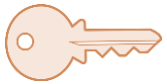
Digital Signatures

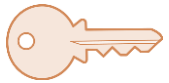


Verify m, σ
using vk

σ, m

Verification key

Signing key



Sign m as
σ using sk

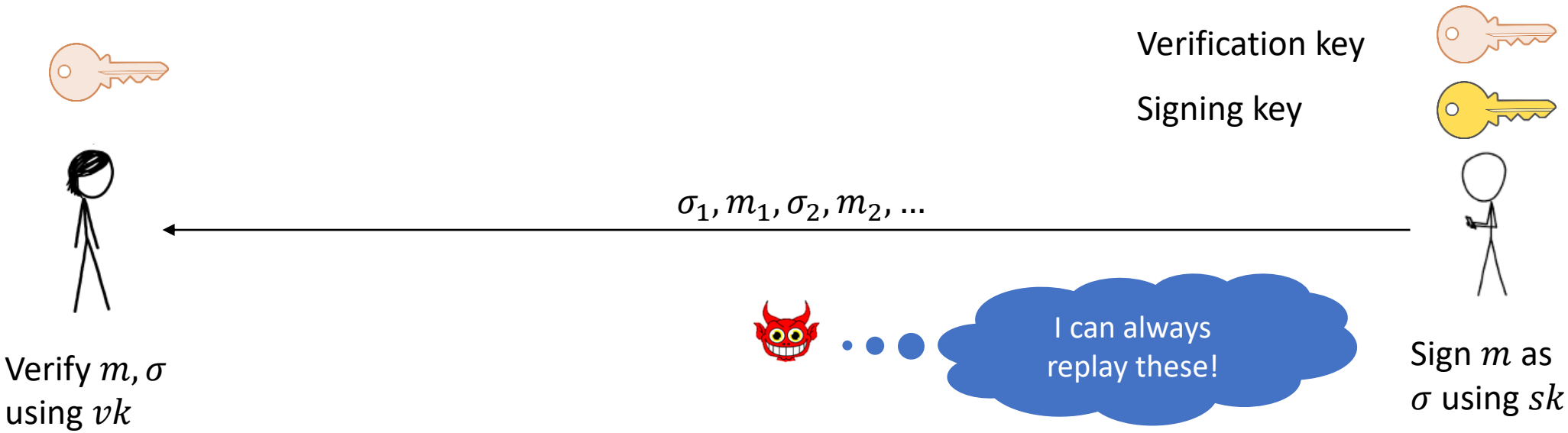
Algorithms:
$(vk, sk) \leftarrow KG$
$\sigma \leftarrow Sign(m, sk)$
$\{\bot, \top\} \leftarrow Verify(m, \sigma, vk)$

Use cases of digital signatures

- ``digital'' equivalent of signing a contract (NemID/MitID)
- Building authenticated channels over insecure network
- Software integrity
- Transactions in cryptocurrencies

Defining Security

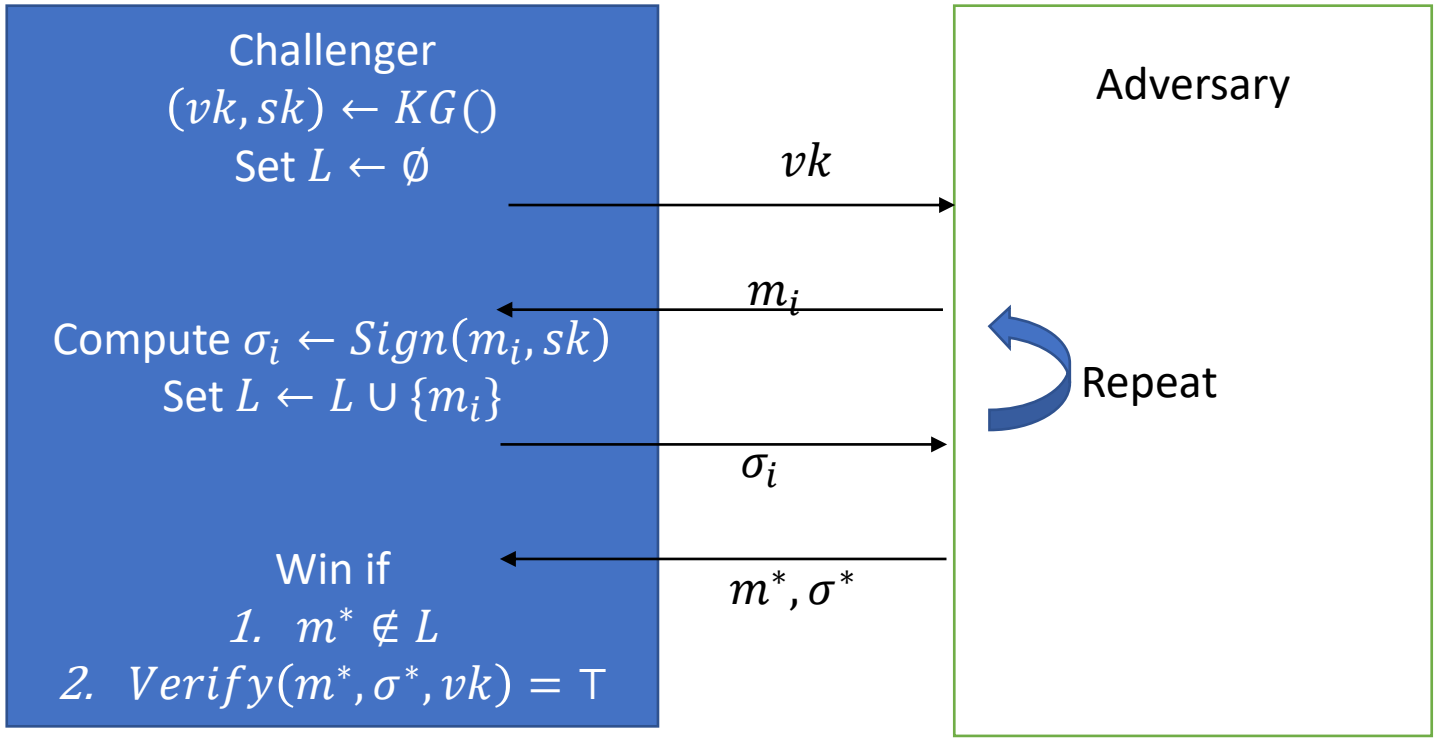
MACs for public key setting!



Unforgeability:

No adversary with vk and message/signature pairs $m_1, \sigma_1, \ldots$ should be able to make new m, σ

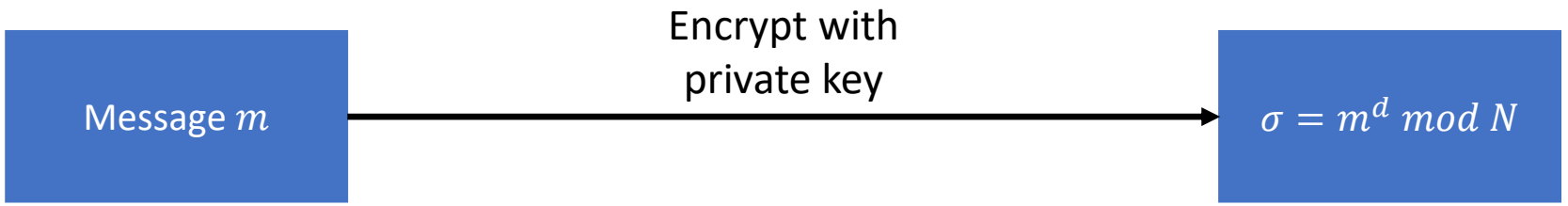
EUF-CMA for Signatures



Signatures from RSA: the wrong way

Signing key: secret d

Verification key: N, e



Verify m, σ:

Check that $m = \sigma^e \ mod \ N$

This clearly fails!

Counterexample 1

Generate signature on ``random'' message:

1. Let $\mathrm{pk} = (N, e)$
2. Fix a random element $\sigma \in Z_N^*$
3. Compute $m = \sigma^e \ mod \ N$

(m, σ) is valid by construction

Counterexample 2 – Inspired by Homework 2

We want to forge a signature on m

1. Choose $m_1 \in Z_N^*$, compute $m_2 \leftarrow \frac{m}{m_1} mod\ N$

2. Ask EUF-CMA oracle to compute $\sigma_1 \leftarrow Sign(m_1, sk), \sigma_2 \leftarrow Sign(m_2, sk)$

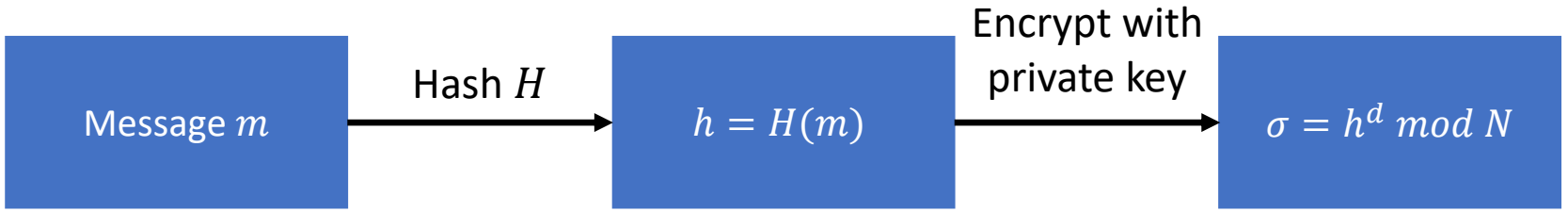
3. Then $\sigma = \sigma_1 \cdot \sigma_2 = m_1^d \cdot m_2^d = m^d$ is a valid signature on m!

Digital Signatures using RSA: RSA-FDH

Signing key: secret d

Verification key N, e

Cryptographic hash $H: \{0,1\}^* \rightarrow Z_N^*$



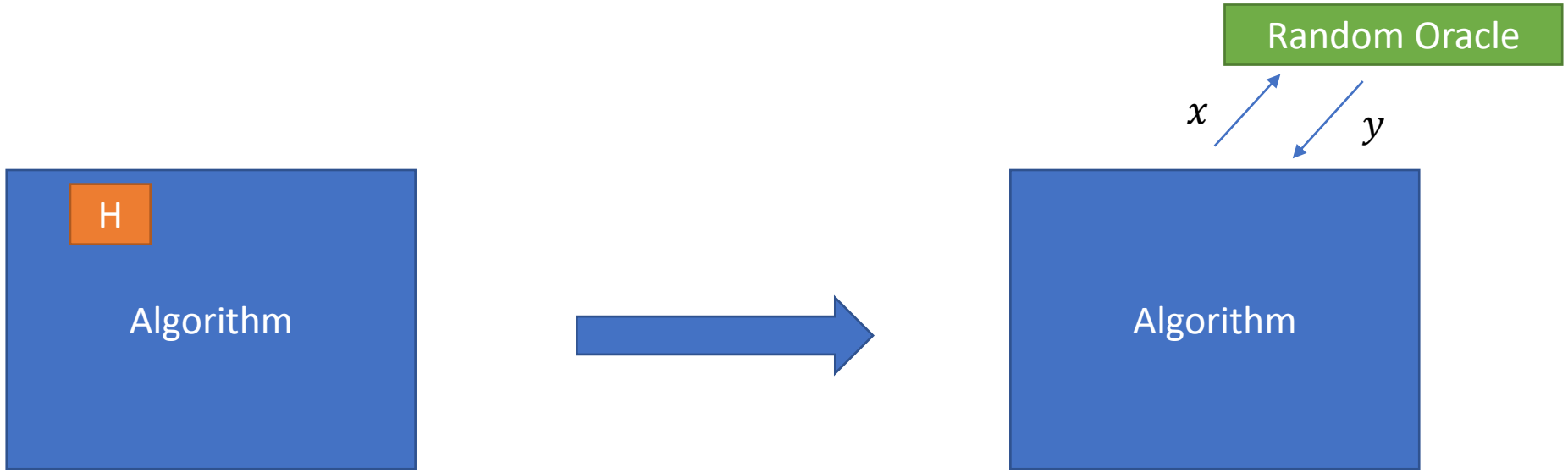
Verify m, σ:

Check that $H(m) = \sigma^e \ mod \ N$

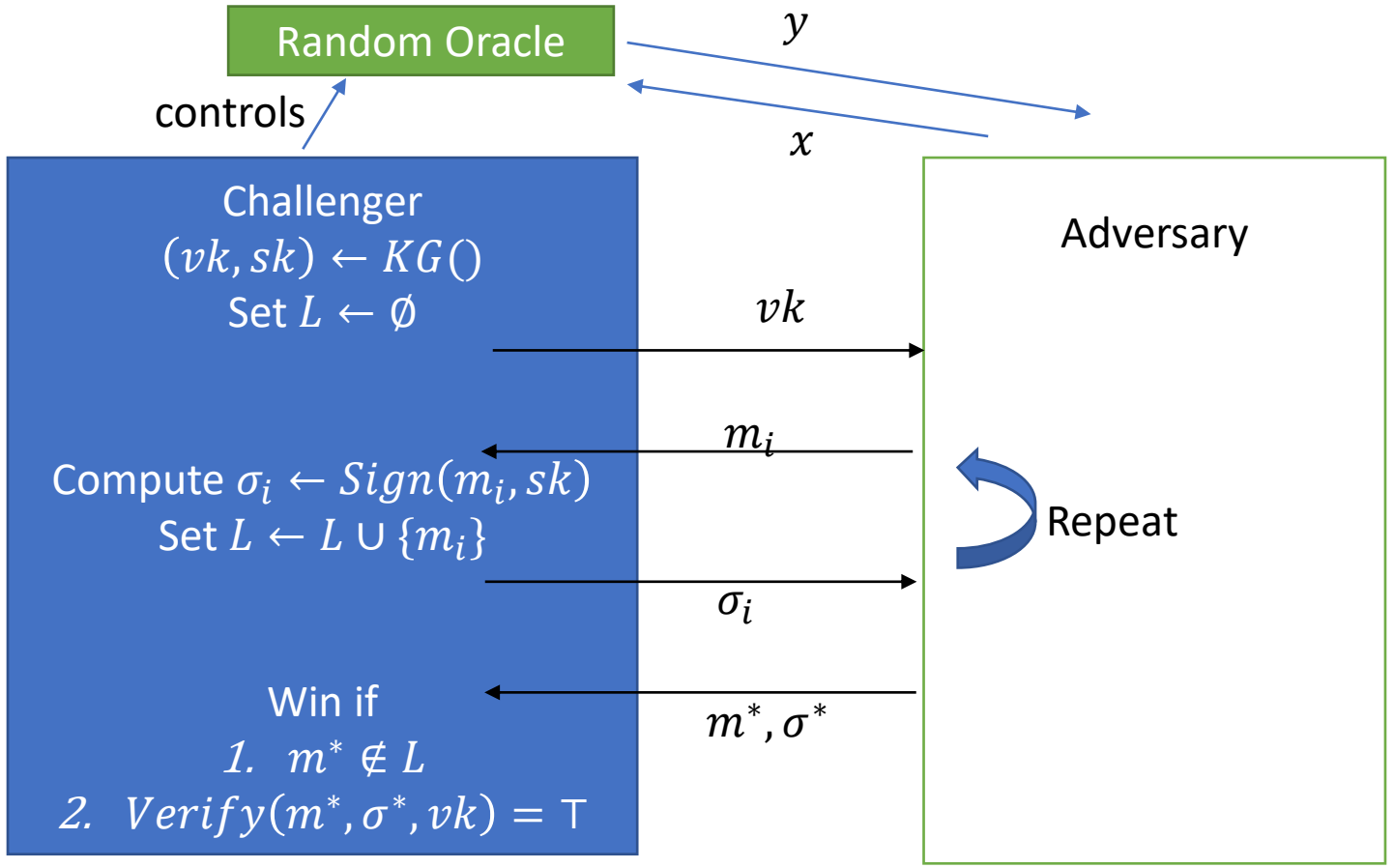
Any RSA instance for encryption can also be used for signing!

EUF-CMA security

Recap from Problem Sheet 5: the Random Oracle Model



Looking at EUF-CMA



What we prove

Assuming H is a random oracle. Then given the RSA problem is hard (Problem Sheet 6), RSA-FDH is EUF-CMA secure.