Block Ciphers and CPA Security

CS/ECE 407

Today’s objectives

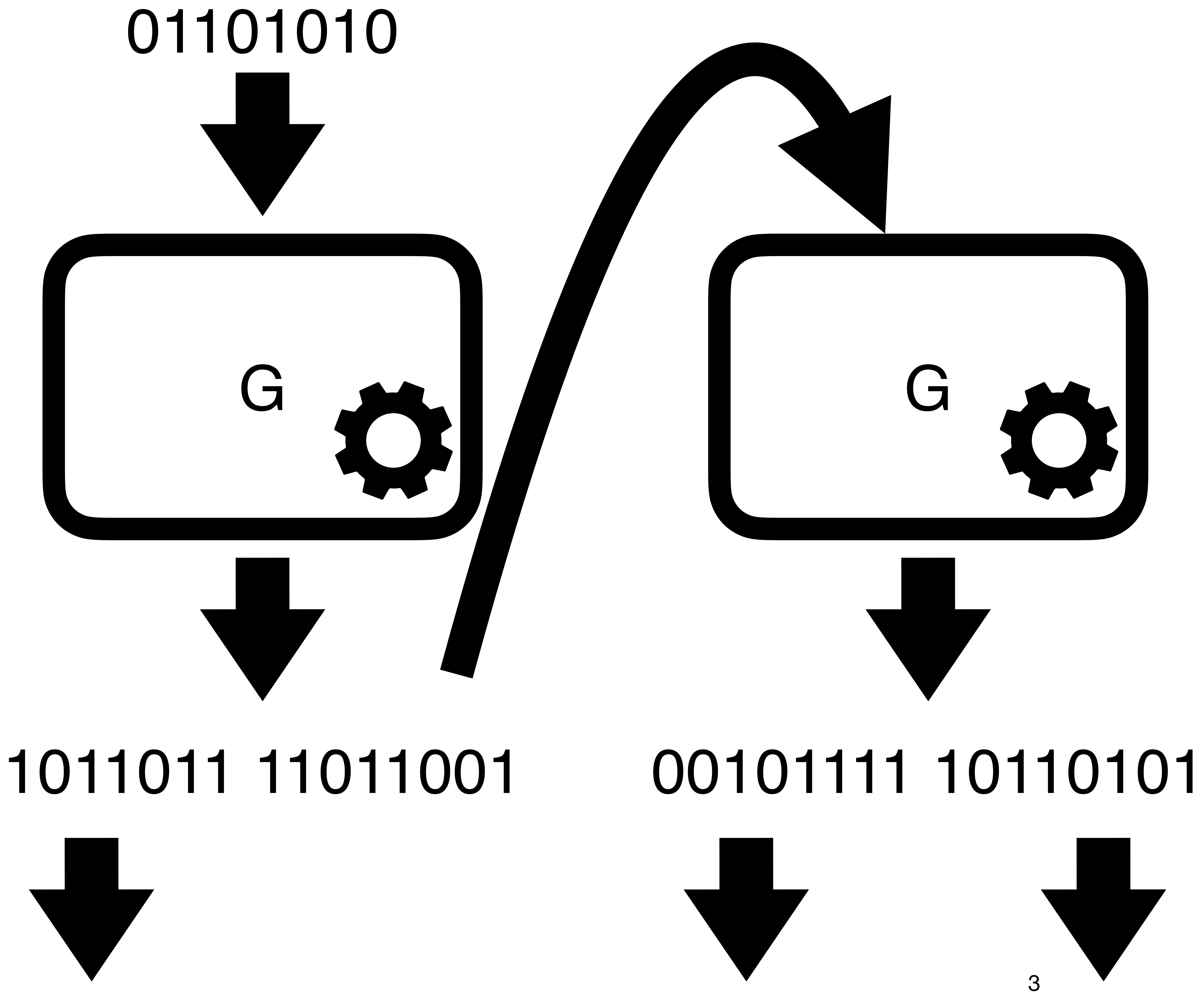
Define block ciphers

See the Advanced Encryption Standard cipher

Define CPA Security

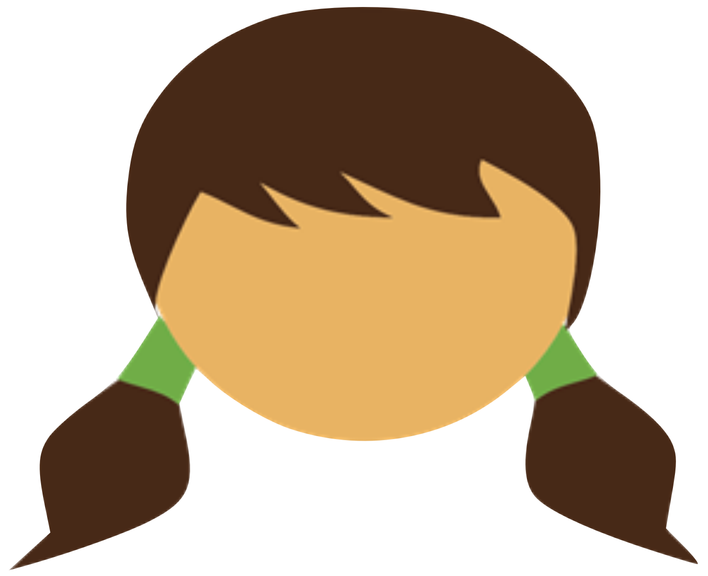
Understand the limitations of deterministic encryption

Stretching the output of a PRG



Call multiple times to get more randomness

What about a cryptographic primitive that generates a lot of randomness "all at once"



Alice

Bob

2^n **rows**

0	01101000
1	11110000
2	10001110
3	01010100
4	11011010
…	…

A pseudorandom function (PRF) allows Alice and Bob to share a huge pseudorandom table via a short key

$$F : \{0,1\}^\lambda \times \{0,1\}^n \rightarrow \{0,1\}^m$$

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

```
k ←$ {0,1}^λ

apply(x):
  return F(k, x)
```

$\overset{c}{\approx}$

```
D ← empty-dictionary

apply(x):
  if x is not in D:
    D[x] ←$ {0,1}^m
  return D[x]
```

Given a PRF, build a PRG

“Straightforward”, homework problem

PRG → PRF

PRG ← PRF

Given a PRG, build a PRF

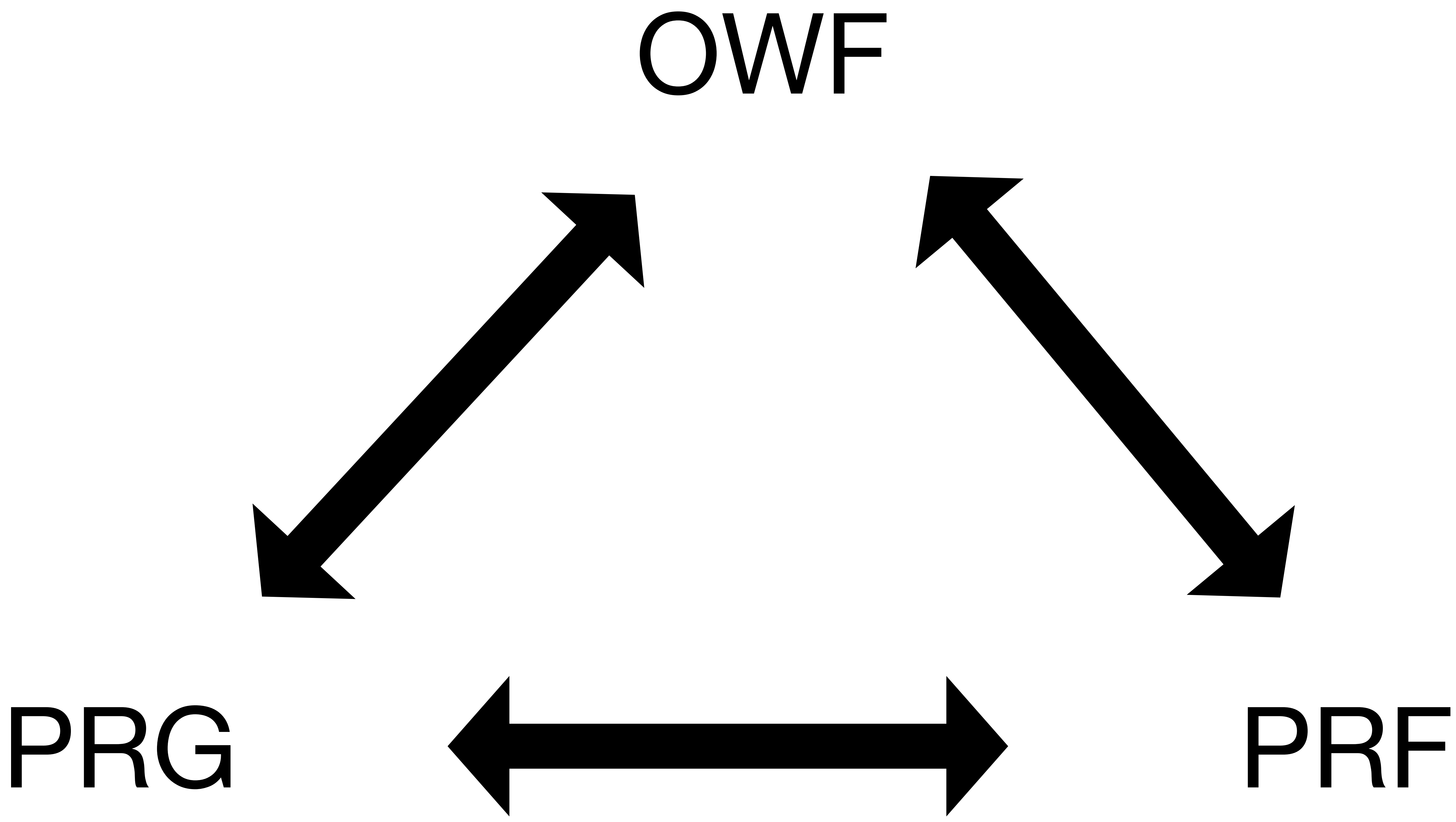
Goldreich-Goldwasser-Micali construction

$$f : \{0,1\}^n \to \{0,1\}^m$$

f is called a **one-way function** if for any poly-time program A and for all inputs x the following probability is negligible (in n):

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\, f(A(f(x)) = f(x) \,\right]$$

"f is hard to invert"



OWFs exist $\implies P \neq NP$

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

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

```
k ←$ {0,1}^λ

apply(x):
  return F(k, x)
```

$$\overset{c}{\approx}$$

```
D ← empty-dictionary

apply(x):
  if x is not in D:
    D[x] ←$ {0,1}^m
  return D[x]
```

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

F is called a **pseudorandom permutation (or block cipher)** if:

There exists F^{-1} s.t. $F^{-1}(k, F(k, x)) = x$

```
k ←$ {0,1}^λ

apply(x):
  return F(k, x)
```

$\overset{c}{\approx}$

```
D ← empty-dictionary

apply(x):
  if x is not in D:
    D[x] ←$ {0,1}^λ \ D.keys
  return D[x]
```

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

F is called a **pseudorandom permutation (or block cipher)** if:

There exists F^{-1} s.t. $F^{-1}(k, F(k,x)) = x$

Dictionary entries are all distinct

```
k ←$ {0,1}^λ

apply(x):
  return F(k, x)
```

$\overset{c}{\approx}$

```
D ← empty-dictionary

apply(x):
  if x is not in D:
    D[x] ←$ {0,1}^λ \ D.keys
  return D[x]
```

A cipher (Enc, Dec) has **one-time semantic security** if:

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m0)
  return ct
```

$$\overset{c}{\approx}$$

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m1)
  return ct
```

How can we implement (Enc, Dec) with a block cipher?

Advanced Encryption Standard (AES)

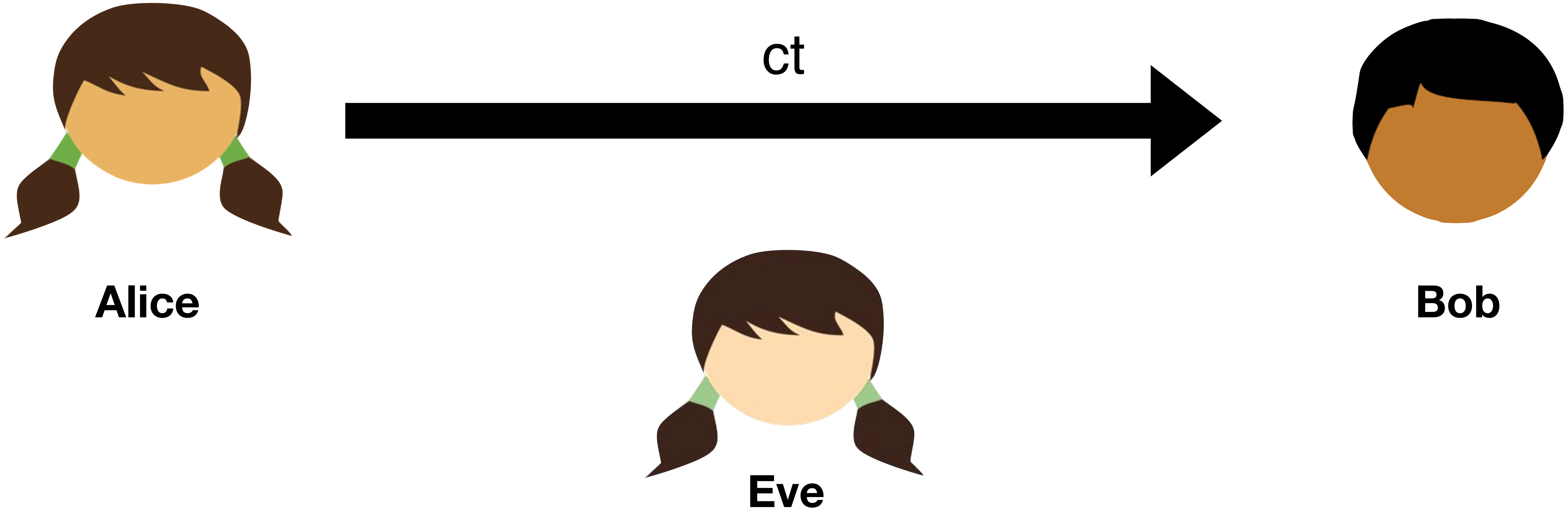
A cipher (Enc, Dec) has **one-time semantic security** if:

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m0)
  return ct
```

$\overset{c}{\approx}$

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m1)
  return ct
```

What if Alice wants to send more than one message?

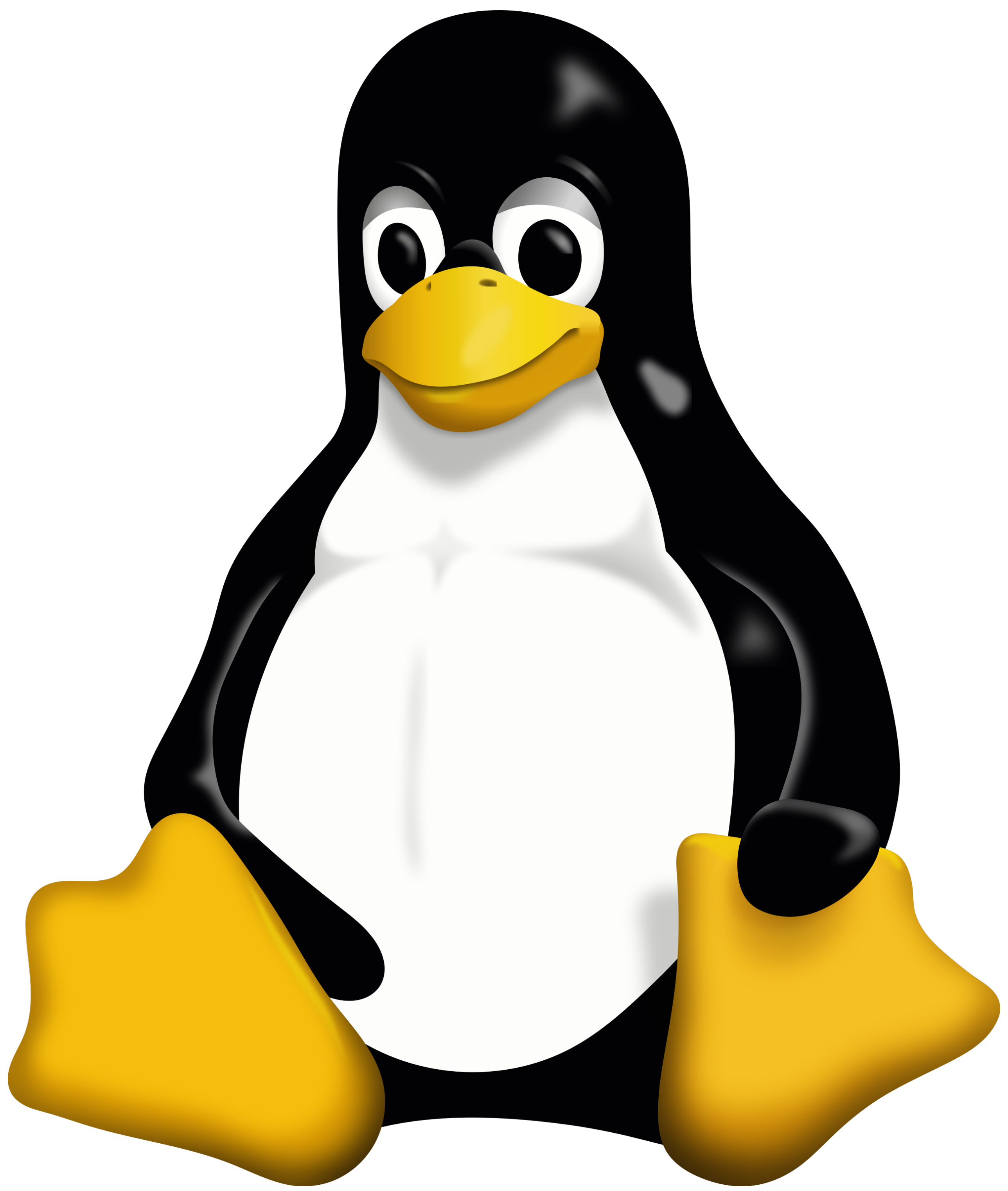


A cipher (Enc, Dec) has
one-time semantic security if:

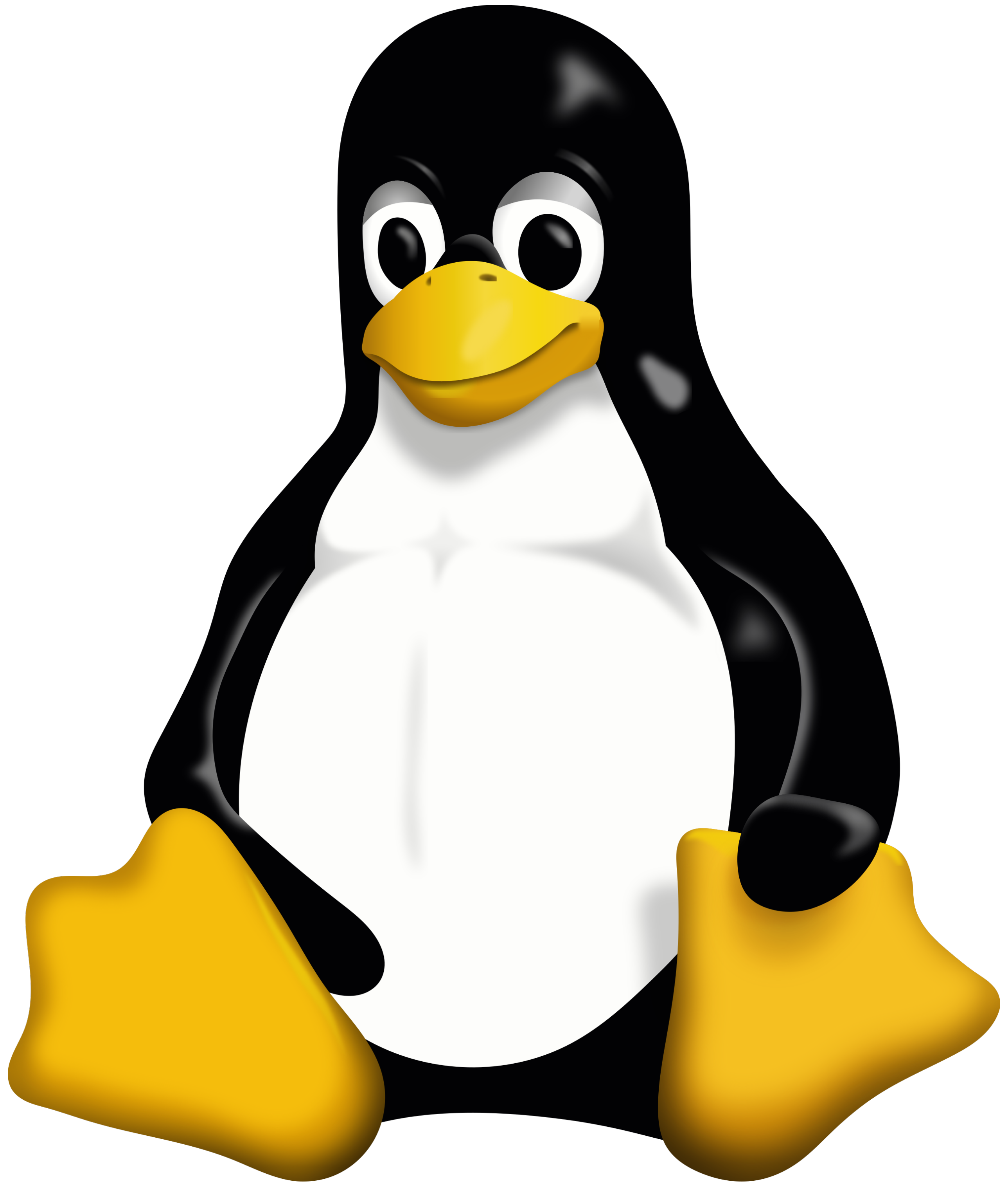
```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m0)
  return ct
```

$\overset{c}{\approx}$

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m1)
  return ct
```

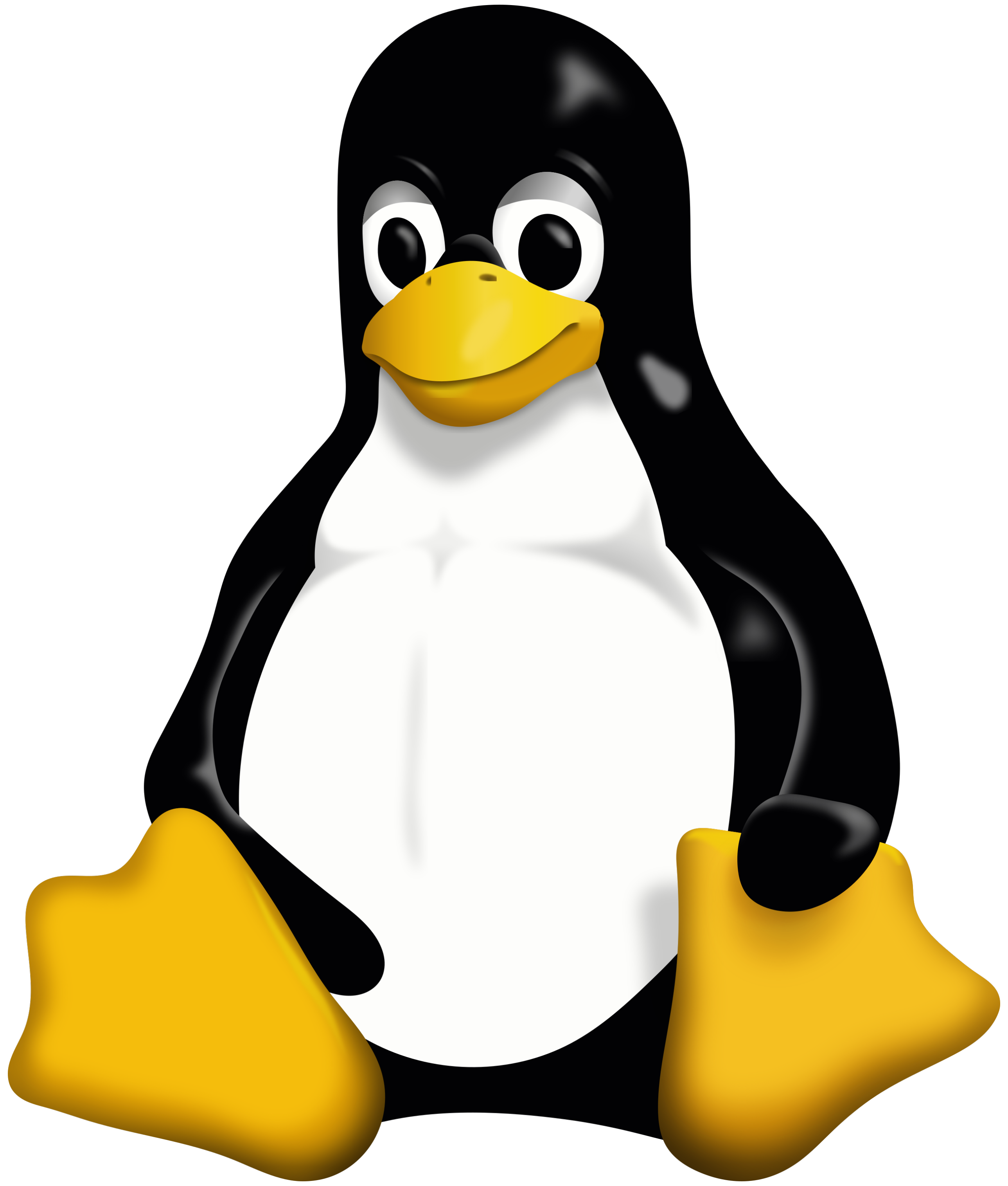


“Tux”



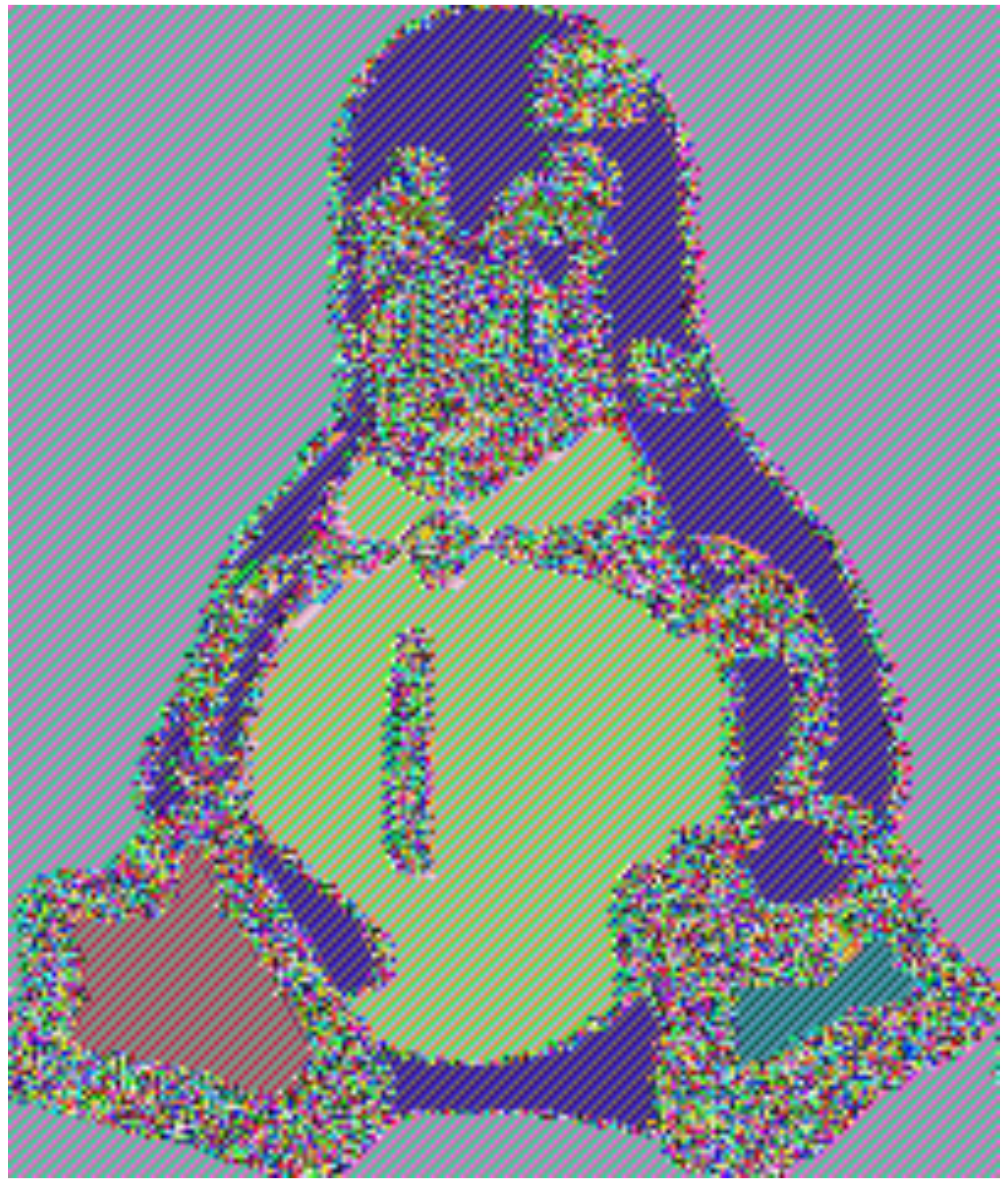
“Tux”

“Good” encryption



“Tux”

“Good” encryption



Naively re-using key

A cipher (Enc, Dec) has **one-time semantic security** if:

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m0)
  return ct
```

$\overset{c}{\approx}$

```
eavesdrop(m0, m1):
  k ←$ {0,1}^λ
  ct ← Enc(k, m1)
  return ct
```

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

```
k ←$ {0,1}^λ
eavesdrop(m0, m1):
  ct ← Enc(k, m0)
  return ct
```

$\overset{c}{\approx}$

```
k ←$ {0,1}^λ
eavesdrop(m0, m1):
  ct ← Enc(k, m1)
  return ct
```

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

```
k ←$ {0,1}^λ
eavesdrop(m0, m1):
  ct ← Enc(k, m0)
  return ct
```

$$\overset{c}{\approx}$$

```
k ←$ {0,1}^λ
eavesdrop(m0, m1):
  ct ← Enc(k, m1)
  return ct
```

Deterministic encryption can never achieve CPA security!

Today's objectives

Define block ciphers

See the Advanced Encryption Standard cipher

Define CPA Security

Understand the limitations of deterministic encryption