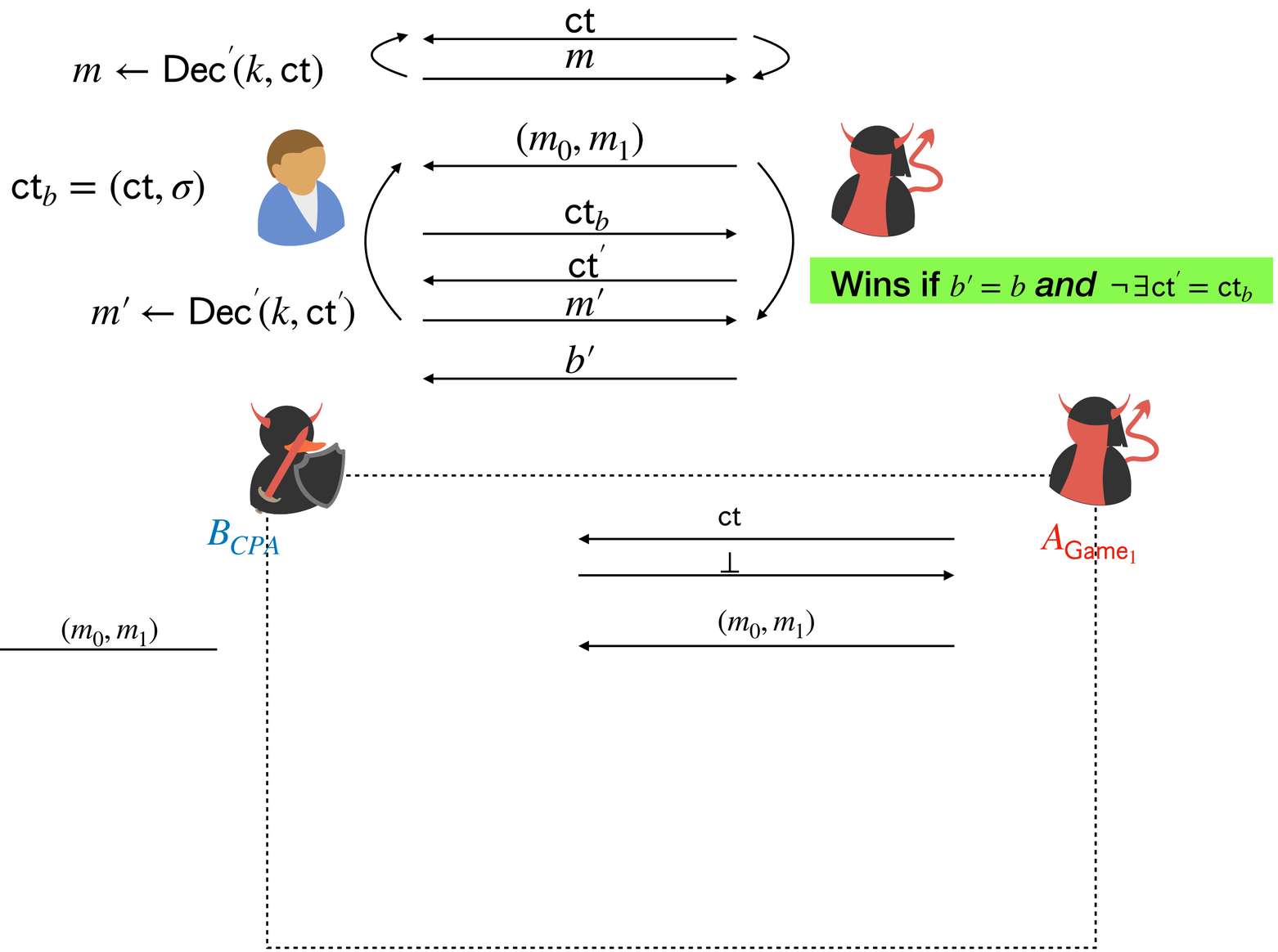
A_{Game_1}



Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

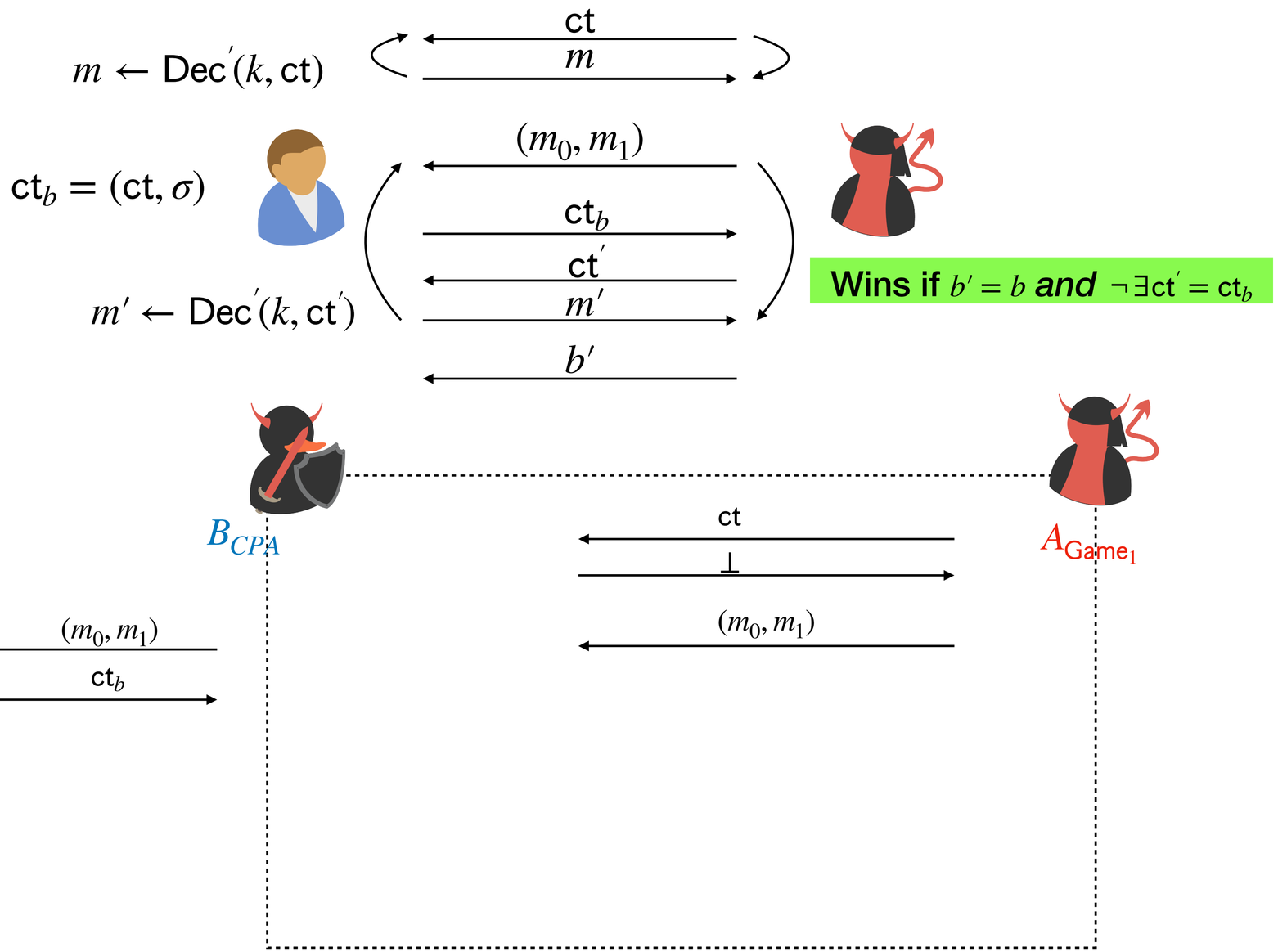


Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1

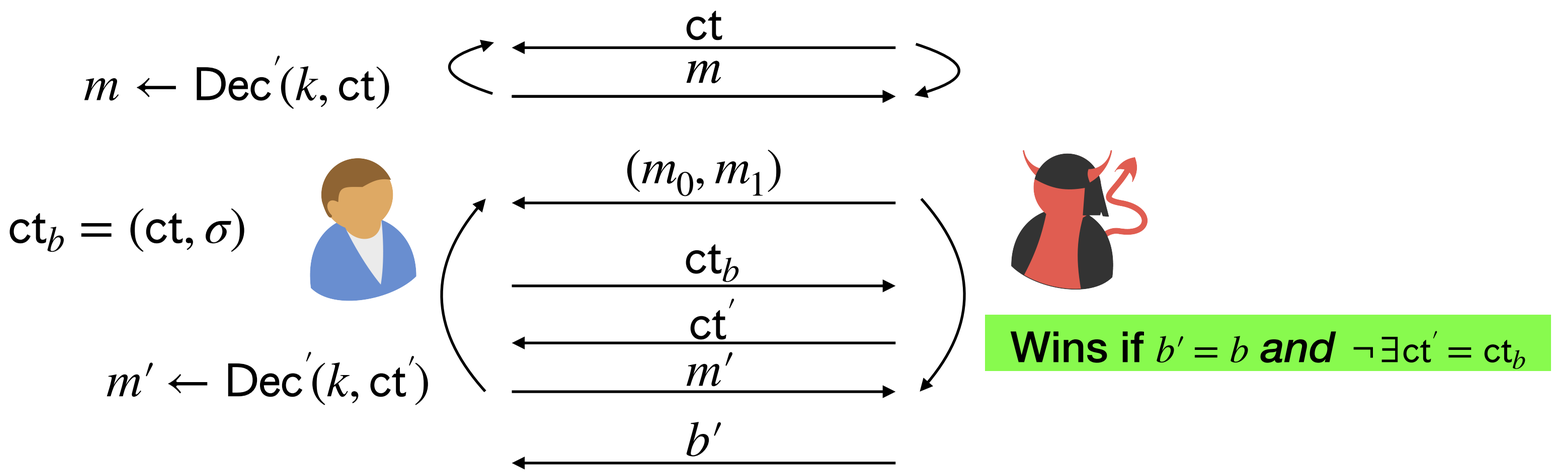


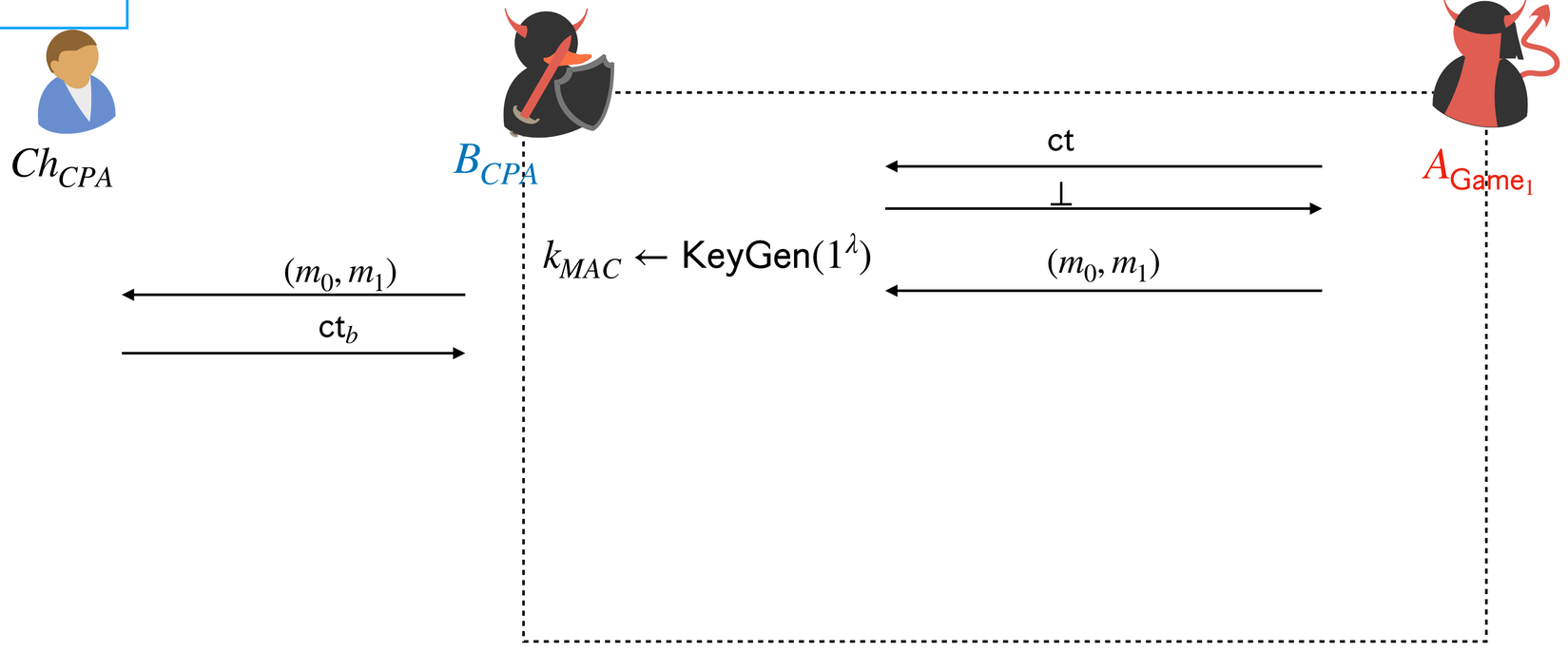
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1

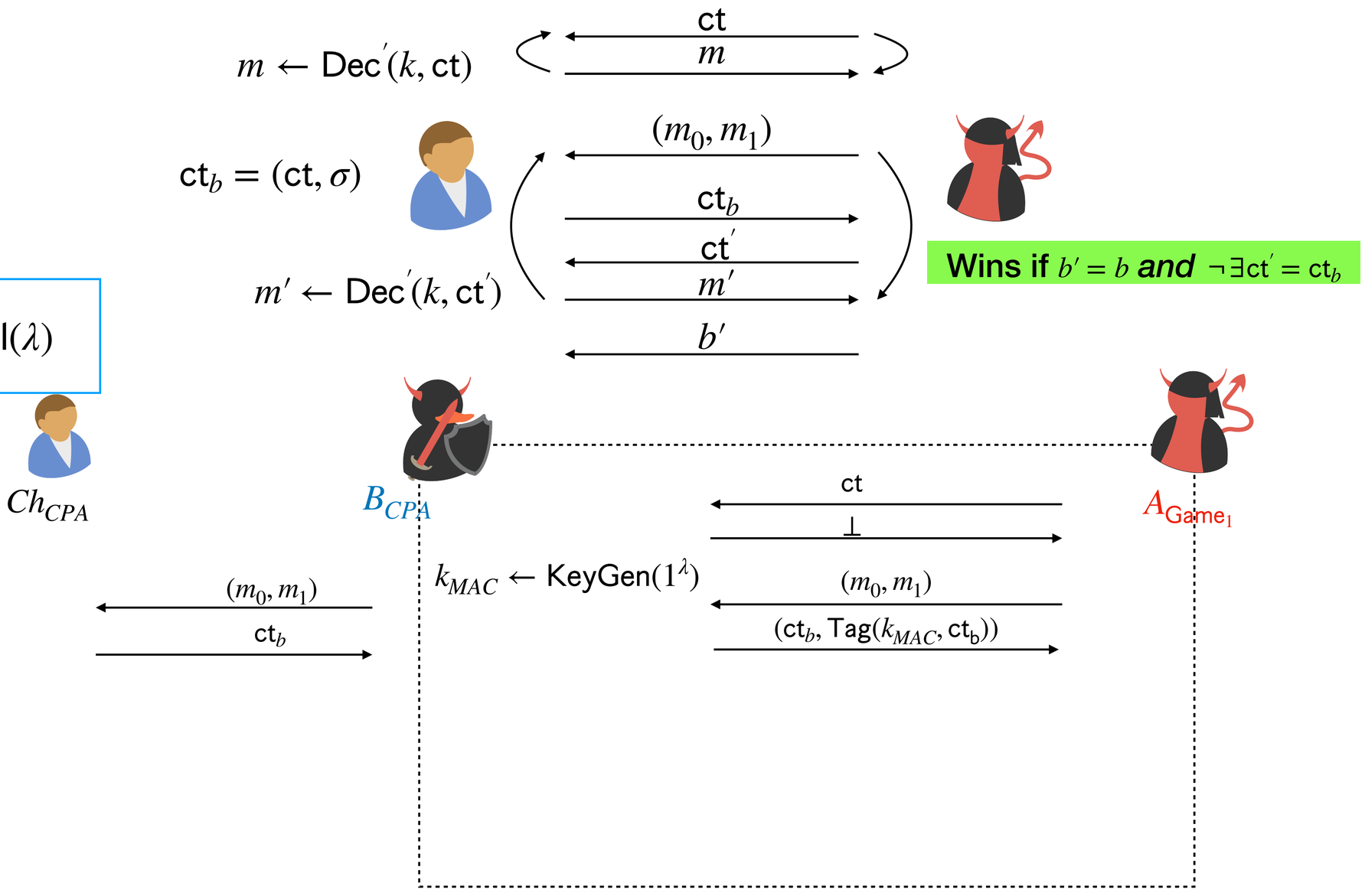




Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

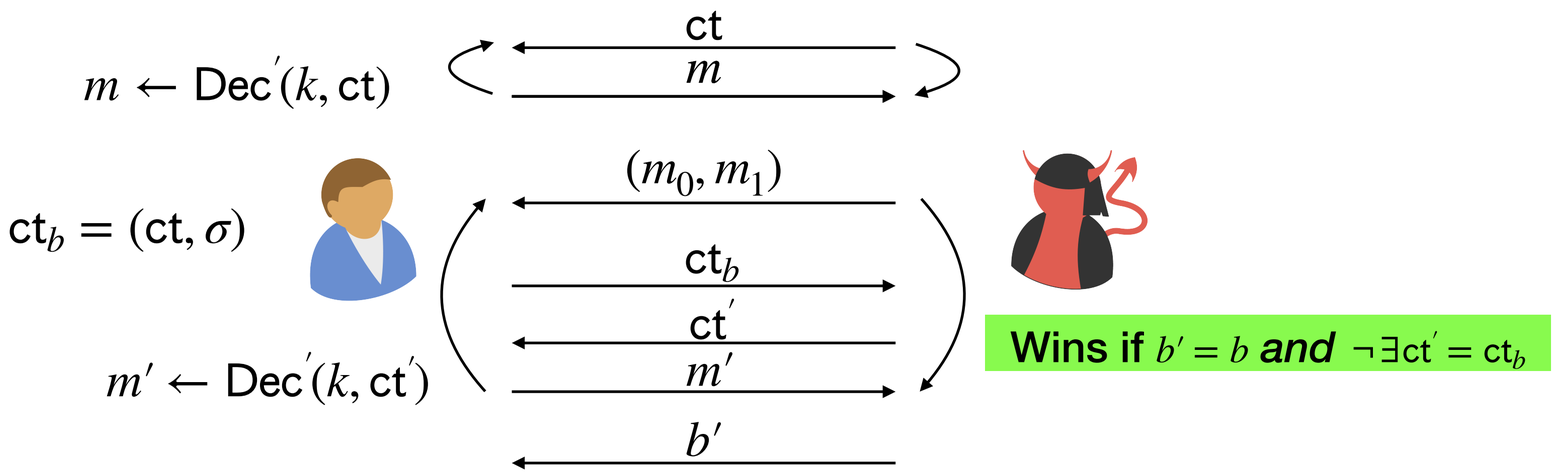


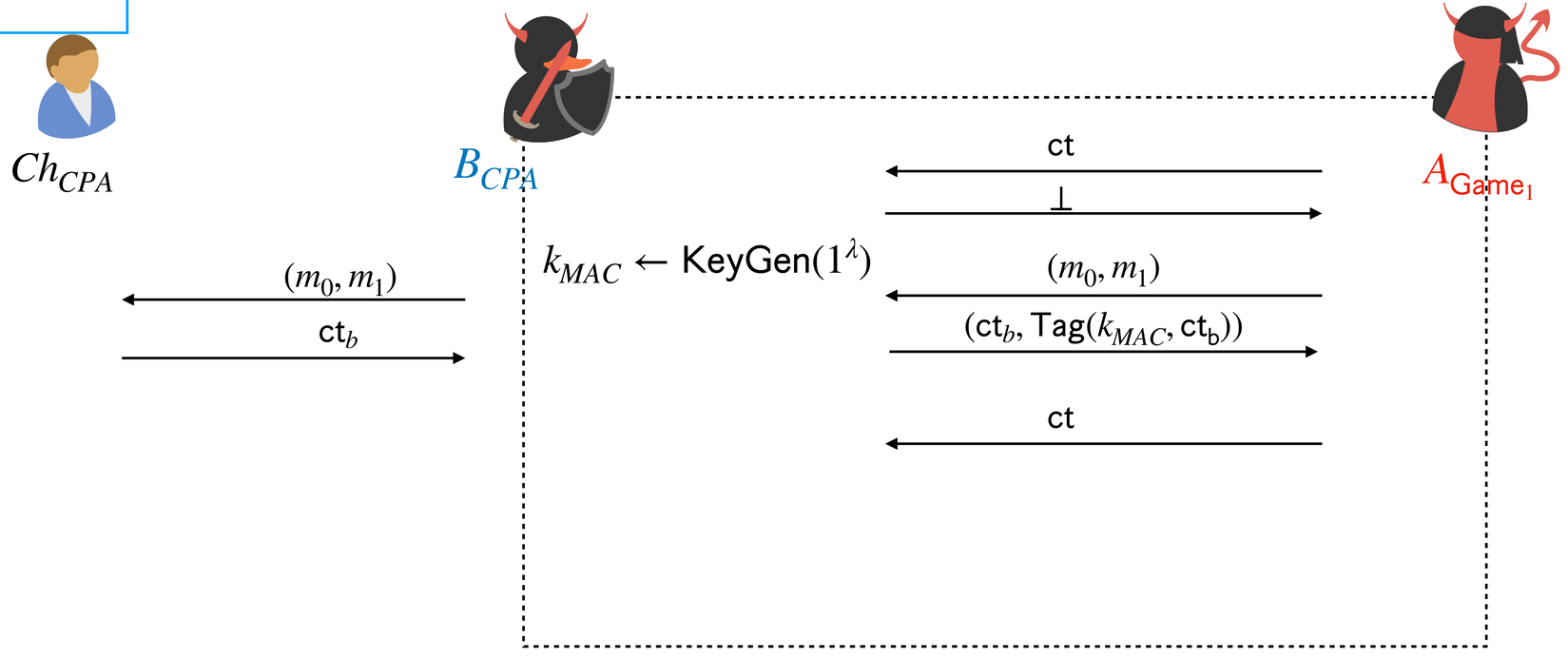
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1



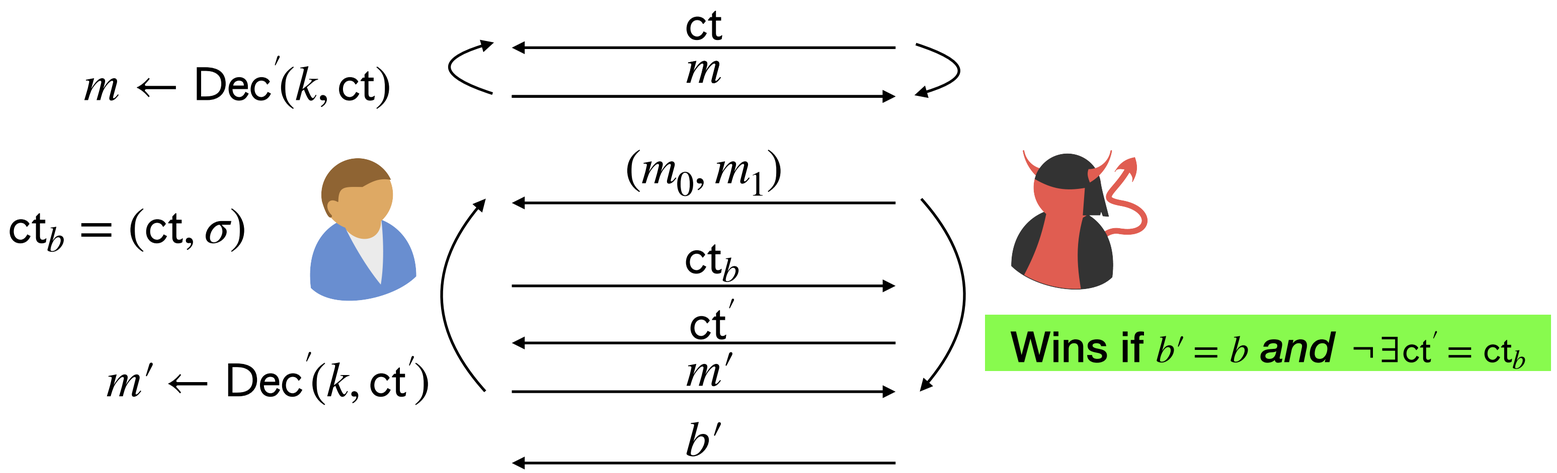


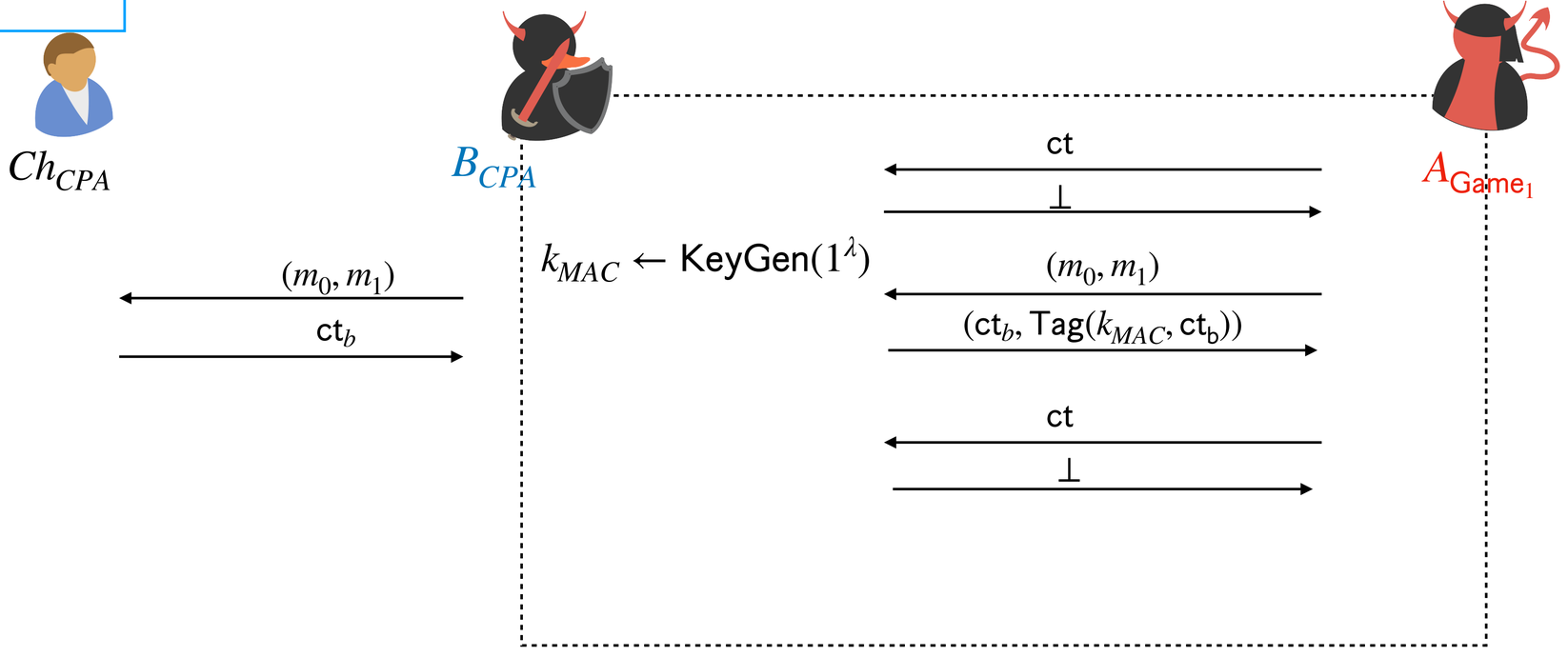
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1



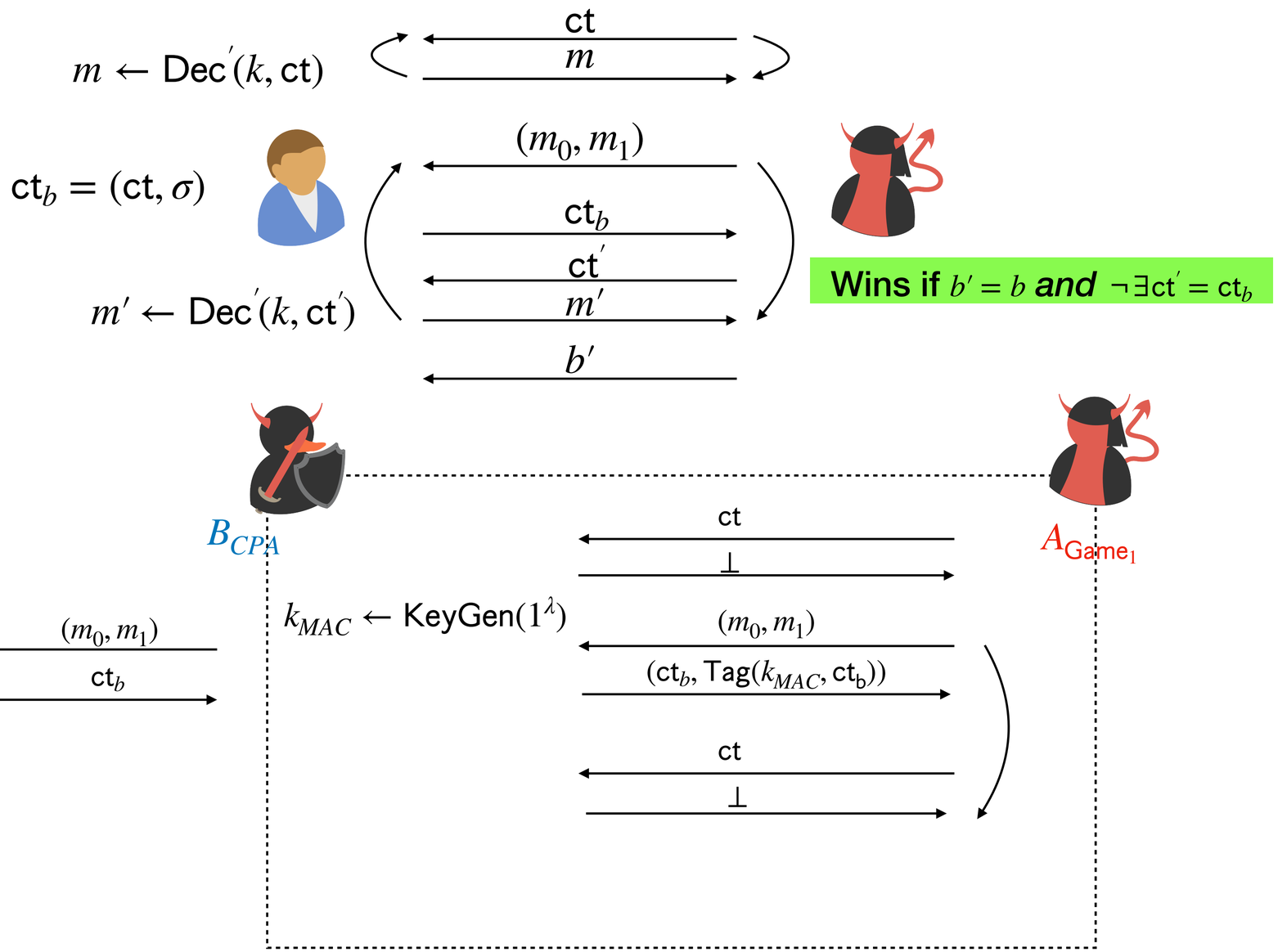


Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1

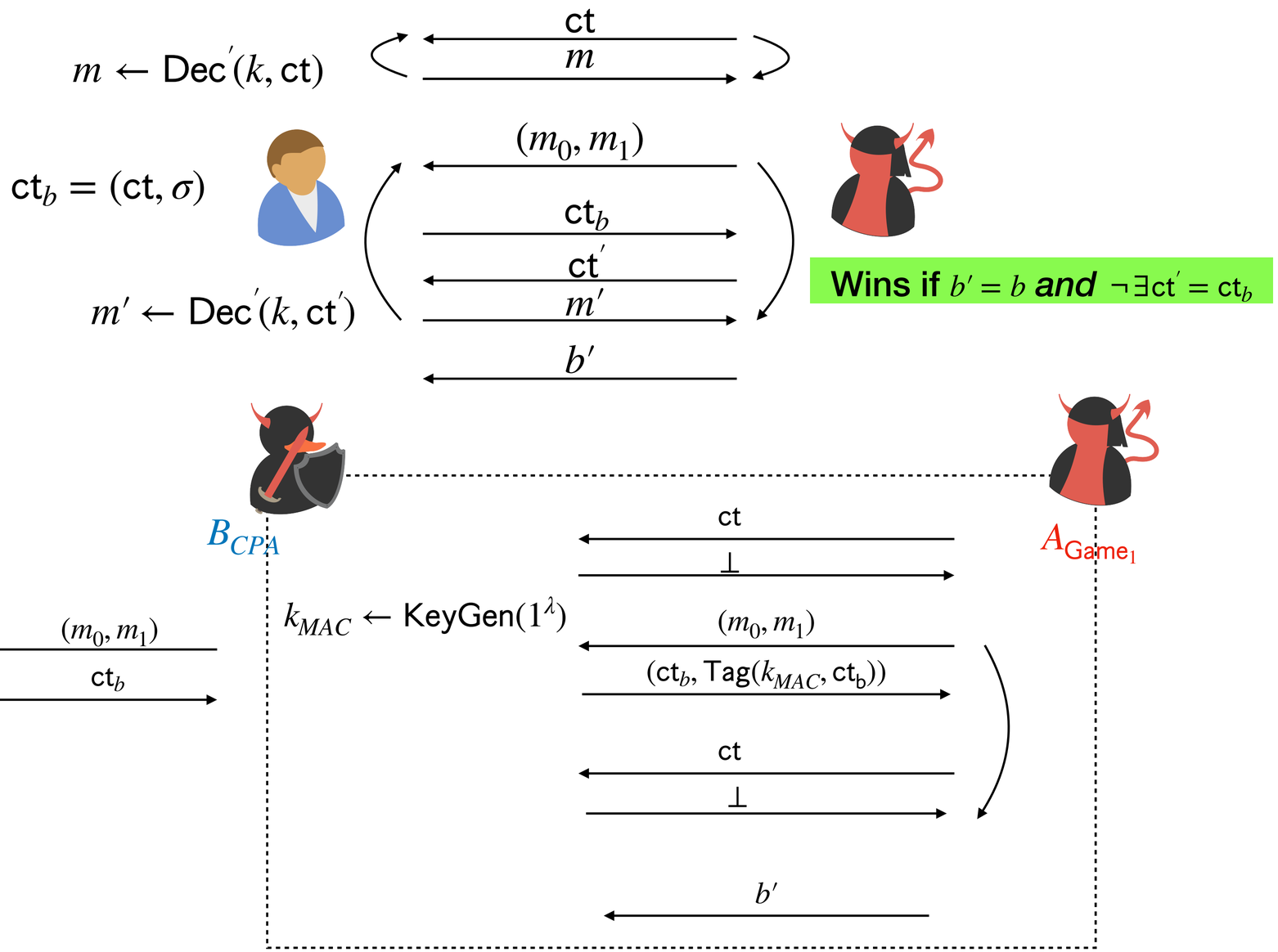


Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1

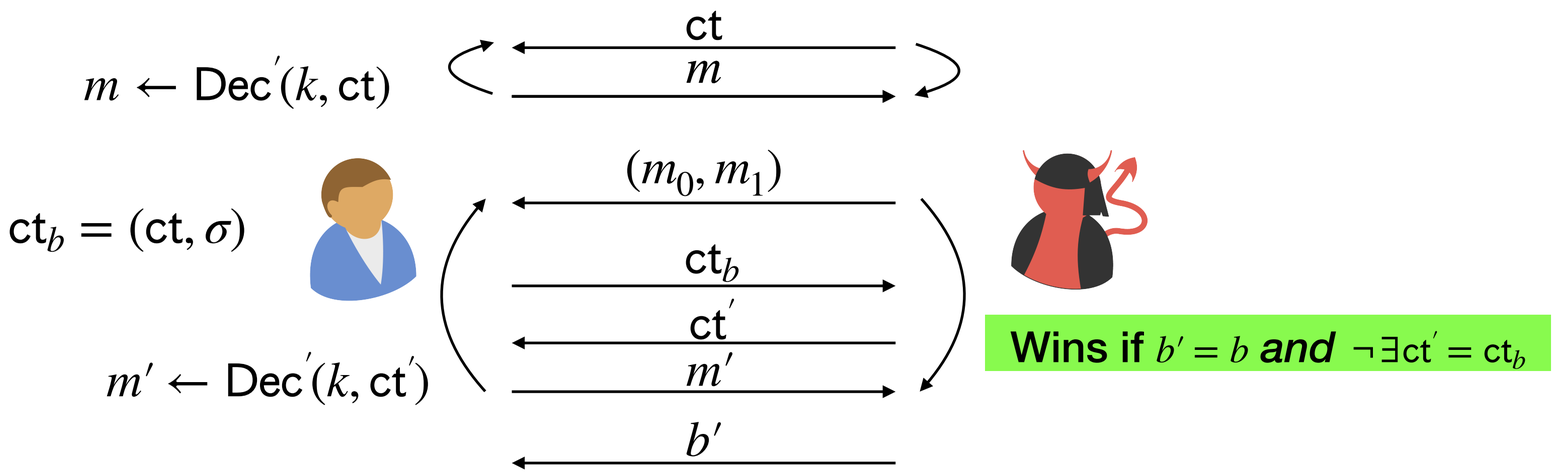


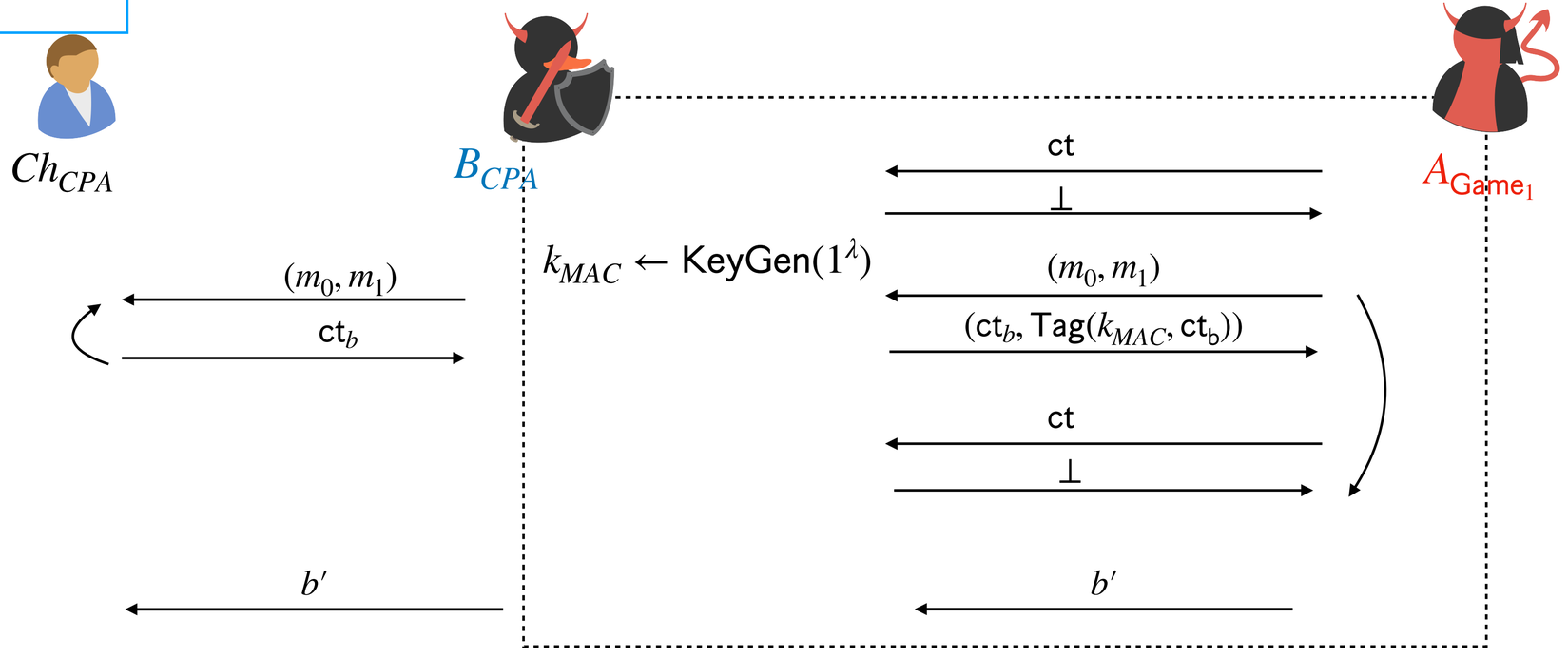
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Game 1





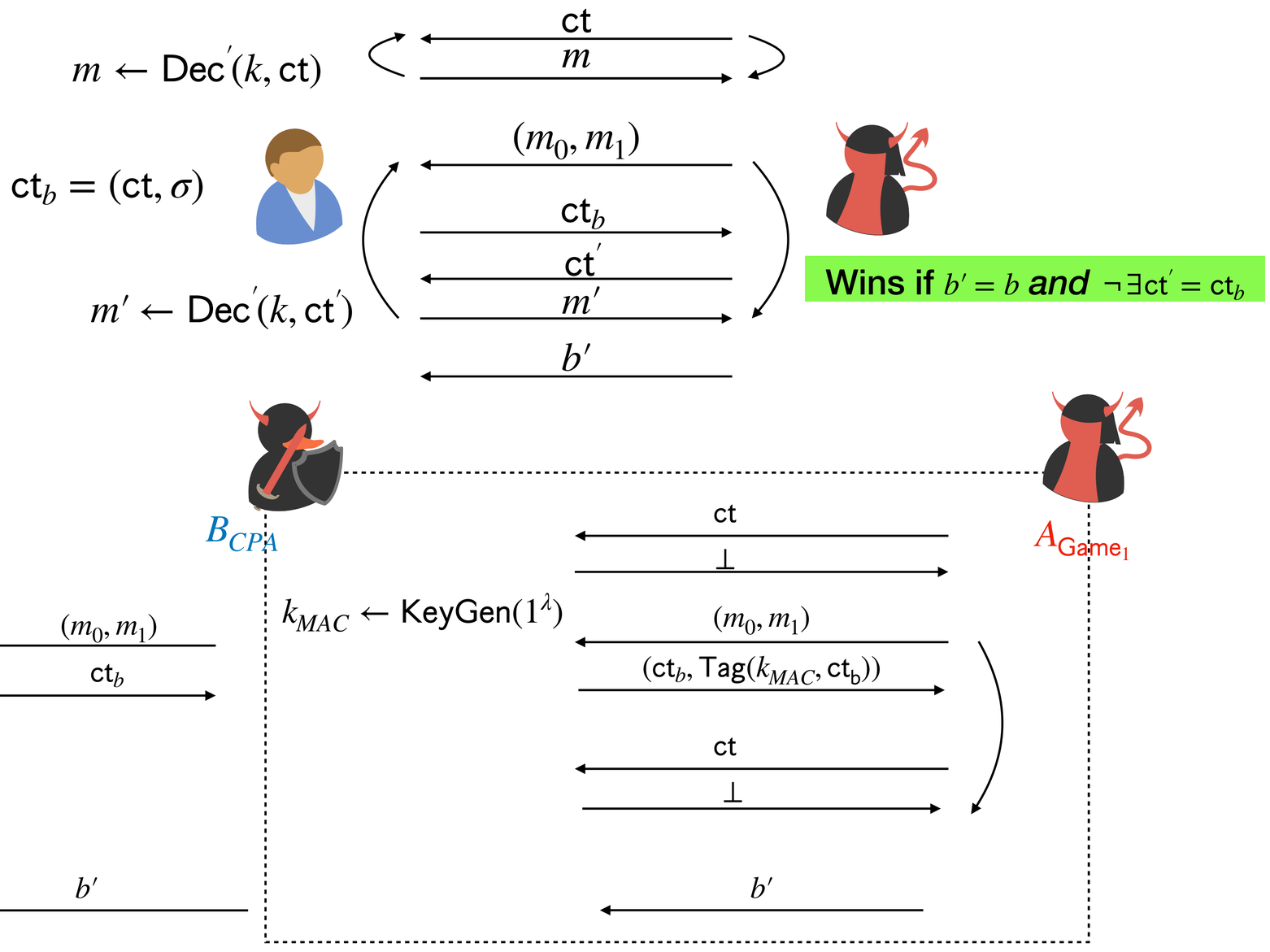
Proof of Security

- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Input mapping: $\mathscr{A}$ sees $\perp$ from decryption queries and ciphertexts with MACs, just like it expects!

Game 1



Proof of Security

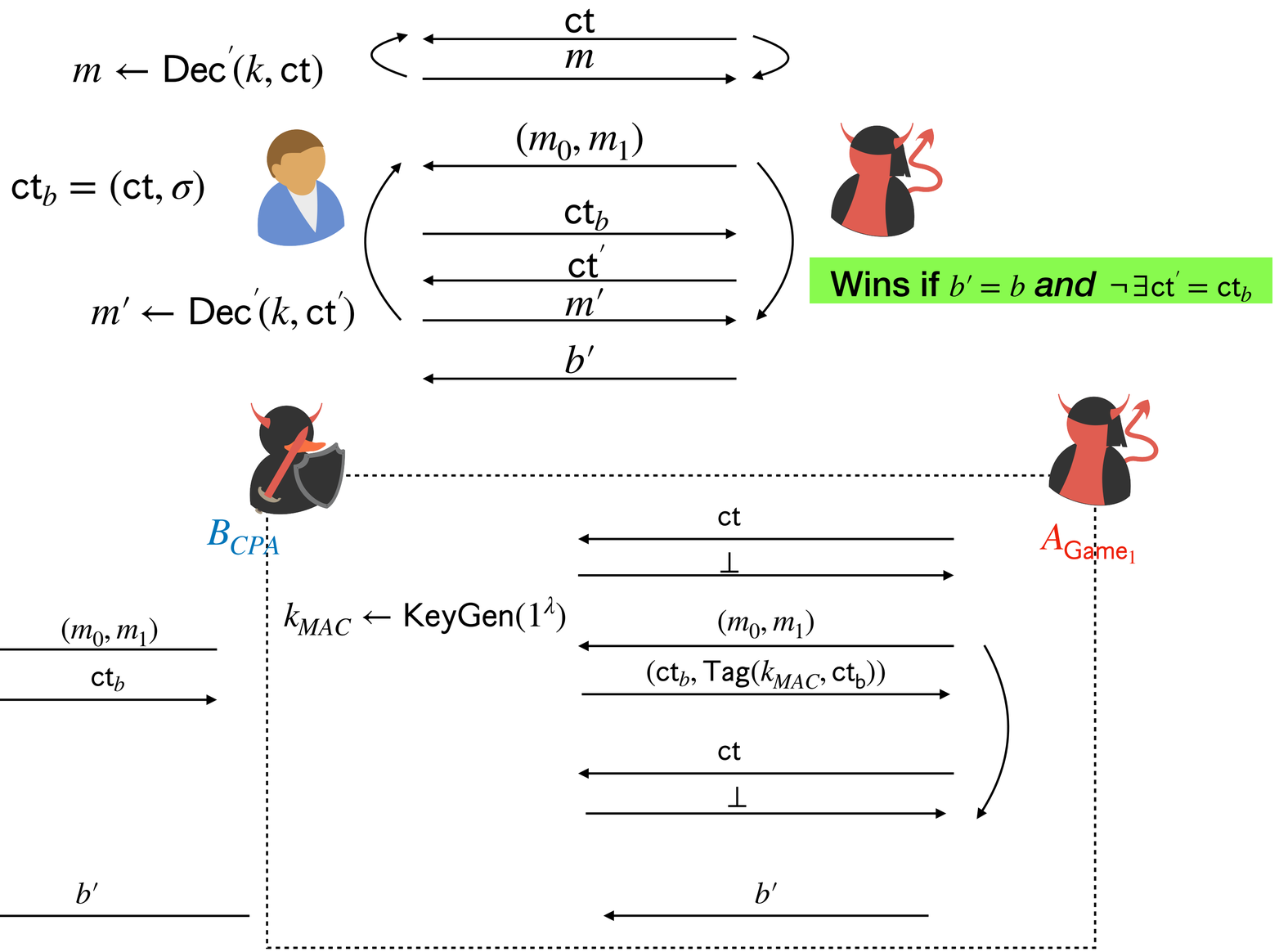
- $\mathsf{Dec}'((k_{MAC}, k_{Enc}), (\mathsf{ct}, \sigma))$:
  - return $\perp$

Claim: $\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] \leq \mathsf{negl}(\lambda)$

Input mapping: $\mathscr{A}$ sees $\perp$ from decryption queries and ciphertexts with MACs, just like it expects!

Therefore:
$\mathsf{Pr}[\mathscr{A}\ \text{wins}\ \mathsf{Game}_1] = \mathsf{Pr}[\mathscr{B}\ \text{wins}\ CPA] = \mathsf{negl}(\lambda)$

Game 1



IND-CCA For Public Key Encryption

IND-CCA For Public Key Encryption

- IND-CCA public key encryption is *much* harder to achieve than the symmetric version

IND-CCA For Public Key Encryption

- IND-CCA public key encryption is *much* harder to achieve than the symmetric version
  - Why? There's no secret we have that tells us if a ciphertext is well formed!

IND-CCA For Public Key Encryption

- IND-CCA public key encryption is *much* harder to achieve than the symmetric version
  - Why? There's no secret we have that tells us if a ciphertext is well formed!
  - Signatures are controlled by the *sender*. The same construction no longer works.

IND-CCA For Public Key Encryption

- IND-CCA public key encryption is *much* harder to achieve than the symmetric version
  - Why? There's no secret we have that tells us if a ciphertext is well formed!
  - Signatures are controlled by the *sender*. The same construction no longer works.
- We will see a scheme that uses NIZKs to prove that ciphertexts are well-formed.

Recap: NIZKs

NIZKs

Recap: NIZKs

NIZKs

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

Recap: NIZKs

NIZKs

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

- CRSGen$(1^\lambda) \rightarrow$ crs

Recap: NIZKs

NIZKs

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

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

Recap: NIZKs

NIZKs

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

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

Recap: NIZKs

NIZKs

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

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

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

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

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 $\hat{P}$

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

Recap: NIZKs

NIZKs

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

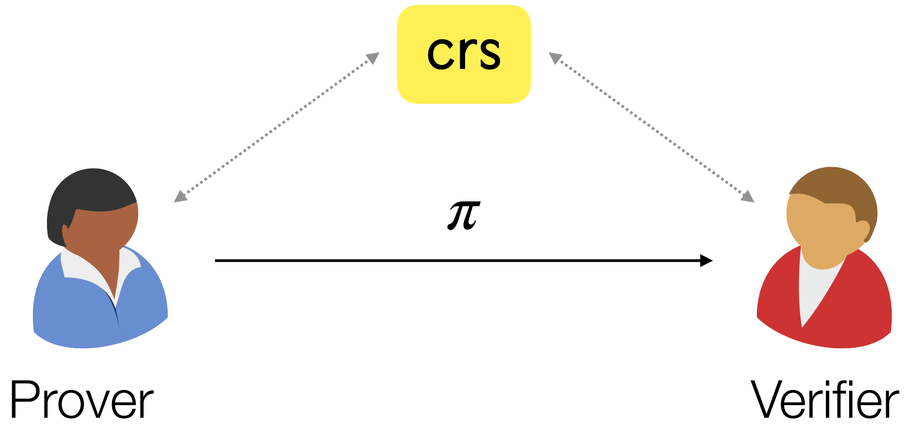
- $\mathsf{CRSGen}(1^\lambda) \to \mathsf{crs}$
- $P(x, \mathsf{crs}) \to \pi$
- $V(\pi, \mathsf{crs}) \to \{0,1\}$

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 $\hat{P}$

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



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

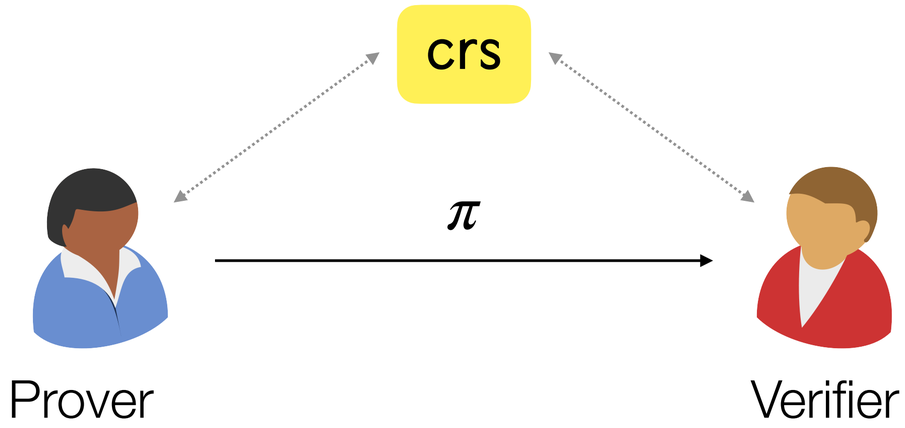
$$\Pr\left[V(\pi) = 1 : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^\lambda) \\ \pi \leftarrow P(x, \mathsf{crs})\end{matrix}\right] = 1$$

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

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

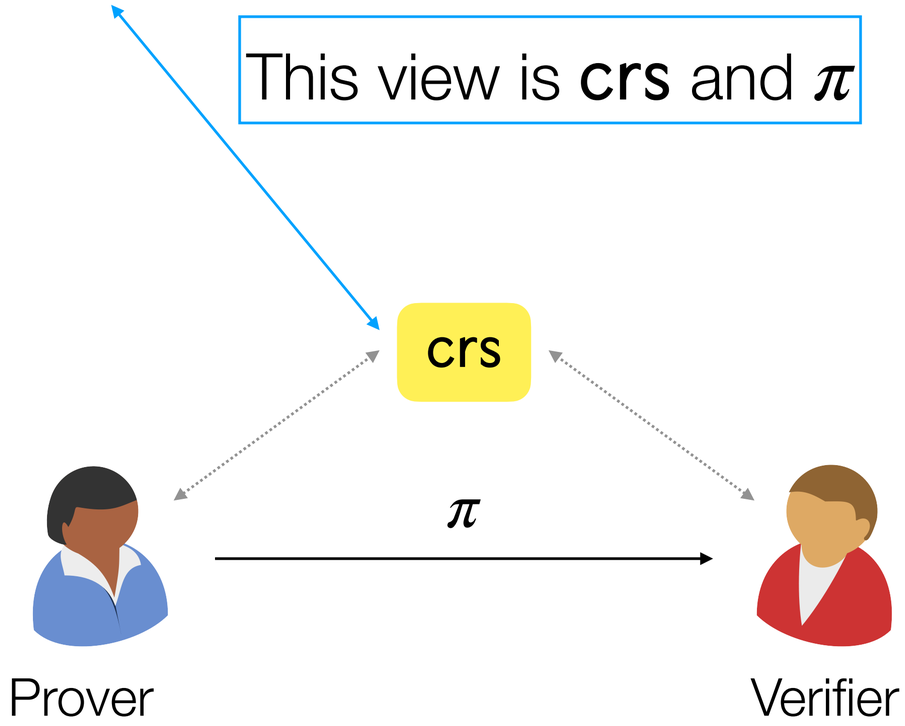
Soundness: for all $x \notin L$ and for all $\hat{P}$

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

NIZKs

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

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



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