# CCA Security

## CS/ECE 407

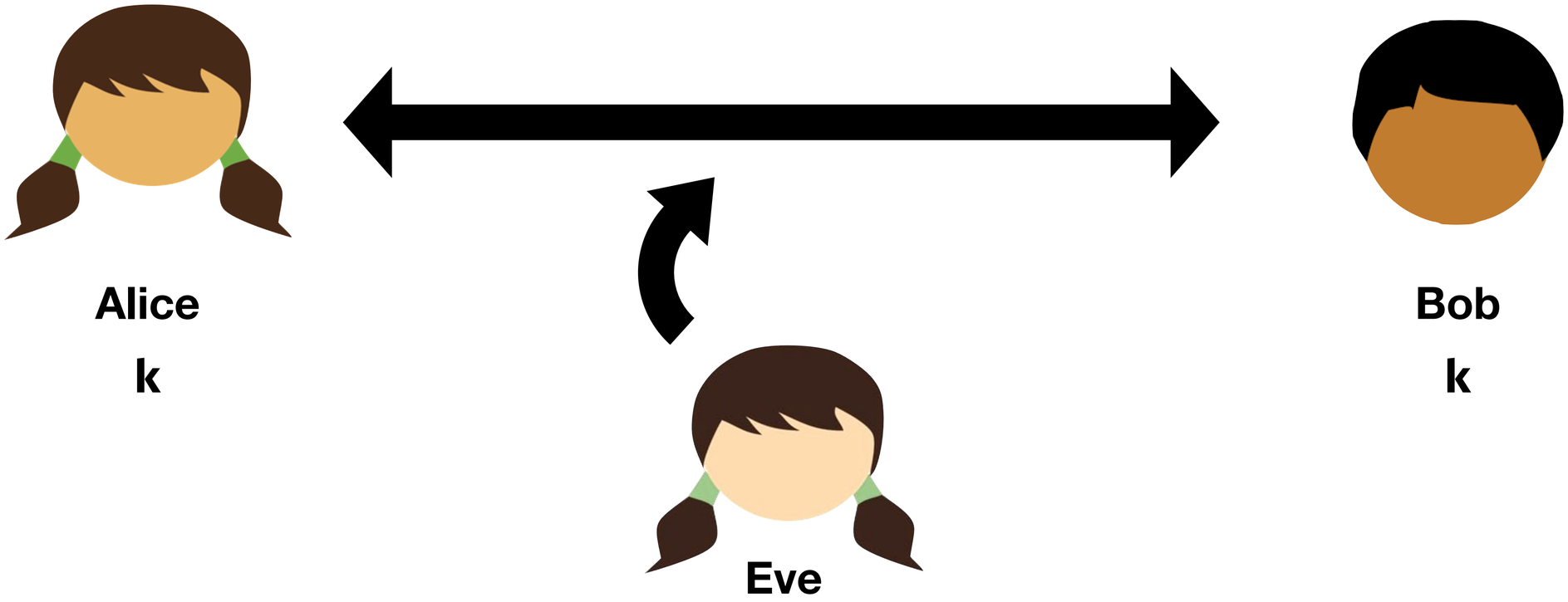
# Today's objectives

See intricate attacks by an active adversary

Understand the definition of **Security against Chosen Ciphertext Attacks (CCA)**

Explain why CTR/CBC mode block cipher is not CCA secure

Connect CCA security with **malleability**




## Confidentiality

Can Alice and Bob prevent Eve from listening?

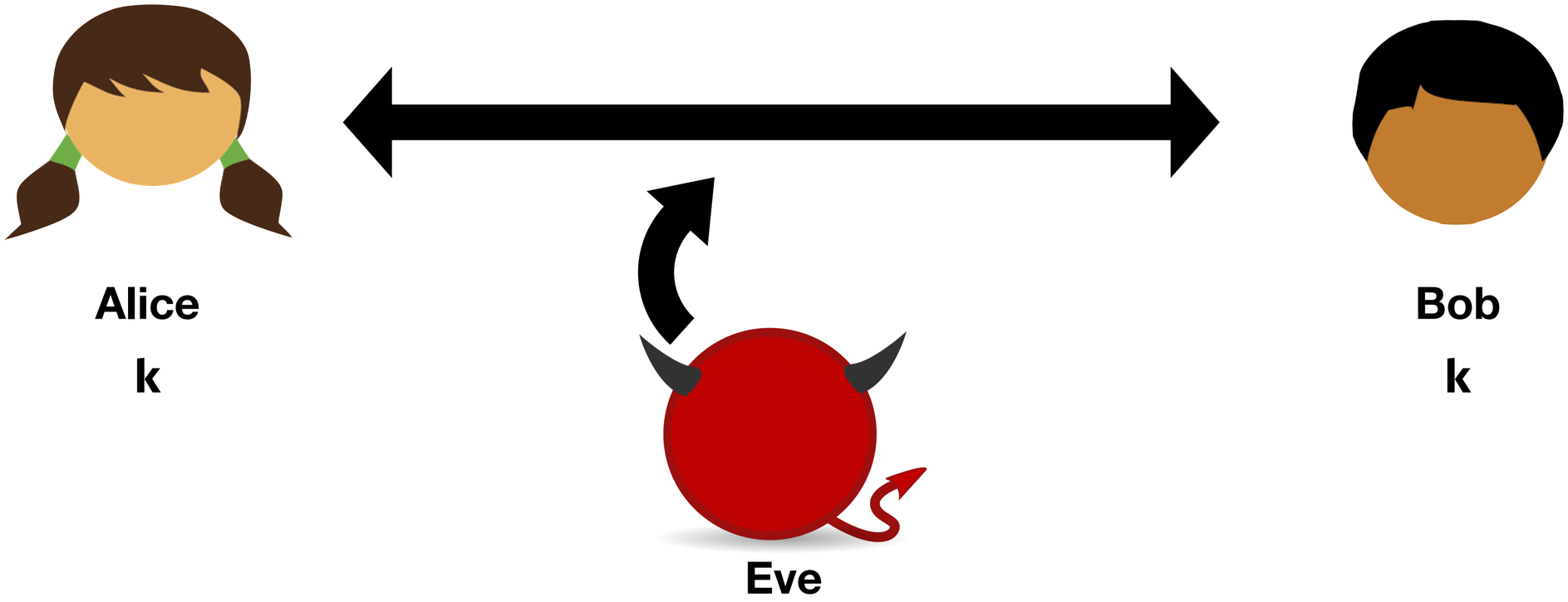
# A cipher (Enc, Dec) has **ciphertext indistinguishability against a chosen plaintext attack (CPA)** if:

Let $Enc_L(k, m_0, m_1) = Enc(k, m_0)$

Let $Enc_R(k, m_0, m_1) = Enc(k, m_1)$

Where $m_0, m_1$ are of the same length

$$\left\{ Enc_L(k, \cdot, \cdot) \;\middle|\; k \leftarrow K \right\} \approx \left\{ Enc_R(k, \cdot, \cdot) \;\middle|\; k \leftarrow K \right\}$$



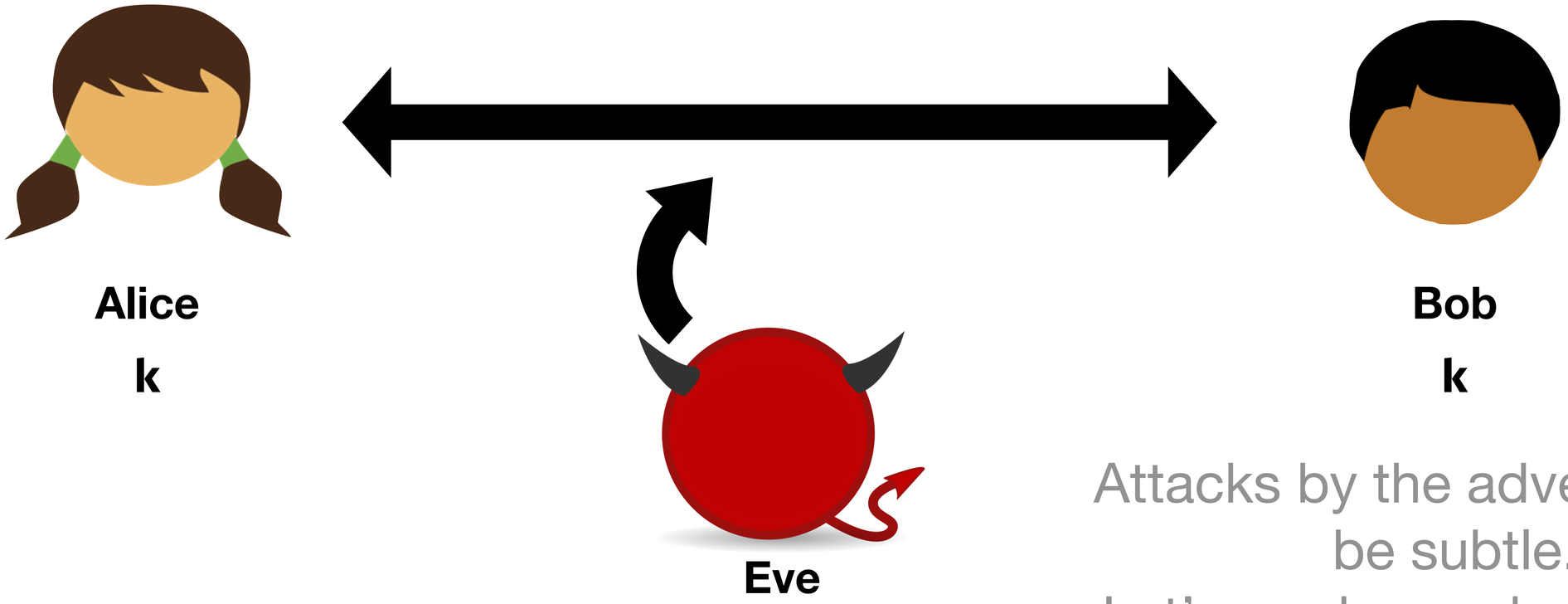

**Confidentiality**

Can Alice and Bob prevent Eve from listening?

**Authenticity**

Can Bob be sure Eve did not send the message?

Can Bob be sure Eve did not alter a message from Alice?




Attacks by the adversary can be subtle.
Let's see how adversary can send messages to break a CPA-secure scheme

**Confidentiality**

Can Alice and Bob prevent Eve from listening?

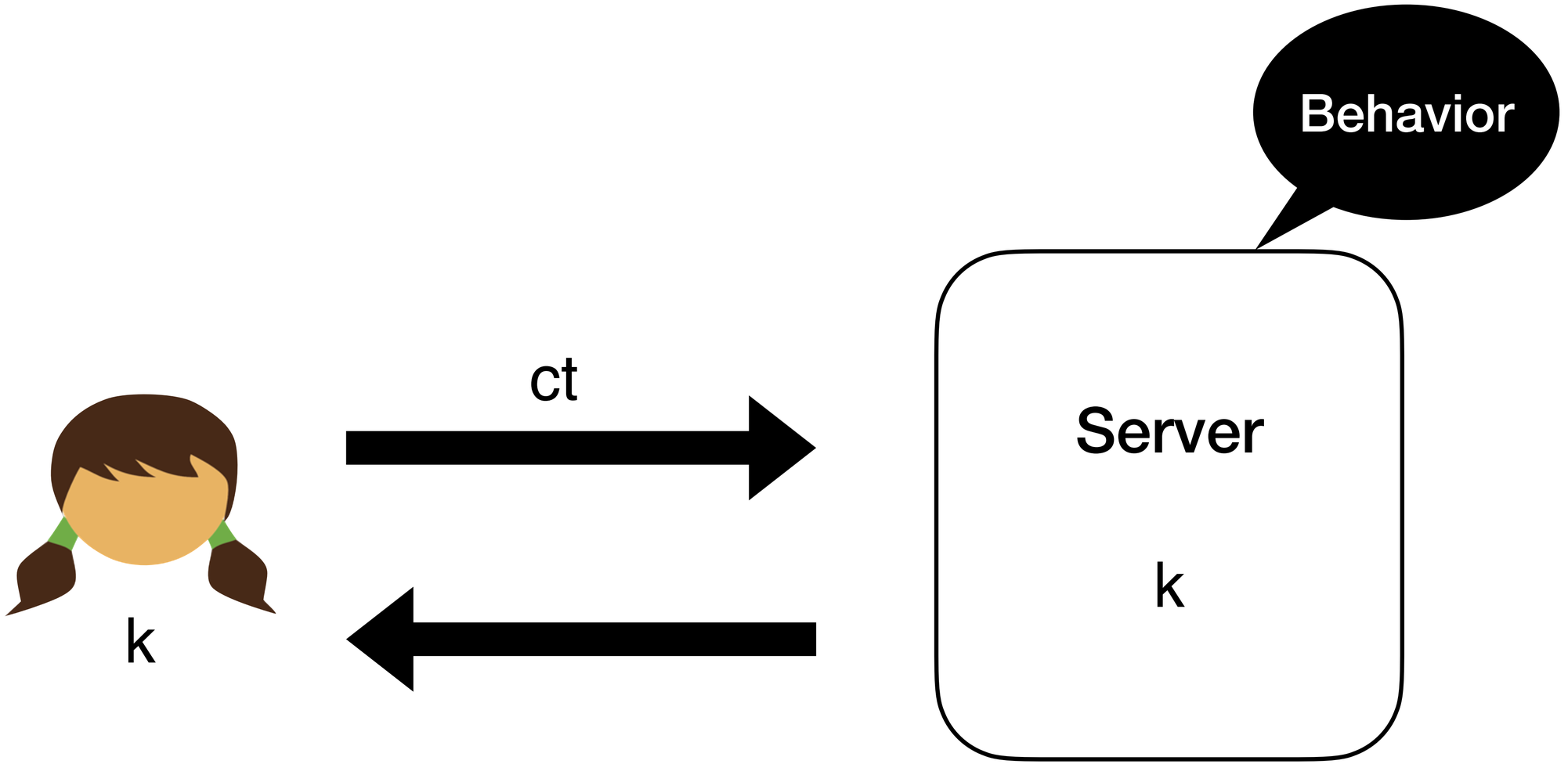
**Authenticity**

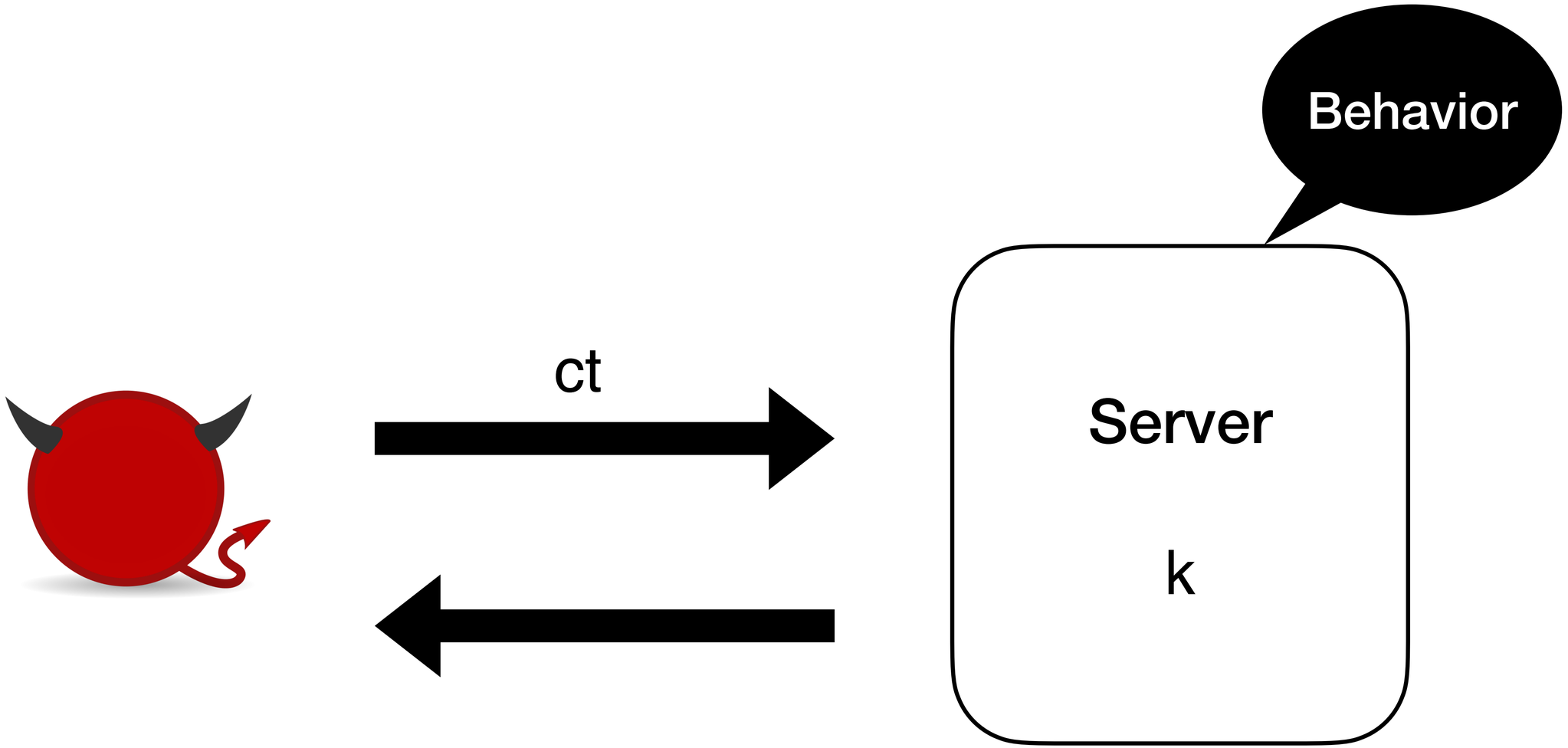
Can Bob be sure Eve did not send the message?
Can Bob be sure Eve did not alter a message from Alice?

k

Server

k


Behavior
ct
Server
k
k



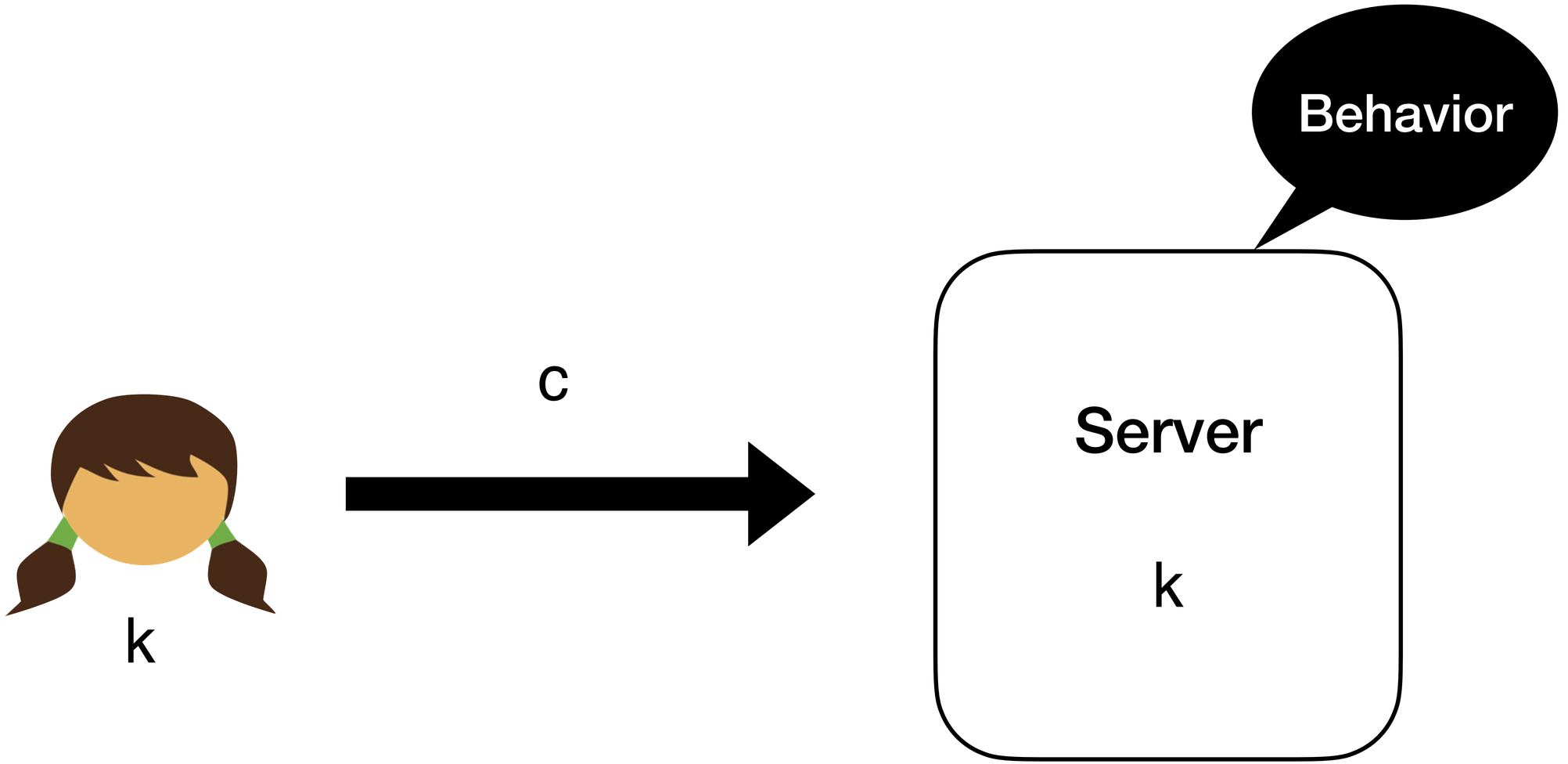

**Adversary can try to send ciphertexts, and see what happens**

# Warm-up:
# **Null Oracle Attack**

# Counter (CTR) Mode

```
Enc(k, m_1 | … | m_n):
  r ←$ {0,1}^λ
  for i in 1 to n
    c_i ← F(k, r + i) ⊕ m_i
  return r | c_1 | … | c_n

Dec(k, r | c_1 | … | c_n):
  for i in 1 to n
    m_i ← F(k, r + i) ⊕ c_i
return m_1 | … | m_n
```


Behavior
c
Server
k
k

k

c = CTR(k, m)

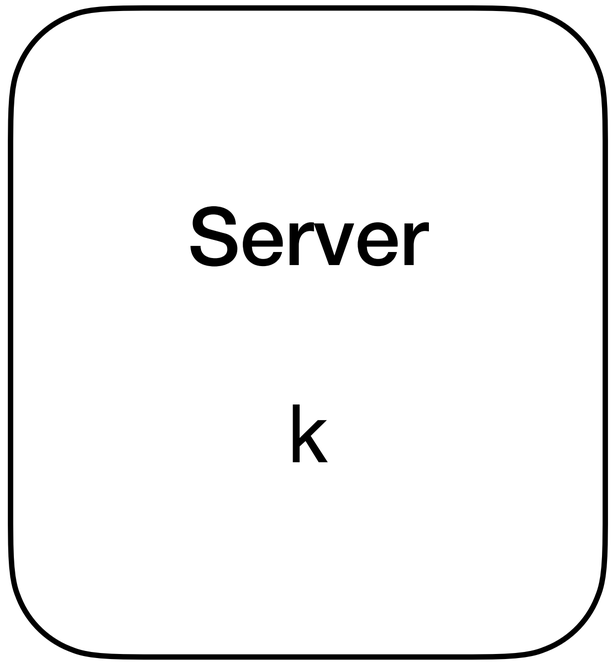
**Server**

k

m = $CTR^{-1}$(k, c)

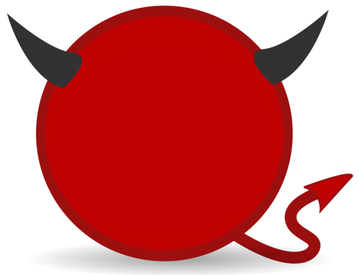
c = CTR(k, m)

k




If m contains
0×00, ERROR

m = $CTR^{-1}$(k, c)



c

Server

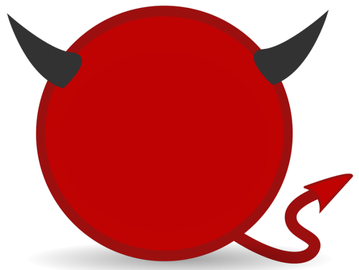
k

If m contains
0×00, ERROR

m = $CTR^{-1}$(k, c)

k

c



Server

k

If m contains
0×00, ERROR

m = $CTR^{-1}$(k, c)

# More Realistic:
# **Padding Attack**

# Cipher Block Chaining (CBC) Mode

```
Enc(k, m_1 | … | m_n):
  c_0 ←$ {0,1}^λ
  for i in 1 to n
    c_i ← F(k, m_i ⊕ c_i-1)
  return c_0 | c_1 | … | c_n

Dec(k, c_0 | c_1 | … | c_n):
  for i in 1 to n
    m_i ← F^-1(k, c_i) ⊕ c_i-1
  return m_1 | … | m_n
```

Decrypting i-th ciphertext is possible from only a small part of the cipher text.

# Padding:

`pad(m)` : takes input message, outputs string whose length is multiple of block length

`unpad(m)` : inverse of pad

**Correctness:** `unpad(pad(m)) = m`

**Suggestion:** Pad by a single 1, then pad with 0s until multiple of block length
To unpad, strip last 1 and all following 0s

# Padding:

`pad(m)` : takes input message, outputs string whose length is multiple of block length

`unpad(m)` : inverse of pad

**Correctness:** `unpad(pad(m)) = m`

**Suggestion:** Pad by zeros, then write a single byte at the end that says how much padding was added.

To unpad, read last byte, check padding is **valid**, then strip that many bytes

**X9.23 ANSI Padding Standard**

Behavior

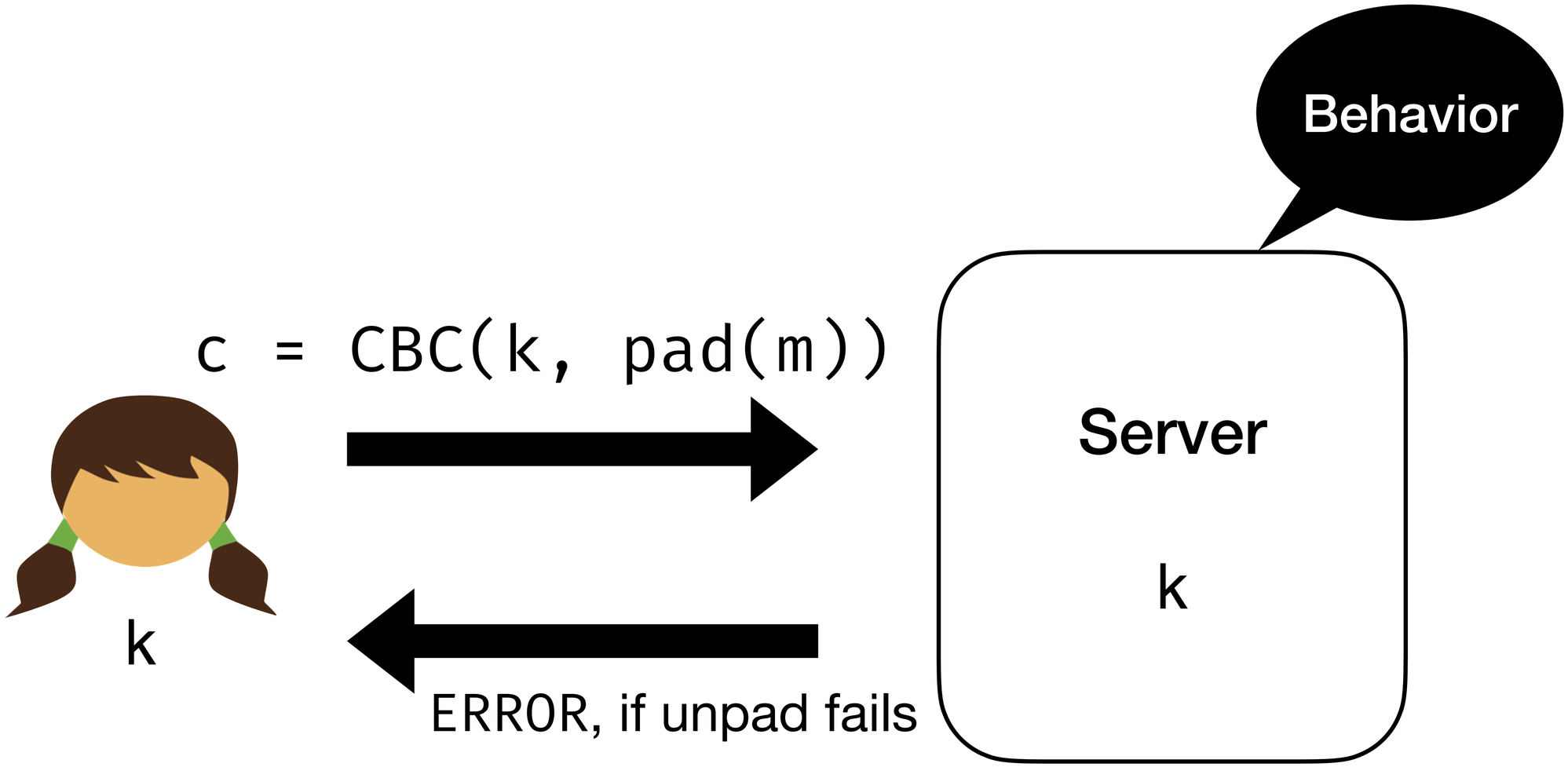
c = CBC(k, pad(m))

k

Server

k

m = unpad($CBC^{-1}$(k, c))


Behavior
c = CBC(k, pad(m))
Server
k
k
ERROR, if unpad fails
m = unpad(CBC⁻¹(k, c))

# Padding Attack

Why didn't CPA security protect us?

CBC Mode encryption is ***malleable:***

Given ciphertext `c` encoding `m`, one can apply some function `f` to `c` such that `f(c)` decrypts to `m'`, and the relationship between `m` and `m'` is *predictable*

Adversary's attack was on **decryption**, and CPA security says **nothing** about decryption

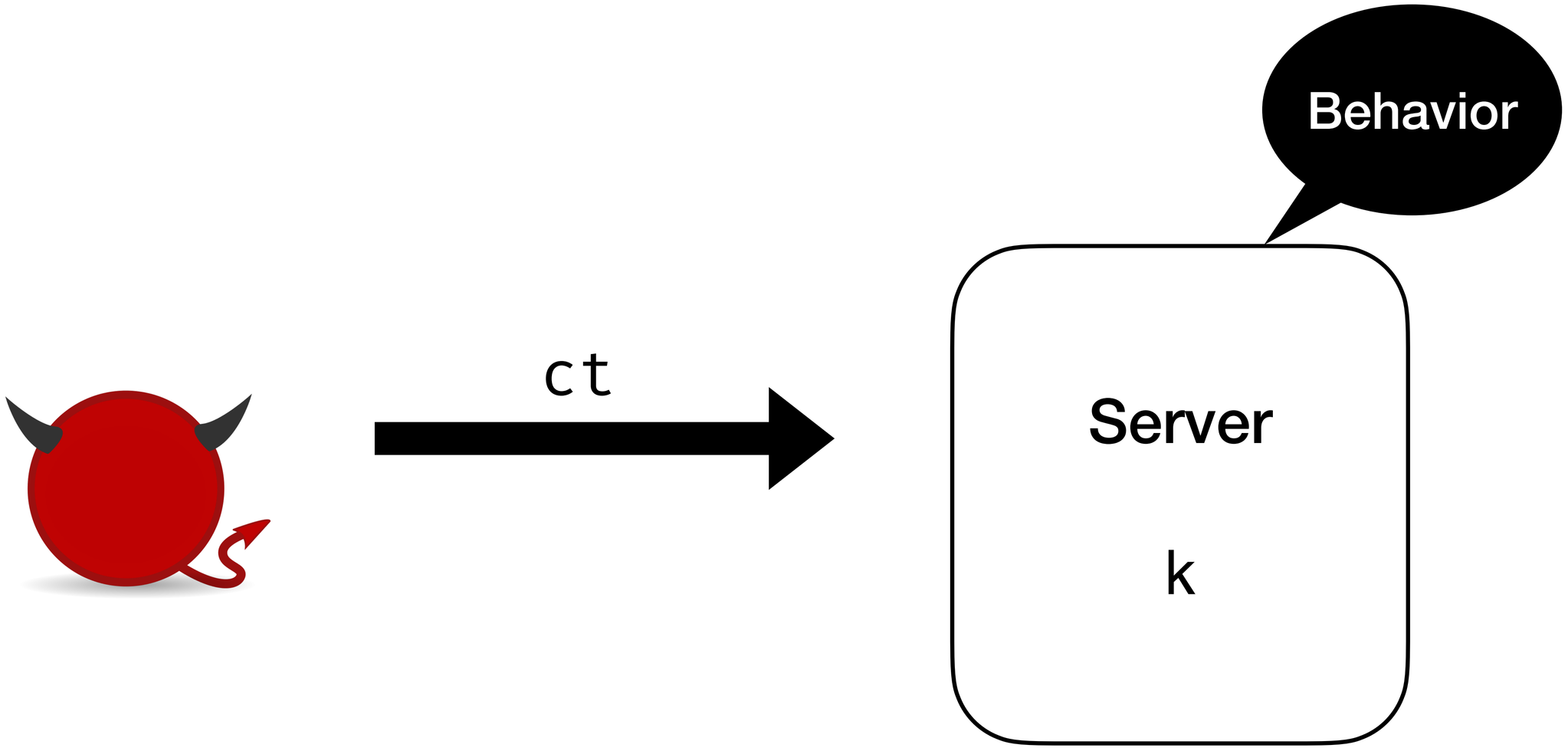
# A cipher (Enc, Dec) has **ciphertext indistinguishability against a chosen plaintext attack (CPA)** if:

Let $Enc_L(k, m_0, m_1) = Enc(k, m_0)$

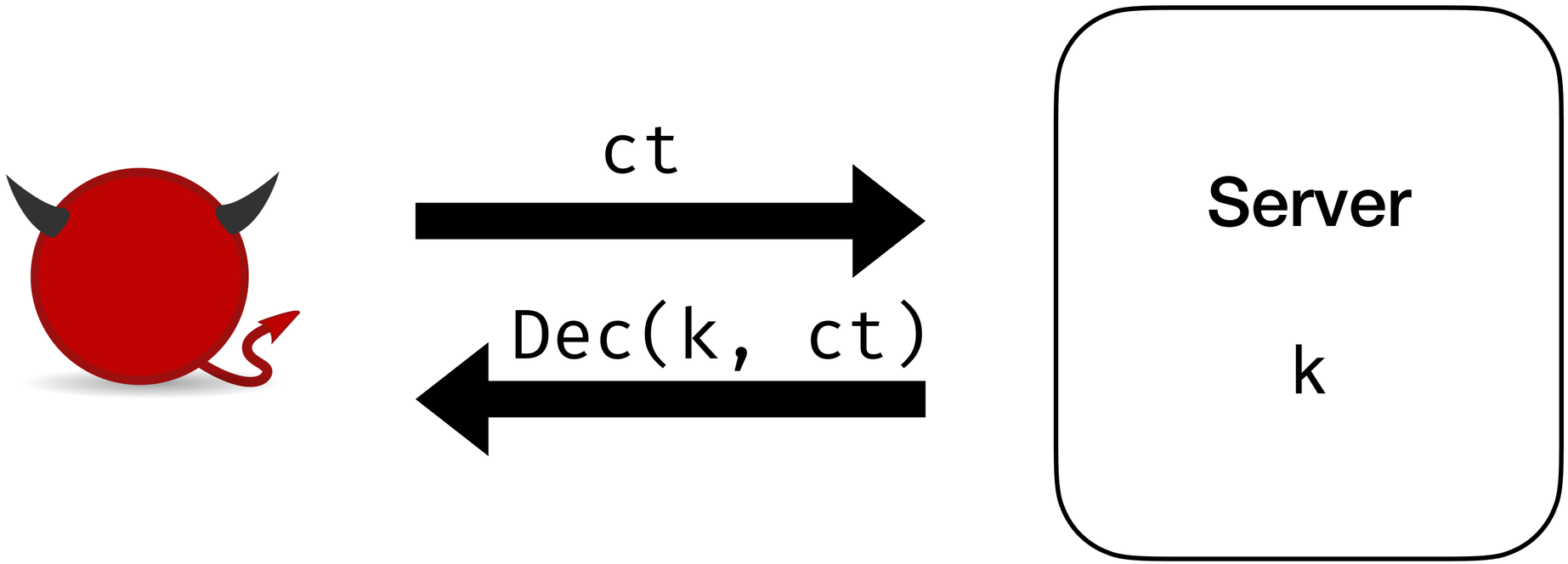
Let $Enc_R(k, m_0, m_1) = Enc(k, m_1)$

Where $m_0, m_1$ are of the same length

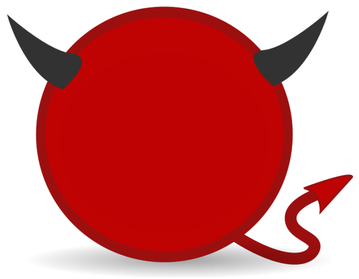
$$\left\{ Enc_L(k, \cdot, \cdot) \;\middle|\; k \leftarrow K \right\} \approx \left\{ Enc_R(k, \cdot, \cdot) \;\middle|\; k \leftarrow K \right\}$$




**security notion should capture**
**Adversary's ability to try ciphertexts**




**security notion should capture**
**Adversary's ability to try ciphertexts**



$m_0, m_1$

$Enc(k, m_b)$

$ct$

$Dec(k, ct)$

**Server**

$b, k$

**CCA Security**

A cipher (Enc, Dec) has **security against a chosen plaintext attack (CPA)** if:

```
k ←$ {0,1}^λ

encL(m0, m1):
  if |m0| ≠ |m1|:
    return error
  ct ← Enc(k, m0)
  return ct
```

$\overset{c}{\approx}$

```
k ←$ {0,1}^λ

encR(m0, m1):
  if |m0| ≠ |m1|:
    return error
  ct ← Enc(k, m1)
  return ct
```

# A cipher (KeyGen, Enc, Dec) has **security against a chosen ciphertext attack (CCA)** if:

```
k ←$ {0,1}^λ
S ← empty-set

encL(m0, m1):
  c ← Enc(k, m0)
  S ← S ∪ {c}
  return c

decrypt(c):
  if c ∈ S:
    return error
  return Dec(k, c)
```

$\overset{c}{\approx}$

```
k ←$ {0,1}^λ
S ← empty-set

encR(m0, m1):
  c ← Enc(k, m1)
  S ← S ∪ {c}
  return c

decrypt(c):
  if c ∈ S:
    return error
  return Dec(k, c)
```

# A cipher (KeyGen, Enc, Dec) has **security against a chosen ciphertext attack (CCA)** if:

```
k ← {0,1}^λ
S ← empty-set

encL(m0, m1):
  c ← Enc(k, m0)
  S ← S ∪ {c}
  return c

decrypt(c):
  if c ∈ S:
    return error
  return Dec(k, c)
```

$\overset{c}{\approx}$

```
k ← {0,1}^λ
S ← empty-set

encR(m0, m1):
  c ← Enc(k, m1)
  S ← S ∪ {c}
  return c

decrypt(c):
  if c ∈ S:
    return error
  return Dec(k, c)
```

Every CCA-secure scheme is also CPA-secure

# Today's objectives

See intricate attacks by an active adversary

Understand the definition of **Security against Chosen Ciphertext Attacks (CCA)**

Explain why CTR/CBC mode block cipher is not CCA secure

Connect CCA security with **malleability**