Message Authentication Codes

CS/ECE 407

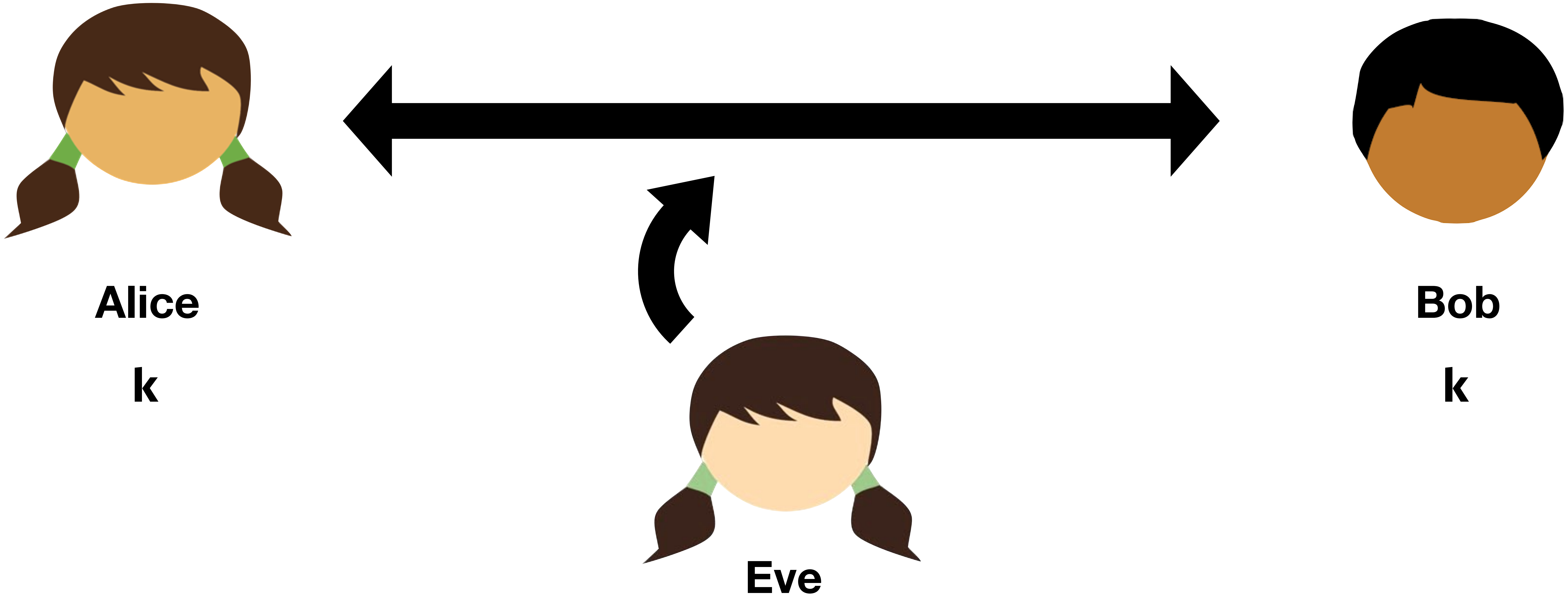
Today's objectives

Define the notion of a Message Authentication Codes (MACs)

Construct a MAC scheme for short messages (from a PRF)

Connect MACs with CCA security, via **encrypt-then-MAC**

See an example of **composing** cryptographic schemes



Alice

k

Bob

k

Eve

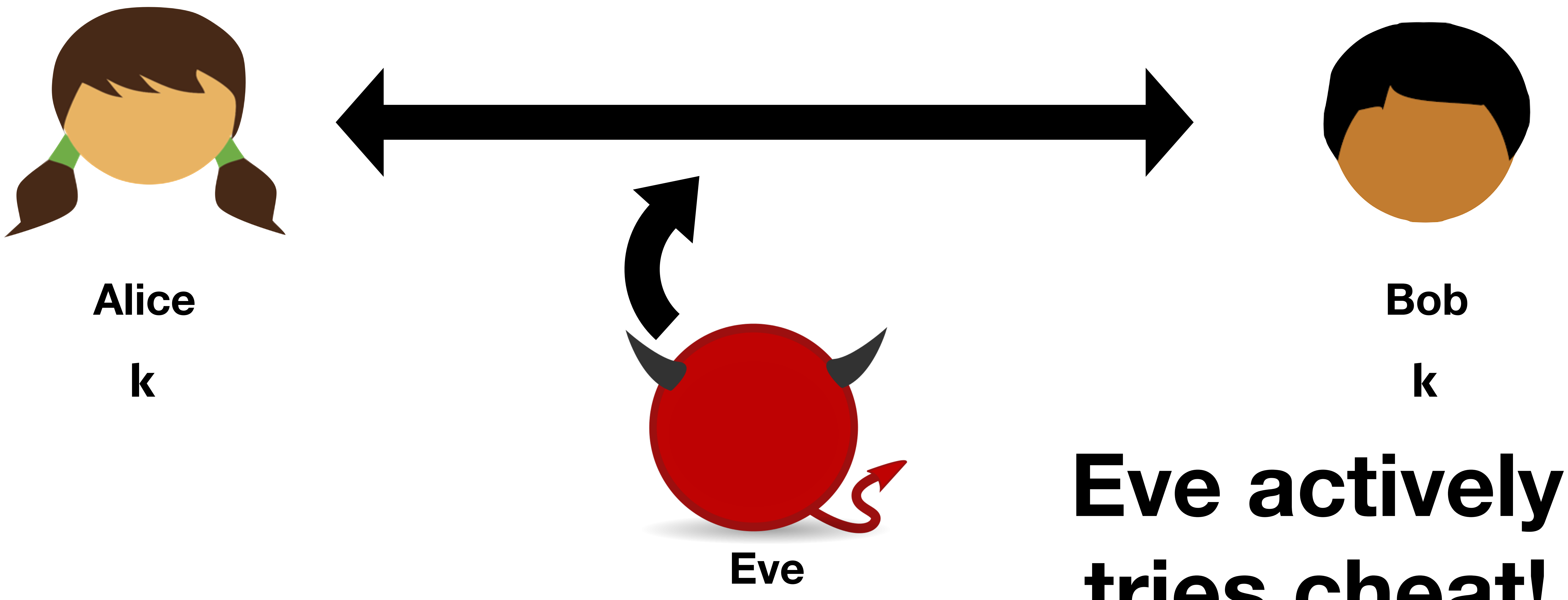
Confidentiality

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?



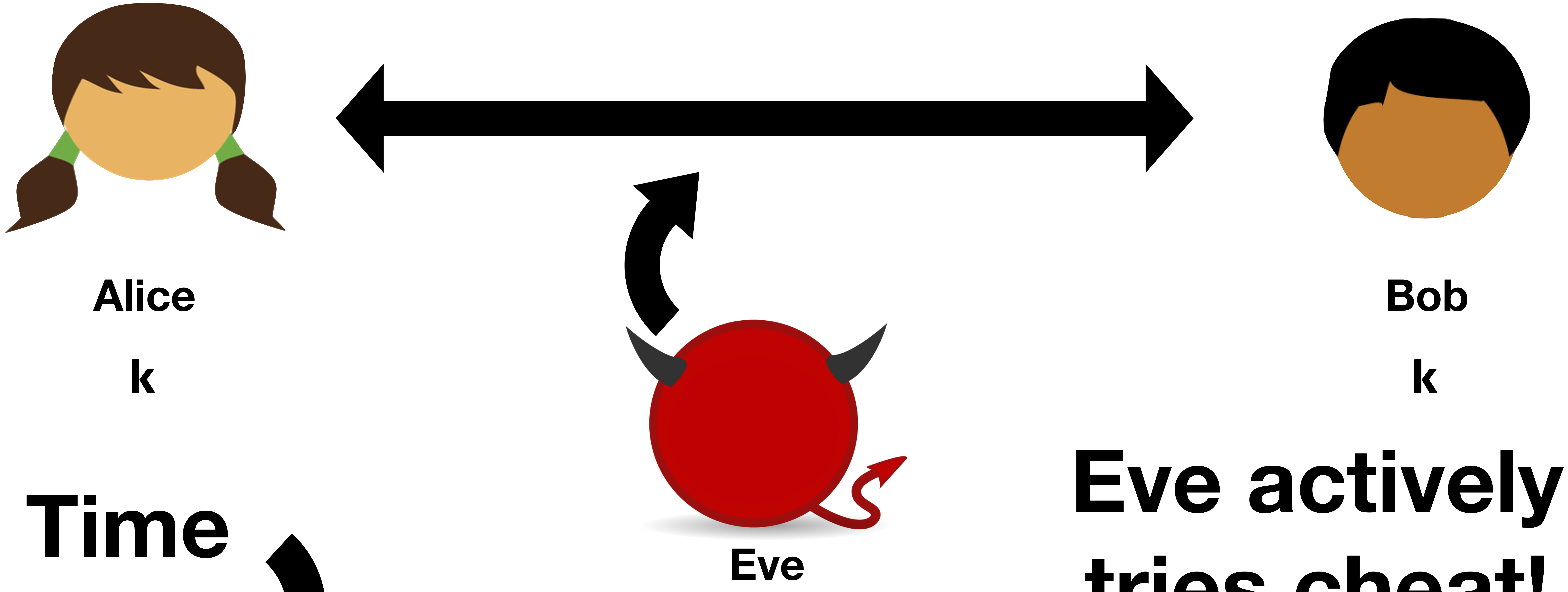
Eve actively tries cheat!

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?



Eve actively tries cheat!

Last Time

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?

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

```
k ← K

encrypt(m0, m1):
  return Enc(k, m0)
```

≈

```
k ← K

encrypt(m0, m1):
  return Enc(k, m1)
```

(messages must be of equal length)

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

```
k ← K
S ← empty-set

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

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

≈

```
k ← K
S ← empty-set

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

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

(messages must be of equal length)

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

```
k ← K
S ← empty-set

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

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

$$\approx$$

```
k ← K
S ← empty-set

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

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

If Enc/Dec are **malleable**, they will not achieve CCA security

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

```
k ← K
S ← empty-set

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

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

How (informally) can we get CCA security?

If adversary changes a ciphertext:

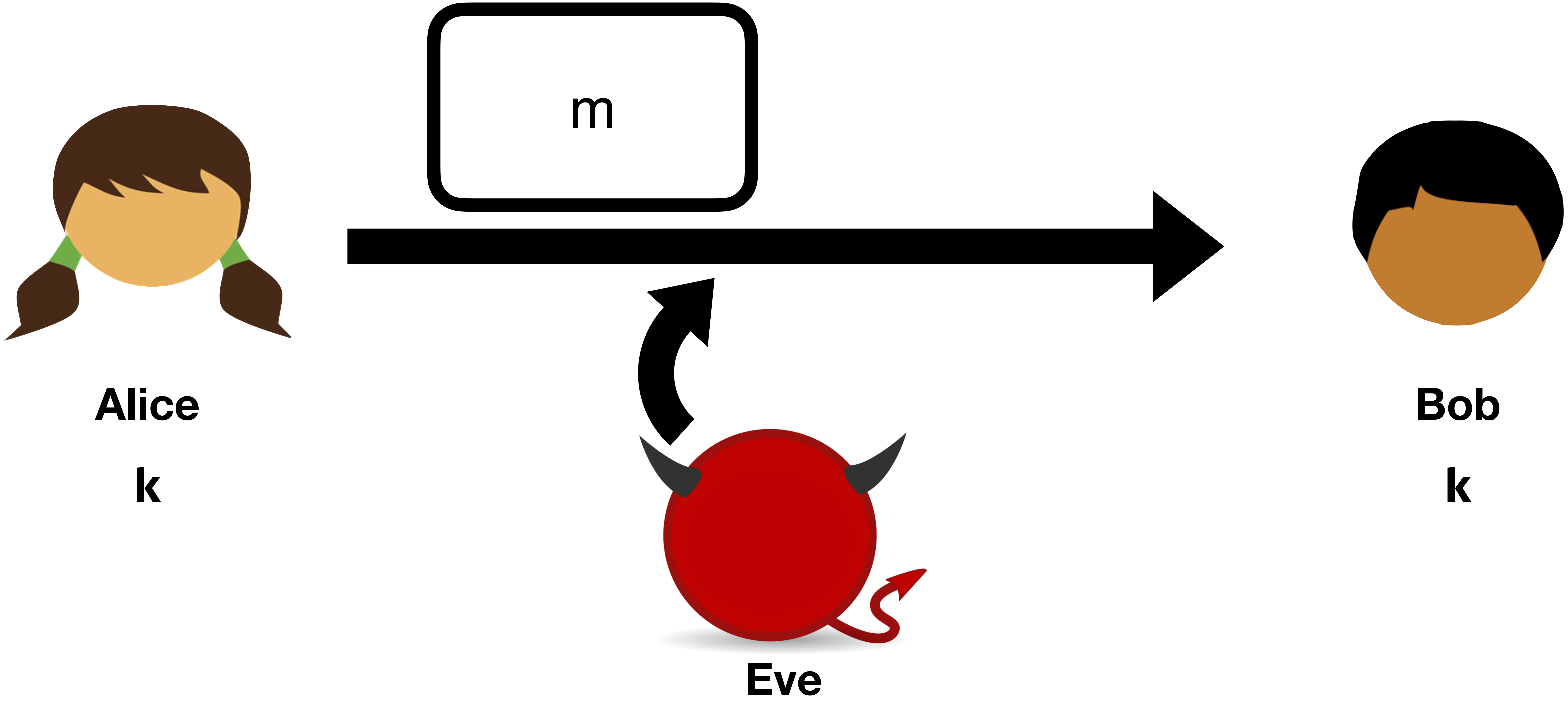
The decryption is “unrelated” to the original message

The decrypt procedure *detects* that when ciphertexts have been changed

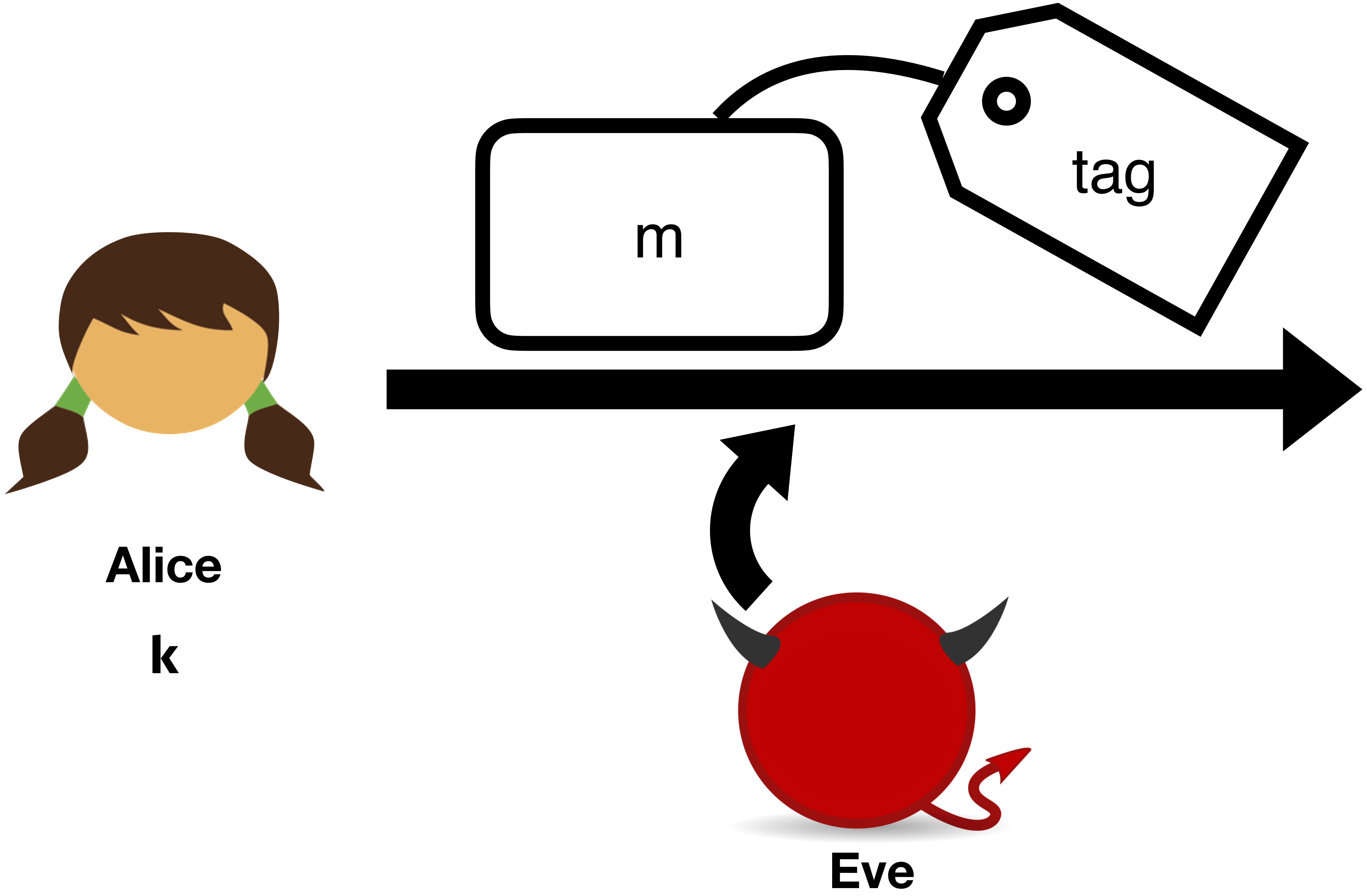
Today

If Enc/Dec are **malleable**, they will not achieve CCA security

Message Authentication Codes (MACs)



Message Authentication Codes (MACs)



"Eve cannot change m without breaking the tag"

Message Authentication Codes

A **MAC scheme** is an algorithm `tag` such that:

```
k ← K

get(m):
  return tag(k, m)

check(m, t):
  return tag(k, m) = t
```

≈

```
k ← K
S ← empty-set

get(m):
  t ← tag(k, m)
  S ← S ∪ {(m, t)}
  return t

check(m, t):
  return (m, t) ∈ S
```

$$F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$$

F is called a **pseudorandom function family** if the following indistinguishability holds:

$$\left\{ F(k, \cdot\,) \;\middle|\; k \leftarrow \{0,1\}^\lambda \right\} \approx \left\{ f \;\middle|\; f \leftarrow \text{uniform function from } \{0,1\}^\lambda \to \{0,1\}^\lambda \right\}$$

Goal:

```
k ← K

get(m):
  return tag(k, m)

check(m, t):
  return tag(k, m) = t
```

$\approx$

```
k ← K
S ← empty-set
get(m):
  t ← tag(k, m)
  S ← S ∪ {(m, t)}
  return t

check(m, t):
  return (m, t) ∈ S
```

Know:

$$\left\{ F(k,\cdot\,) \;\middle|\; k \leftarrow \{0,1\}^\lambda \right\} \approx \left\{ f \;\middle|\; f \leftarrow \text{uniform function from } \{0,1\}^\lambda \rightarrow \{0,1\}^\lambda \right\}$$

Lemma

```
f ← uniform function

eval(x):
  return f(x)

guess(x, g):
  return g = f(x)
```

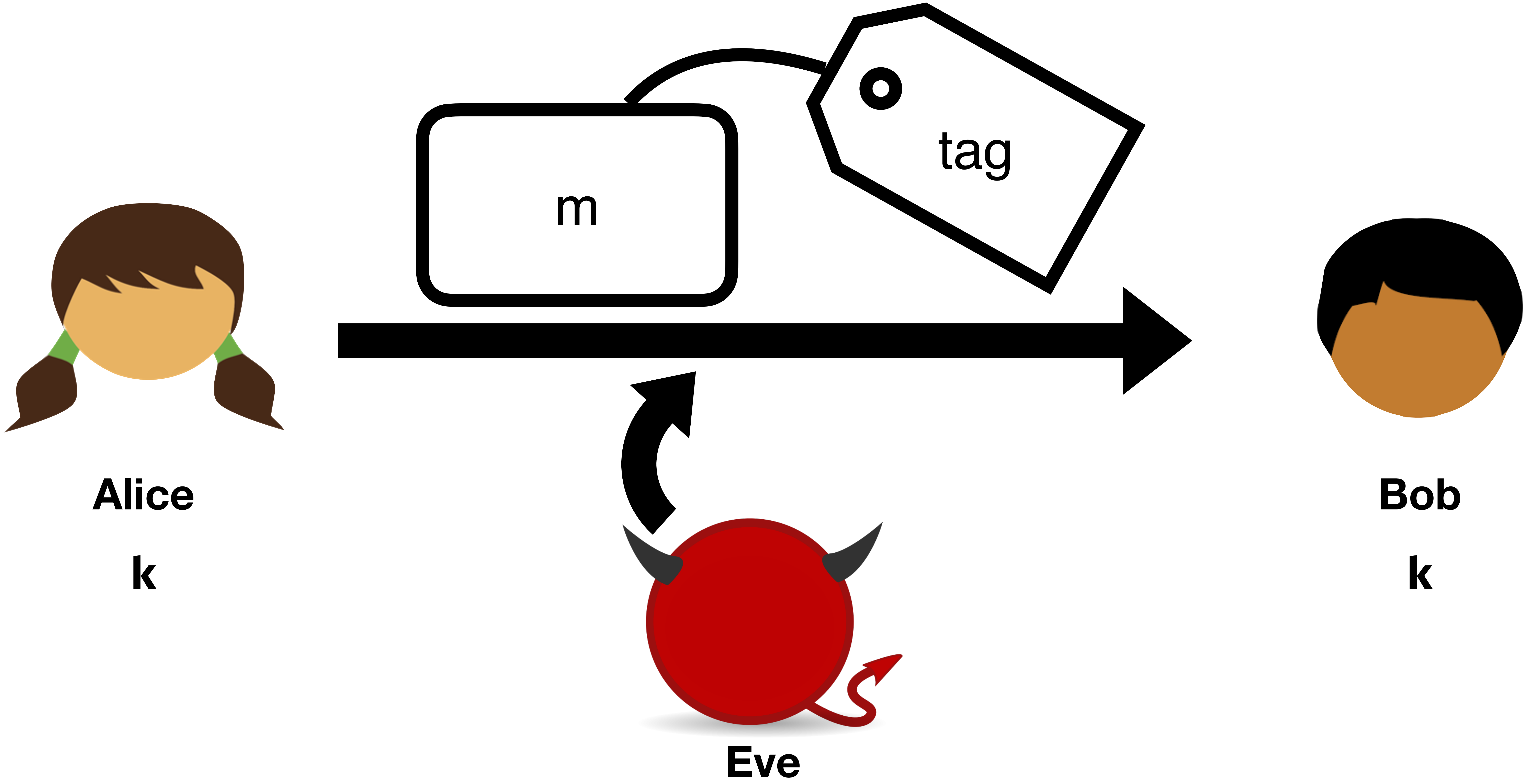
$\approx$

```
f ← uniform function
S ← empty-set

eval(x):
  S ← S ∪ {x}
  return f(x)

guess(x, g):
  if x not in S:
    return false
  return g = f(x)
```

Message Authentication Codes (MACs)



A PRF is a MAC (for short messages)

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

```
k ← K
S ← empty-set

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

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

$\approx$

```
k ← K
S ← empty-set

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

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

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

```
k ← K
S ← empty-set

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

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

≈

```
k ← K
S ← empty-set

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

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

Encrypt-then-MAC

Given CPA encryption scheme E and MAC scheme M

$K = E.K \times M.K$

$M = E.K$

$C = E.C \times M.T$

```
Enc(($k_e$, $k_m$), m):
  c ← E.Enc($k_e$, m)
  t ← M.tag($k_m$, c)
  return (c, t)
```

```
Dec(($k_e$, $k_m$), c):
  if t $\neq$ M.tag($k_m$, c)
    return error
  return E.Dec($k_e$, c)
```

Encrypt-then-MAC is CCA secure

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

```
k ← K
S ← empty-set

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

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

$$\approx$$

```
k ← K
S ← empty-set

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

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

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

```
k ← K
S ← empty-set

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

decrypt(c):
  return error
```

$$\approx$$

```
k ← K
S ← empty-set

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

decrypt(c):
  return error
```

By MAC security

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

```
k ← KeyGen()

encrypt(m0, m1):
  return Enc(k, m0)

decrypt(c):
  return error
```

$\approx$

```
k ← KeyGen()

encrypt(m0, m1):
  return Enc(k, m1)

decrypt(c):
  return error
```

By MAC security

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

```
k ← KeyGen()

encrypt(m0, m1):
  return Enc(k, m0)
```

≈

```
k ← KeyGen()

encrypt(m0, m1):
  return Enc(k, m1)
```

By MAC security

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

```
k ← KeyGen()

encrypt(m0, m1):
  return Enc(k, m0)

decrypt(c):
  return error
```

≈

```
k ← KeyGen()

encrypt(m0, m1):
  return Enc(k, m1)

decrypt(c):
  return error
```

By MAC security
By CPA security

Today’s objectives

Define the notion of a Message Authentication Codes (MACs)

Construct a MAC scheme for short messages (from a PRF)

Connect MACs with CCA security, via **encrypt-then-MAC**

See an example of **composing** cryptographic schemes