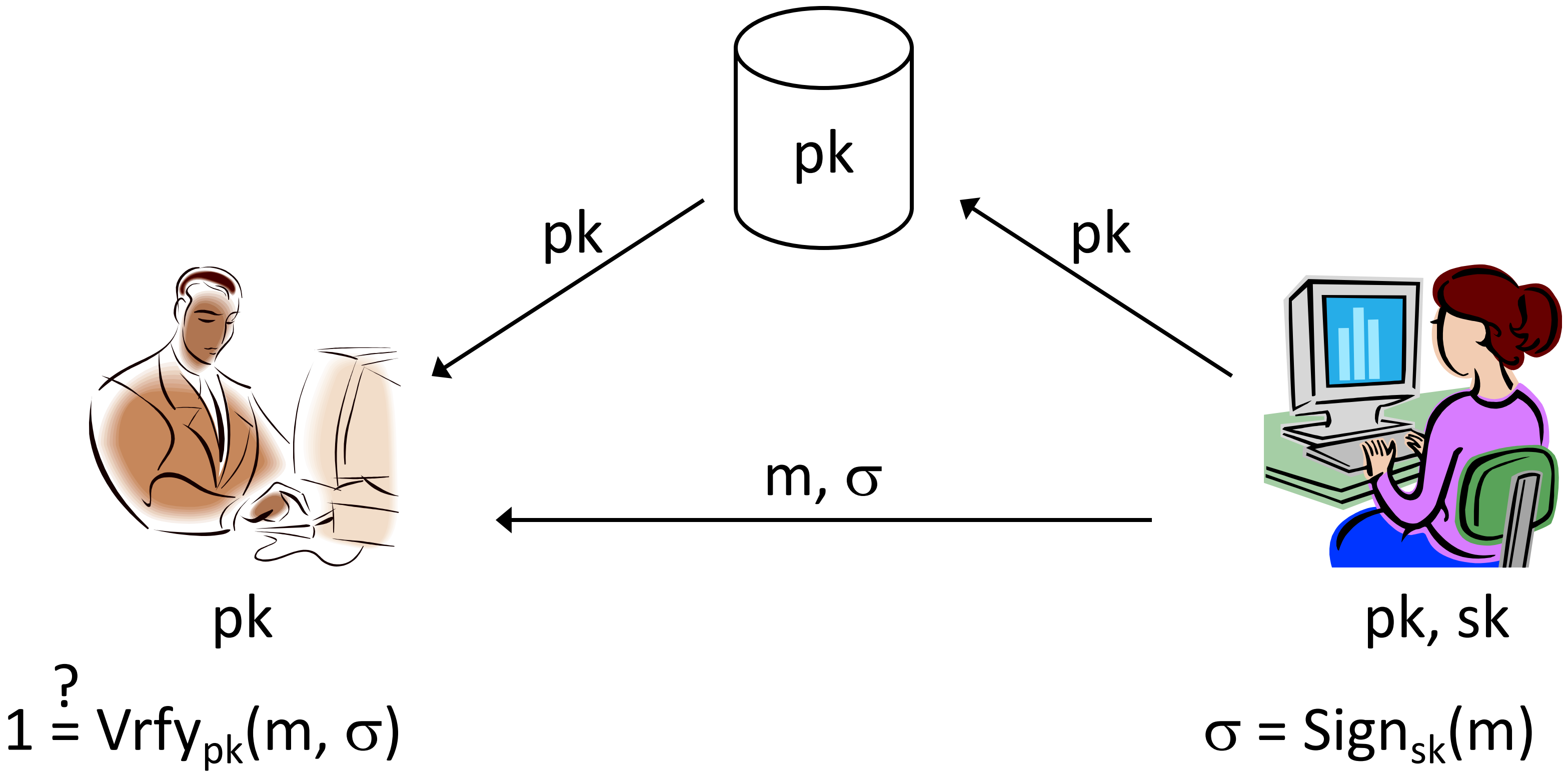
Defining Digital Signatures

Slides by Prof. Jonathan Katz.
Lightly edited by me.

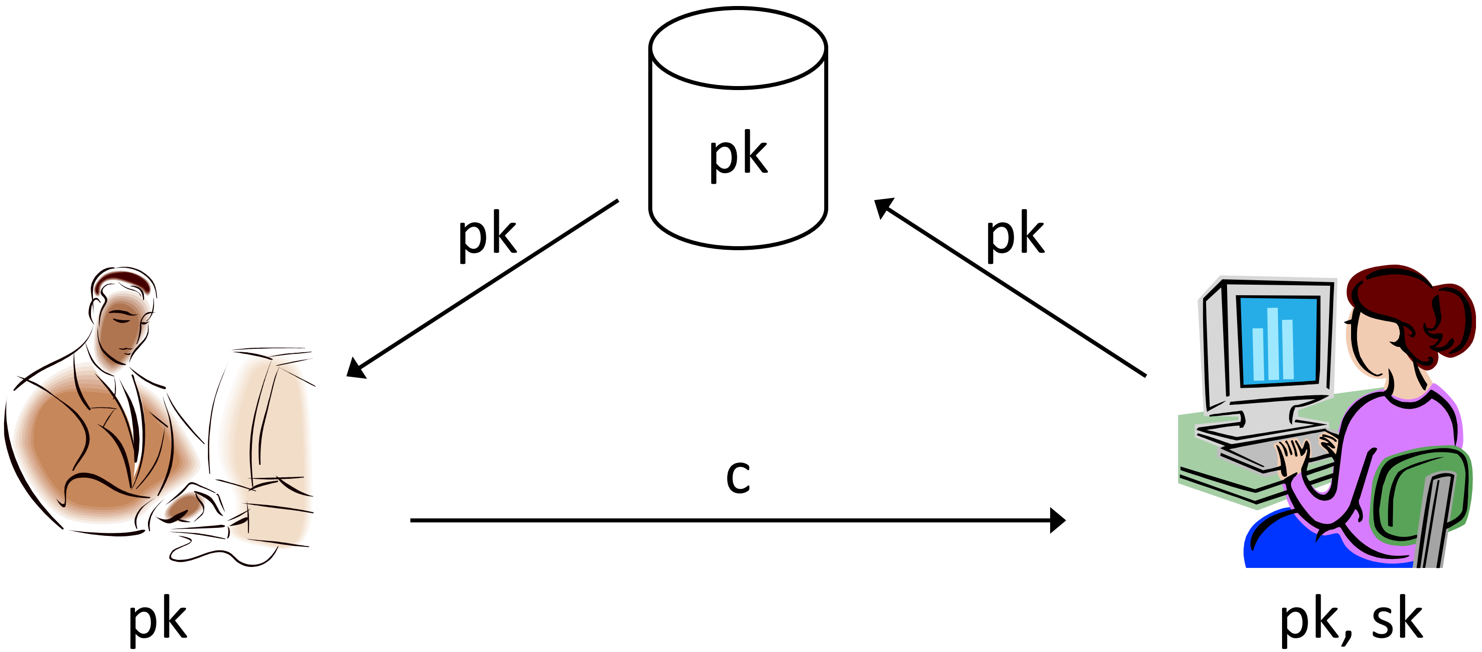
Digital signatures

- Provide *integrity* in the public-key setting
- Analogous to message authentication codes, but some key differences…

Digital signatures



Public-key encryption



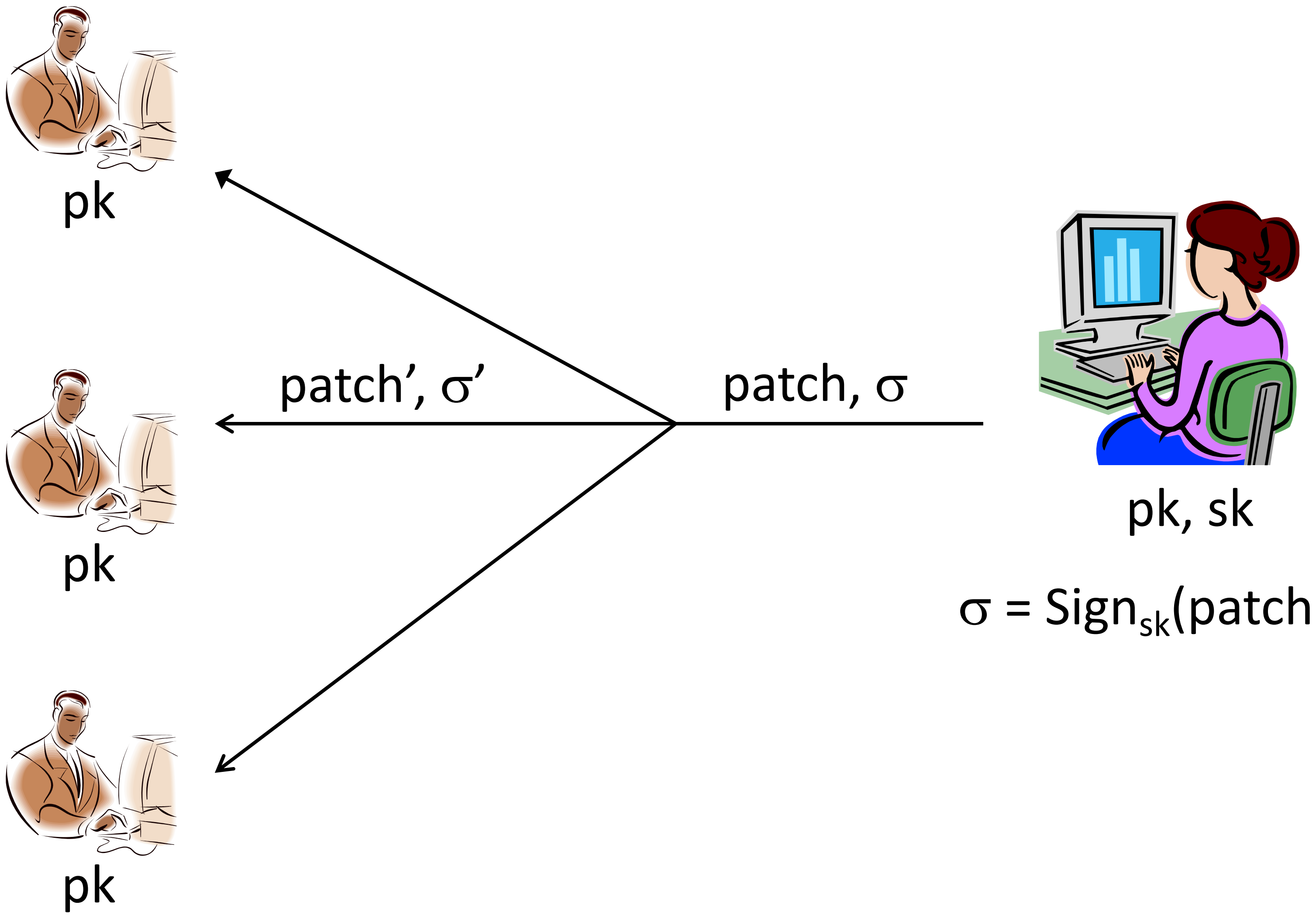
$c \leftarrow Enc_{pk}(m)$

$m = Dec_{sk}(c)$

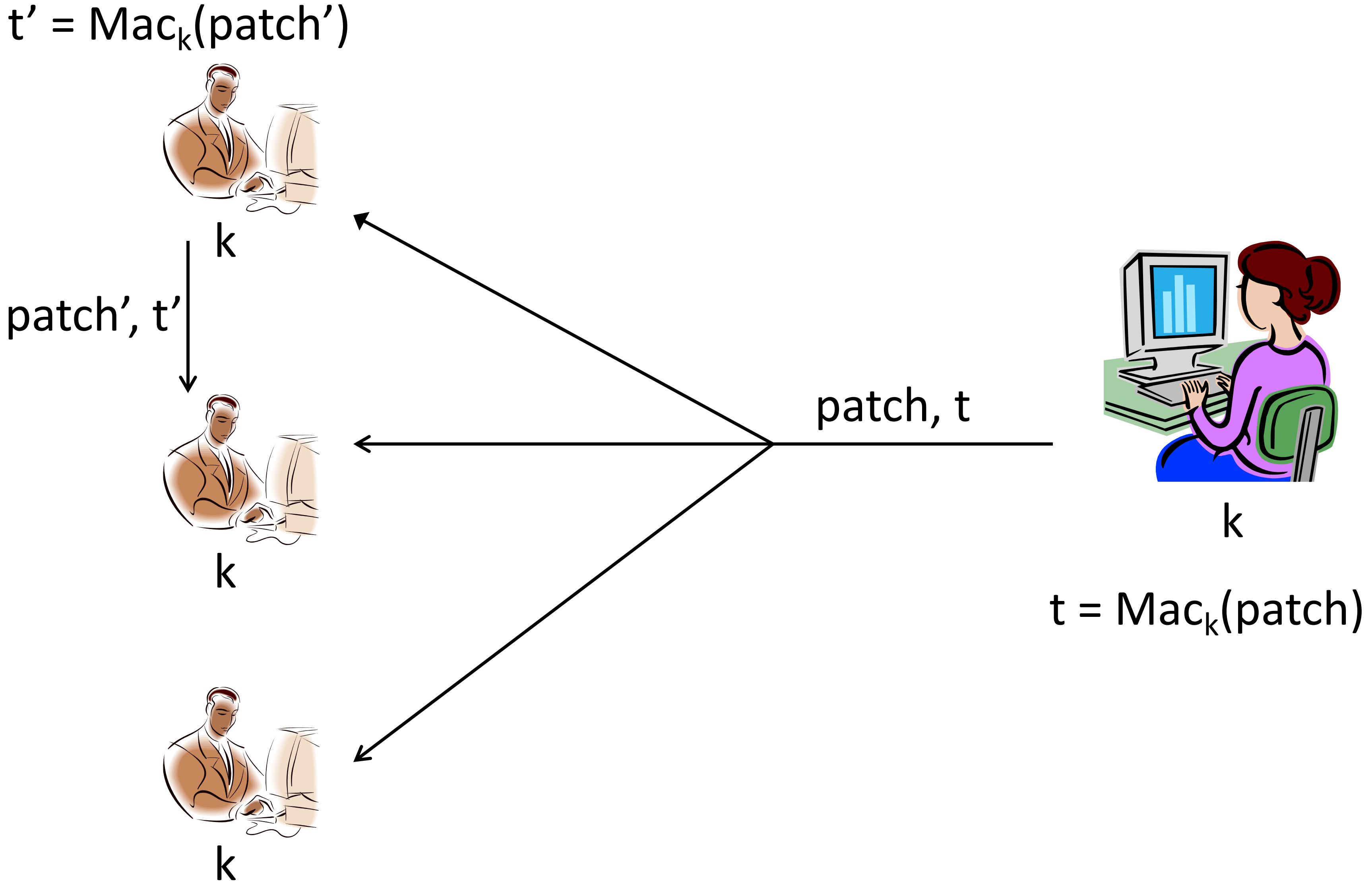
Security (informal)

- Even after observing signatures on multiple messages, an attacker should be unable to *forge* a valid signature on a *new* message

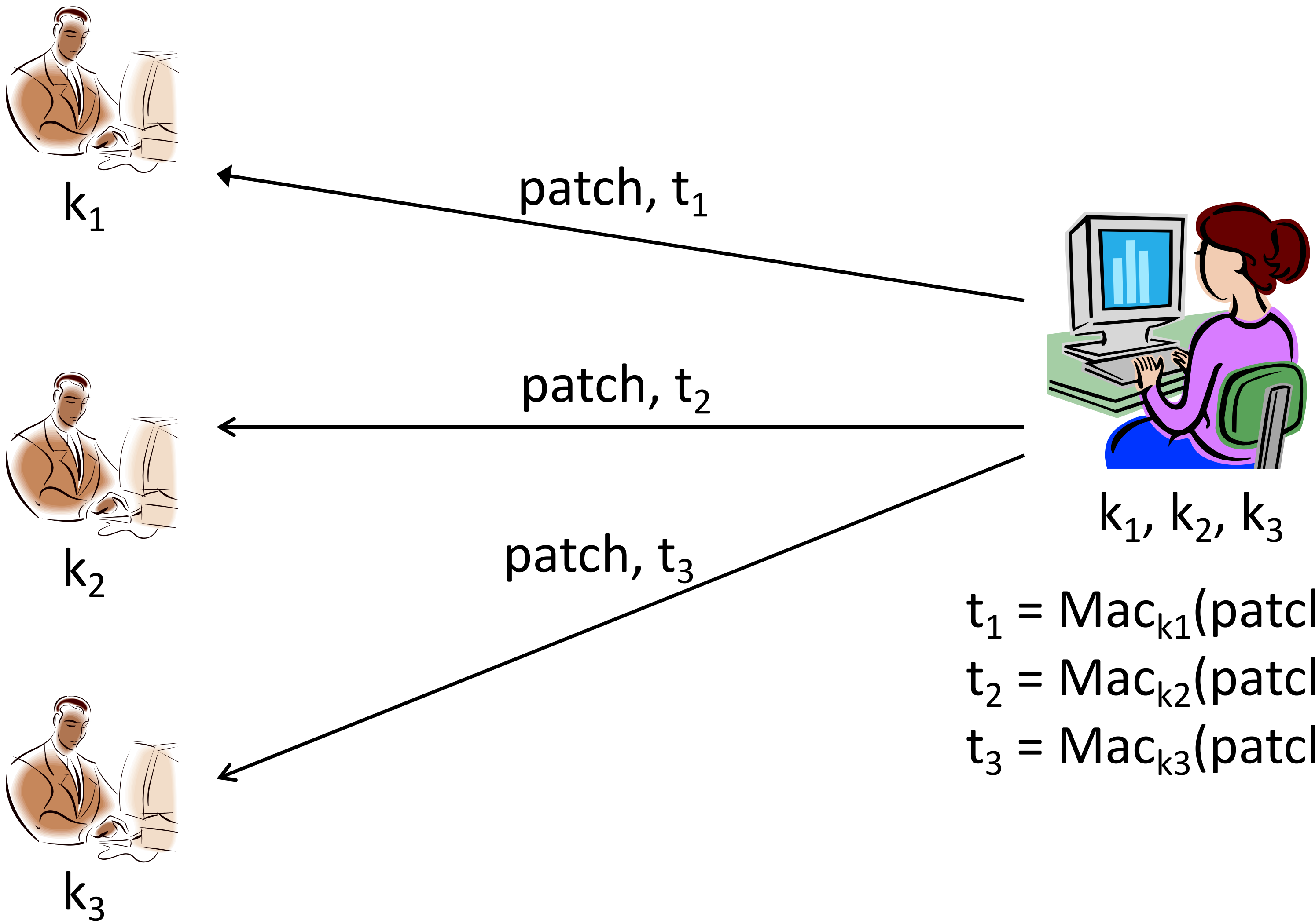
Prototypical application



Comparison to MACs?



Comparison to MACs?



$t_1 = \mathrm{Mac}_{k1}(\mathrm{patch})$

$t_2 = \mathrm{Mac}_{k2}(\mathrm{patch})$

$t_3 = \mathrm{Mac}_{k3}(\mathrm{patch})$

Comparison to MACs?

- *Public verifiability*
  - “Anyone” can verify a signature
  - (Only a holder of the key can verify a MAC tag)

$\Rightarrow$ *Transferability*
  - Can forward a signature to someone else…

$\Rightarrow$ *Non-repudiation*

Non-repudiation

- Signer cannot (easily) deny issuing a signature
  - Crucial for legal applications
  - Judge can verify signature using public copy of pk
- MACs cannot provide this functionality!
  - Without access to the key, no way to verify a tag
  - Even if receiver gives key to judge, how can the judge verify that the key is correct?
    - Even if key is correct, receiver could have generated the tag also!

Signature schemes

- A *signature scheme* is defined by three PPT algorithms (Gen, Sign, Vrfy):
  - Gen: takes as input 1^n; outputs pk, sk
  - Sign: takes as input a private key sk and a message $m \in \{0,1\}^*$; outputs signature σ

$$\sigma \leftarrow \text{Sign}_{sk}(m)$$

  - Vrfy: takes public key pk, message m, and signature σ as input; outputs 1 or 0

For all m and all pk, sk output by Gen,
$\text{Vrfy}_{pk}(m, \text{Sign}_{sk}(m)) = 1$

Security?

- Threat model
  - "Adaptive chosen-message attack"
  - Assume the attacker can induce the sender to sign *messages of the attacker's choice*
- Security goal
  - "Existential unforgeability"
  - Attacker should be unable to forge valid signature on *any* message not signed by the sender
- Attacker gets the public key…

Security for signature schemes

- Π is *secure* if for all PPT attackers A, there is a negligible function ε such that

$$\Pr[\mathrm{Forge}_{A,\Pi}(n) = 1] \leq \varepsilon(n)$$

Replay attacks

- Replay attacks need to be addressed just as in the symmetric-key setting