Digital Signatures

CS/ECE 407

Today's objectives

Introduce notion of digital signature

Construct an *identification scheme*

See instance of Fiat Shamir transform

Construct a digital signature scheme

Public Key Encryption

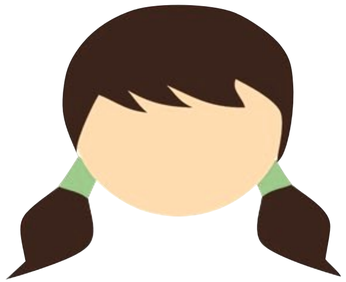
Alice

sk

Enc(pk, m)

Bob

pk



???

Eve

pk

Key Exchange

g^a

g^b

Alice

Bob

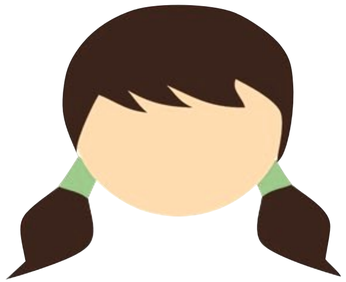
$a \xleftarrow{\$} \mathbb{Z}_q$

g^{ab}

$b \xleftarrow{\$} \mathbb{Z}_q$

g^{ab}

???



Eve

g^a

g^c

g^d

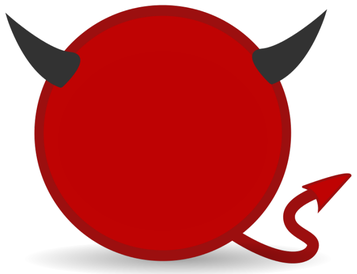
g^b

Alice

Bob

$a \xleftarrow{\$} \mathbb{Z}_q$

$b \xleftarrow{\$} \mathbb{Z}_q$



g^{ad}

$c \xleftarrow{\$} \mathbb{Z}_q$

g^{bc}

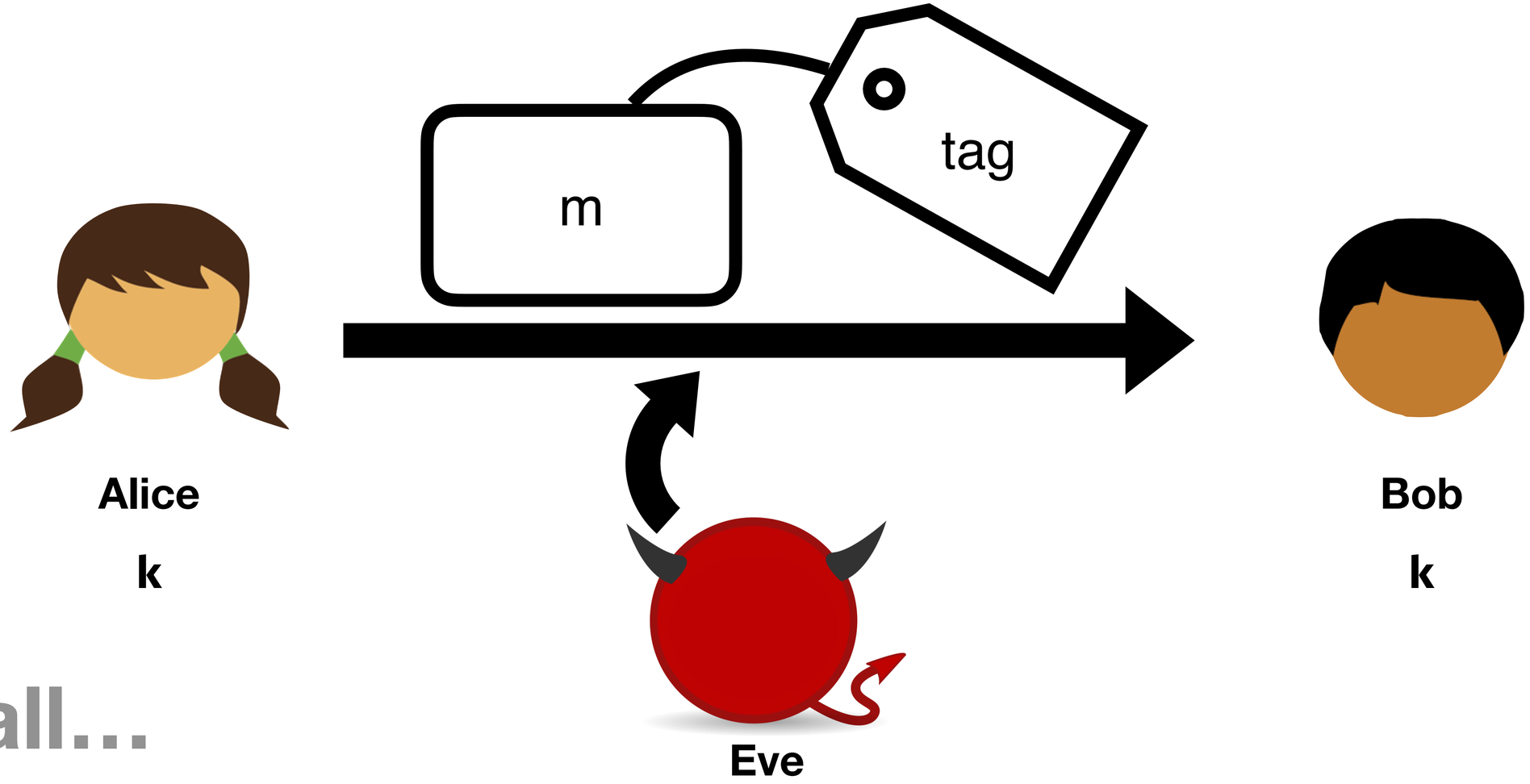
$d \xleftarrow{\$} \mathbb{Z}_q$

g^{ad}, g^{bc}

Eve

To prevent this attack, we will need some kind of public-key authentication

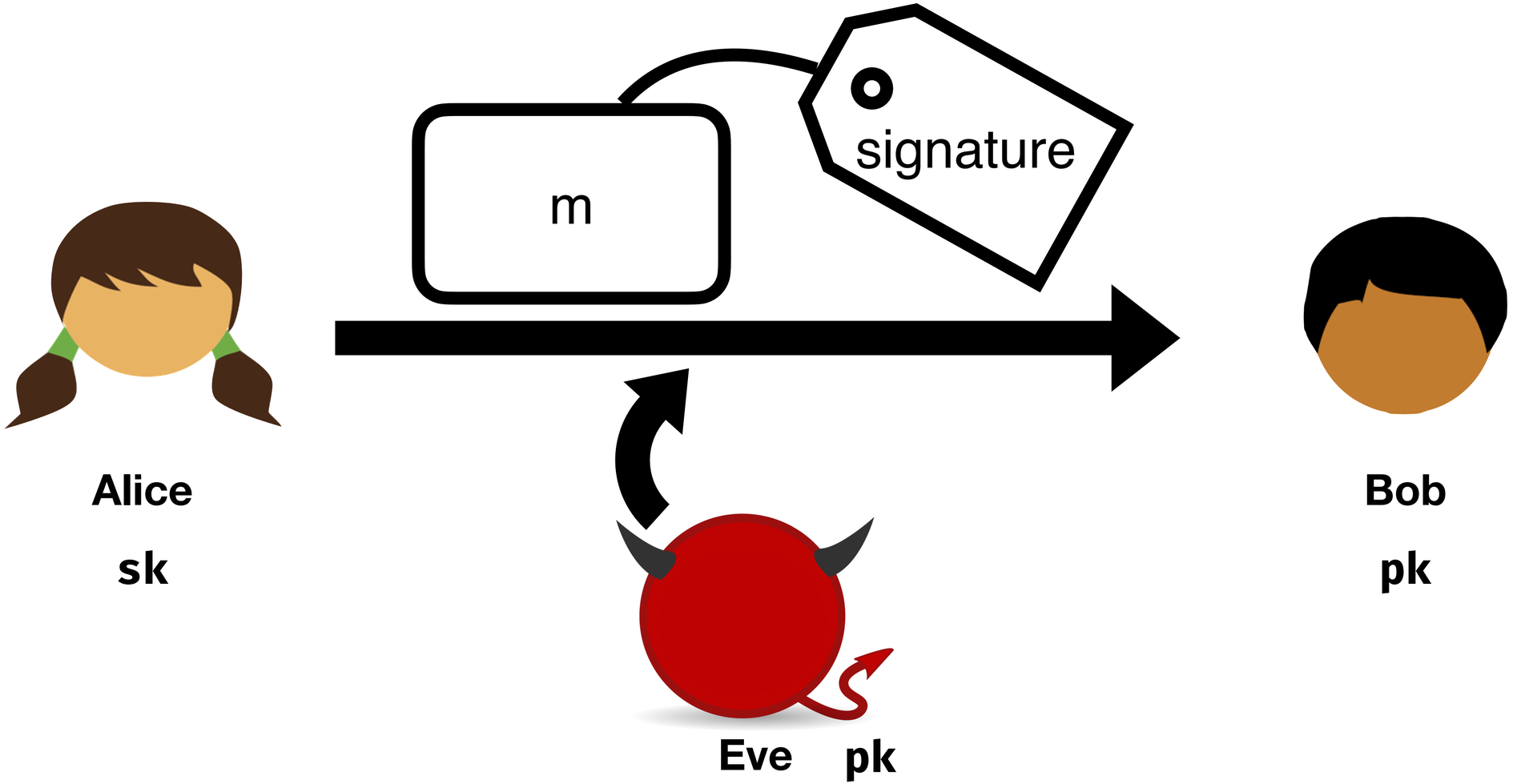
Message Authentication Codes (MACs)



Recall…

“Eve cannot change m without breaking the tag”

Digital Signatures



“Eve cannot change m without breaking the signature”

Digital Signatures

`Gen()` outputs a **key pair** `(pk, sk)`

`Sign(sk, m)` outputs a **signature** σ

`Verify(pk, m, σ)` outputs `{0,1}`

Correctness:

Digital Signatures

`Gen()` outputs a **key pair** `(pk, sk)`

`Sign(sk, m)` outputs a **signature** σ

`Verify(pk, m, σ)` outputs `{0,1}`

Correctness: `(pk, sk) ← Gen()`

`Verify(m, Sign(sk, m), pk) = 1`

“Public key MACs”

Existential Unforgeability under Chosen Message Attack (EUF-CMA)

```
(pk, sk) ← KeyGen()

key(): return pk

get(m):
  return sign(sk, m)

check(m, σ):
  return verify(pk, m, σ)
```

≈

```
(pk, sk) ← KeyGen()
S ← empty-set

key(): return pk

get(m):
  σ ← sign(sk, m)
  S ← S ∪ {(m, σ)}
  return σ

check(m, σ):
  return (m, σ) ∈ S
```

Discrete Logarithm Assumptions

Let $\mathbb{G}_\lambda, g_\lambda$ be a family of cyclic groups and generators

We say the discrete logarithm is hard for that family if for all polytime adversaries A the following probability is negligible:

$$\Pr\left[A(1^\lambda, g_\lambda^x) = x \;\middle|\; \begin{array}{l} q = |\mathbb{G}_\lambda| \\ x \leftarrow \mathbb{Z}_q \end{array}\right]$$

Identification Scheme

Alice

$sk \leftarrow Z_q$

$r \leftarrow Z_q$

g^r

$c \leftarrow Z_q$

$s = r + sk \cdot c$

Bob

$pk = g^{sk}$

$g^r \cdot pk^c \stackrel{?}{=} g^{r+sk \cdot c}$

Identification Scheme

Alice

sk

Alice and Bob interact

Bob becomes convinced Alice knows sk

Bob

$\mathrm{pk} = g^{\mathrm{sk}}$

Identification Scheme

Alice

sk

Alice and Bob interact

Bob becomes convinced Alice knows sk

Bob learns nothing about sk

Bob

$pk = g^{sk}$

Schnorr Signatures

```
KeyGen():
  sk ←Zq
  return (sk, g^sk)
```

q prime, **g** generator

```
sign(sk, m):
  r ← Zq
  c ← H(g^r || m)
  s ← r + sk·c
  return (g^r, s)
```

```
verify(g^sk, m, (g^r, s)):
  c ← H(g^r || m)
  return g^s = g^r·(g^sk)^c
```

If discrete log assumption holds and if H is a random oracle, Schnorr scheme is EUF-CMA secure

Today's objectives

Introduce man-in-the-middle attack

Introduce notion of digital signature

Construct an *identification scheme*

See instance of Fiat Shamir transform

Construct a digital signature scheme