650.445: Practical Cryptographic Systems

Provable Security II

Review

- Housekeeping:
  - Readings: two new papers on Syllabus
  - For 3/23 (3/25)
  - Midterm on 3/25
  - A1

-Mean: 85, Median 90, Stdev ~15

-Have a great spring break

Review

- Last time:
  - Intro to Provable Security
  - Information-Theoretic vs. Complexity-Theoretic
  - One-Way Functions

-Implications: P = NP?

  - Schnorr vs. DSA
  - Random Oracles (!!)

Today:

- Reduction Proofs
  - Exploring the concept
  - Specific examples
  - How Random Oracles help

Reduction Proofs

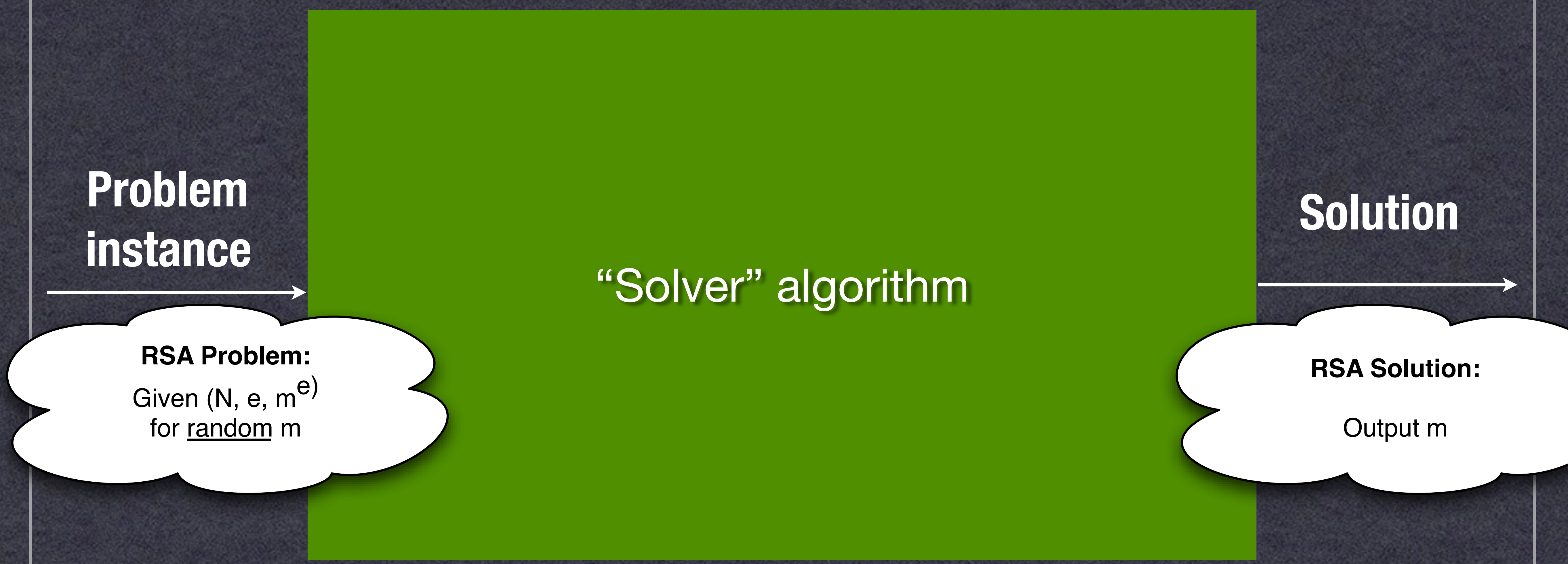
- The basic idea:
  - I assume that problem X is <u>hard</u>
  - And demonstrate that:

-If there <u>exists</u> an adversary (program) that "breaks" my scheme

-Then I could use this adversary
(as a subroutine) to solve problem X

  - By contradiction: <u>the adversary cannot exist!</u>

Example: RSA Problem



Example: RSA Assumption

Problem instance

RSA Problem:
Given (N, e, m^e)
for random m

Hypothesis:
No efficient (polynomial time) algorithm solves this problem with greater than negligible probability.

Solution

RSA Solution:
Output m

Theorem

- **Statement:**
  - If the RSA assumption holds,
    then it's hard to decrypt an RSA ciphertext

Example

- **Statement:**
  - ~~**If the RSA assumption holds, then it's hard to decrypt an RSA ciphertext**~~
  - **If the RSA assumption holds, there is no (efficient) algorithm that, given pk = (N, e) and random ciphertext m^e, outputs m (except with negligible probability)**

Precisely states what we want to prove. But how do we prove this?

Example

- **Statement:**
  - If the RSA assumption holds,
    there is no (efficient) algorithm that, given $pk = (N, e)$ and ciphertext m^e, outputs m (except with negligible probability).

Example

- **Statement:**
  - **If the RSA assumption holds, there is no (efficient) algorithm that, given pk = (N, e) and ciphertext m^e, outputs m (except with negligible probability).**

Not quite sure how to prove this...

1st Try

Example

1st Try

- **Statement:**
  - If the RSA assumption holds,
    there is no (efficient) algorithm that, given
    pk = (N, e) and ciphertext m^e, outputs m
    (except with negligible probability).

Example

2nd Try

- **Statement:**
  - **If there is an efficient algorithm <u>A</u> that, given pk = (N, e) and ciphertext m^e, outputs m, (with > negigible probability) THEN the RSA assumption would not hold.**

    By contradiction: if we assume that the RSA assumption <u>does</u> hold, then <u>A</u> cannot exist.

Example

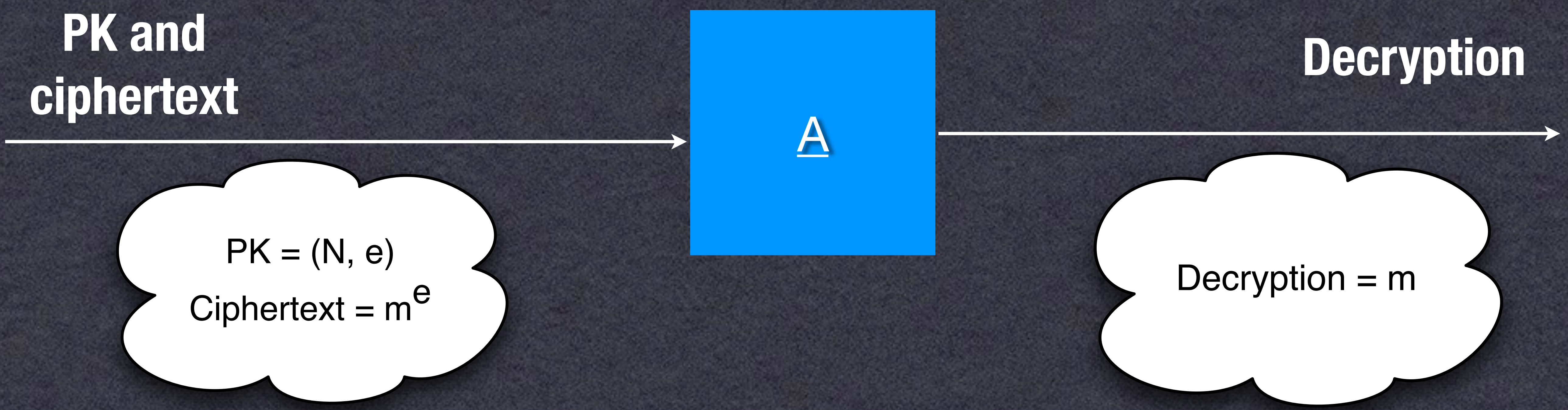
3rd Try

- **Statement:**
  - If there is an efficient algorithm A that, given pk = (N, e) and ciphertext m^e, outputs m, (with > negigible probability) THEN...

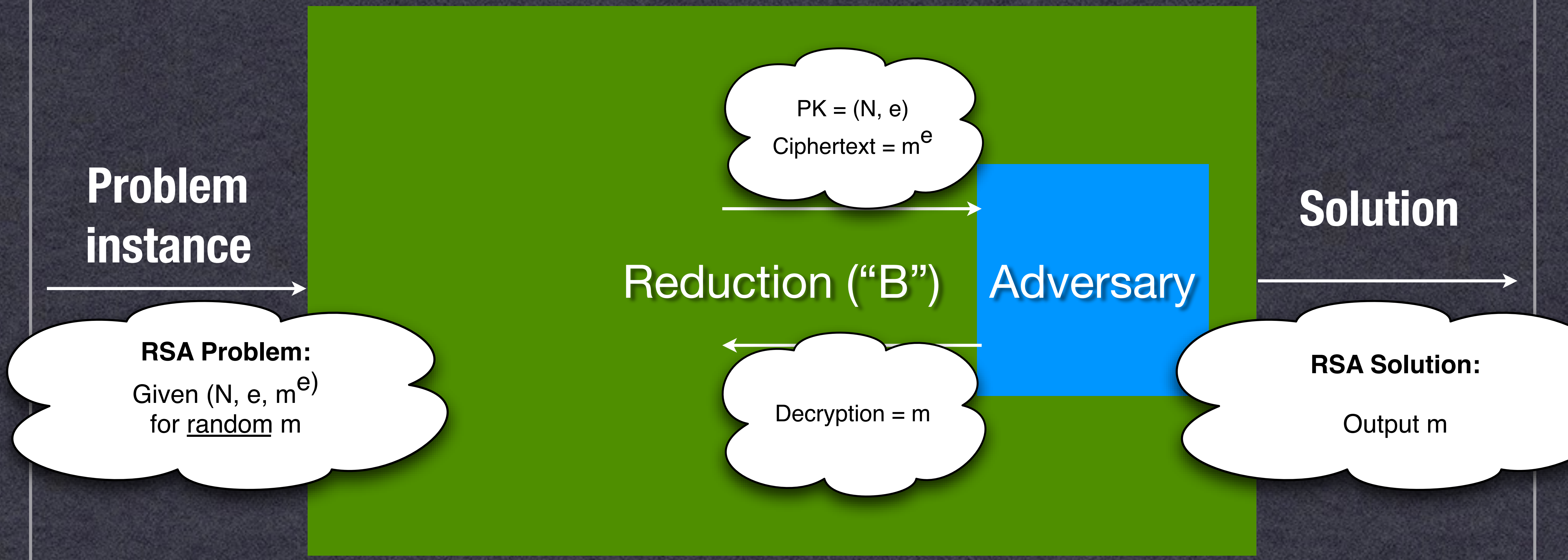
    we can show the existence of an efficient algorithm B that solves the RSA problem with > negl. probability.

    By contradiction: if we assume that B cannot exist, then A cannot exist.

Imagine that A exists



Then we can construct B



Security Definitions

- What does it mean to "break" a scheme?
  - Formal security definitions
  - Often described as "games"
  - Examples:

-Semantic security

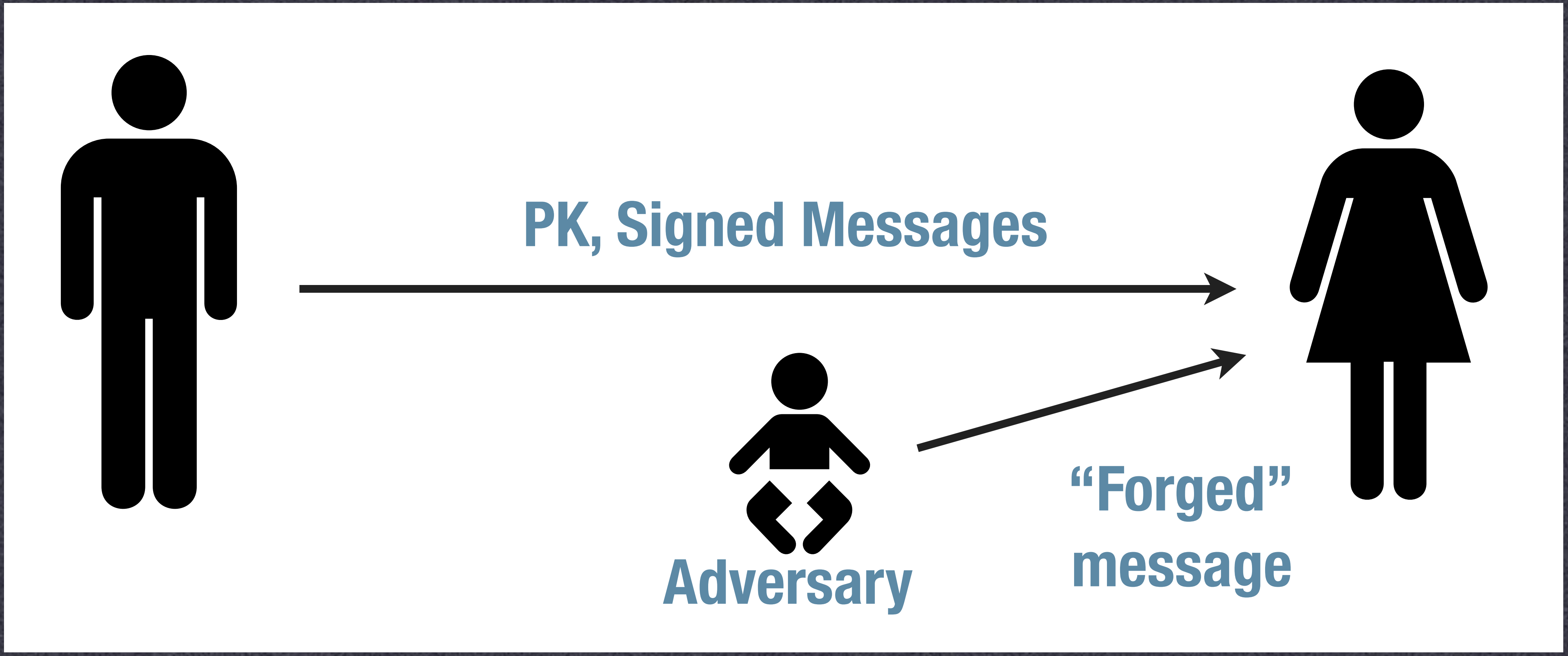
-Signature unforgeability

Digital Signatures

- The real world (scenario 1):

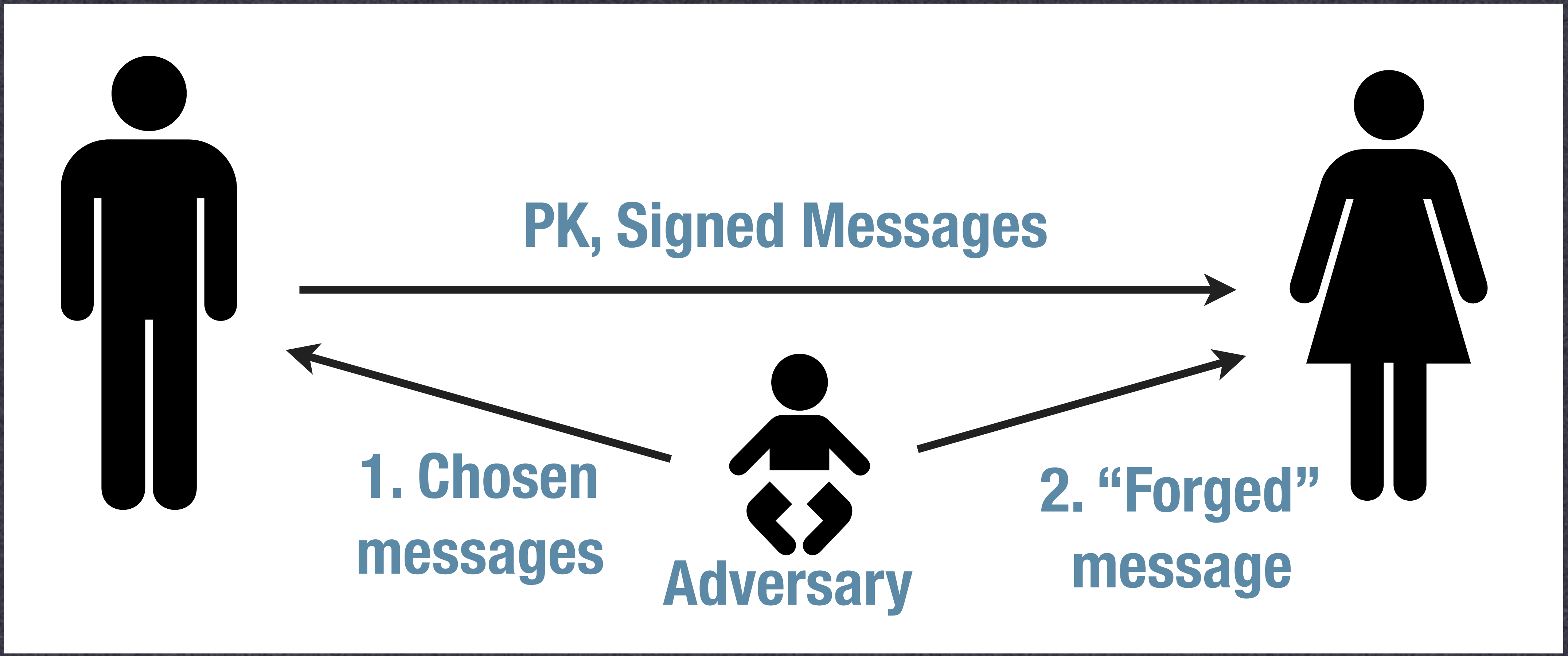
Digital Signatures

- The real world (scenario 1):

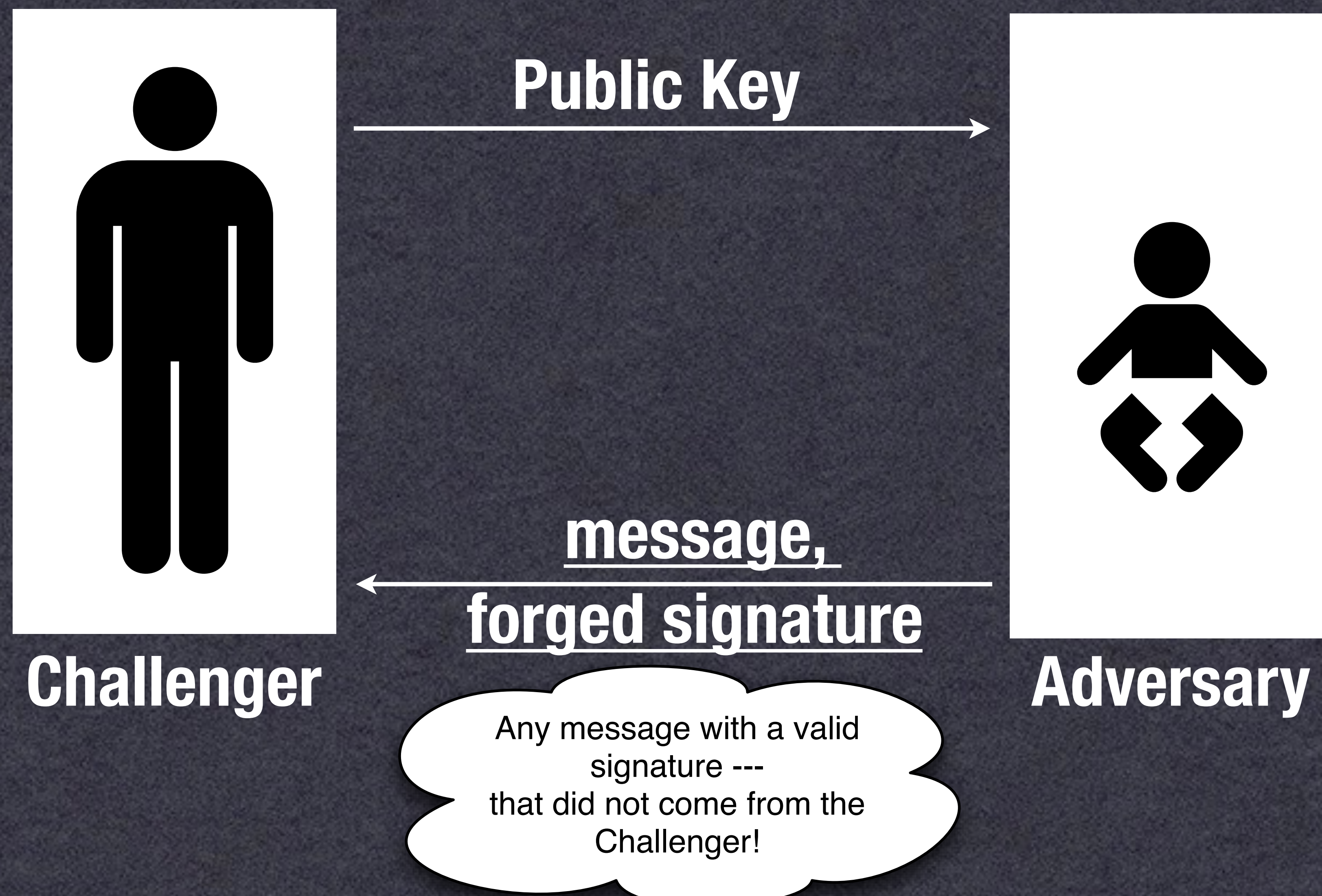


Digital Signatures

- The real world (scenario 2):

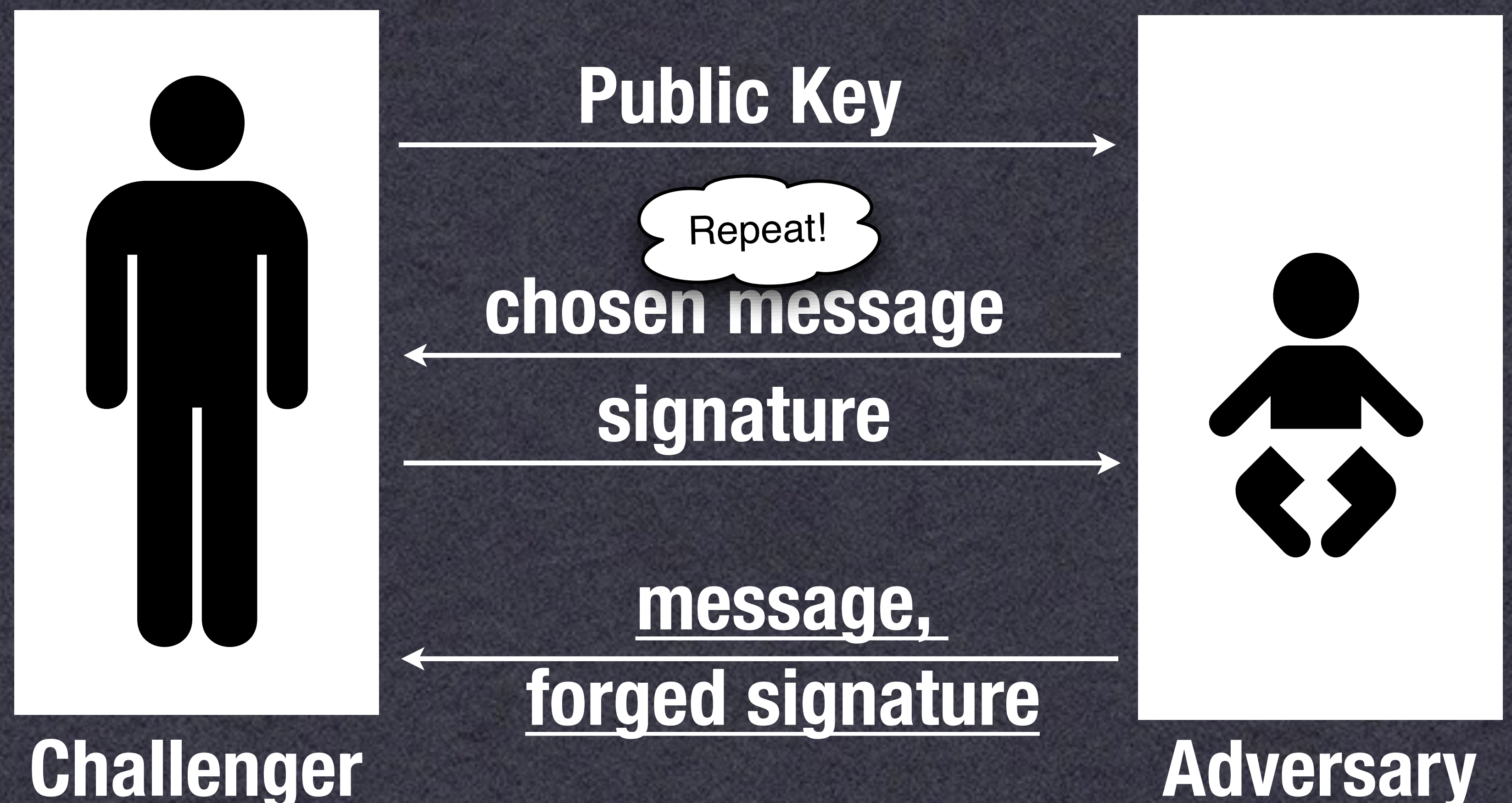


Security Game (1)



Existential Unforgeability (no messages)

Security Game (2)



Existential Unforgeability under Chosen Message Attack

EU-CMA Schemes

- **Problem:**
  - **"Textbook" RSA signatures are not EU-CMA**
  - **Simple attack, given (N, e):**

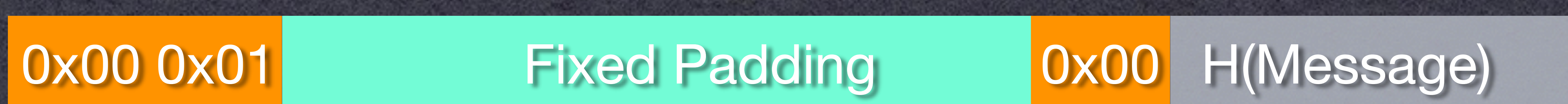
-Pick random s, compute $m = s^e$

The pair (m, s) **is a valid message, signature!**

(admittedly, m may not be very meaningful)

EU-CMA Schemes

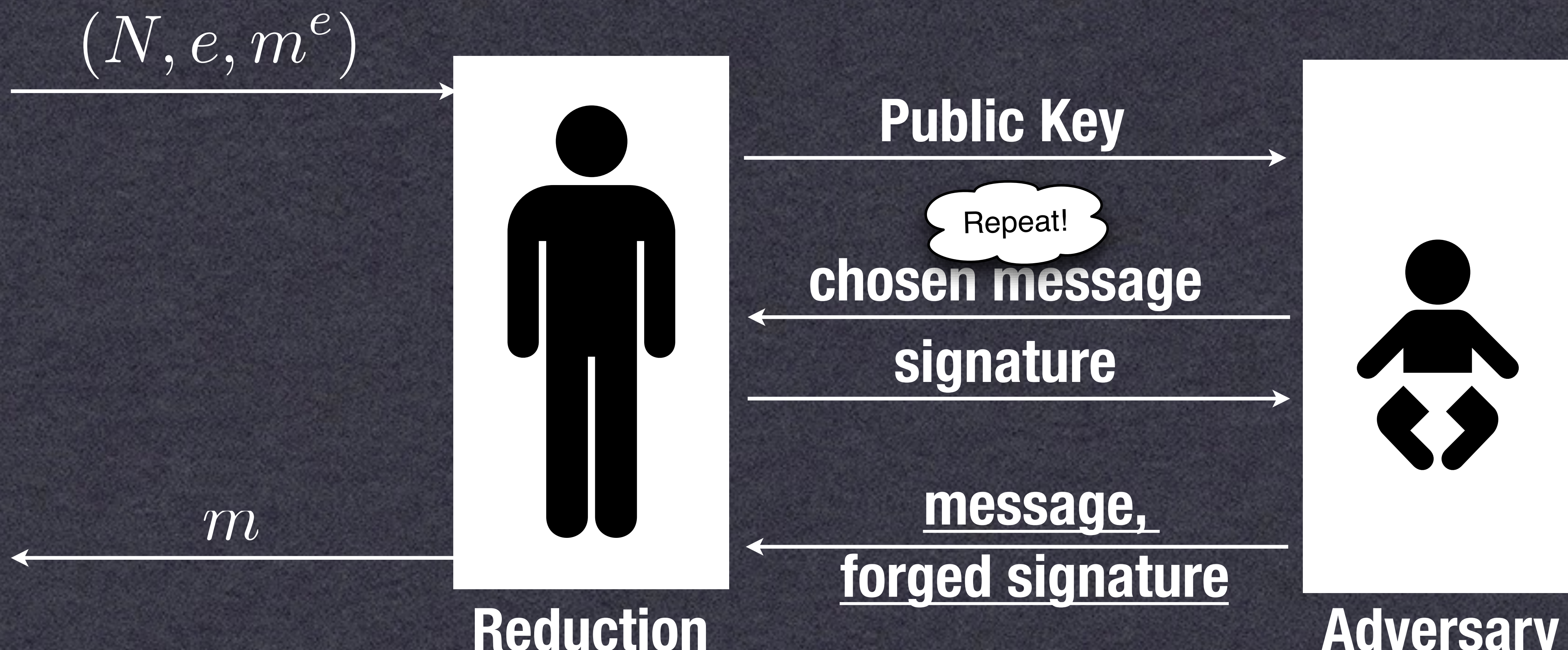
- Ad-hoc fix: Make the messages <u>meaningful</u>
  - Use a hash function H()
  - Add some defined padding bytes
  - Ex. PKCS #1 v1.5:



~ 1024 bits (128 bytes)

PKCS #1 v1.5 Signature

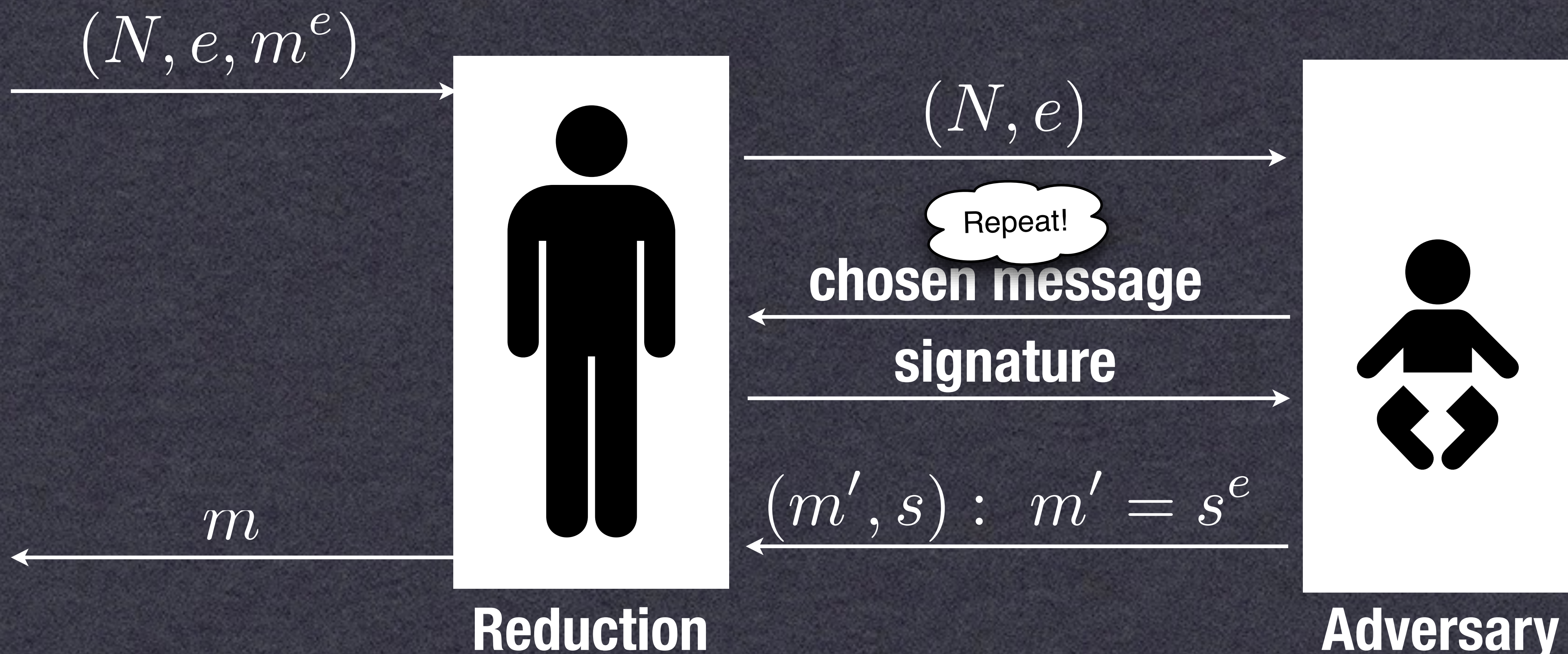
- This seems to sort-of solve the problem
  - But can we prove it's EU-CMA?
  - Let's think about how a proof might work:



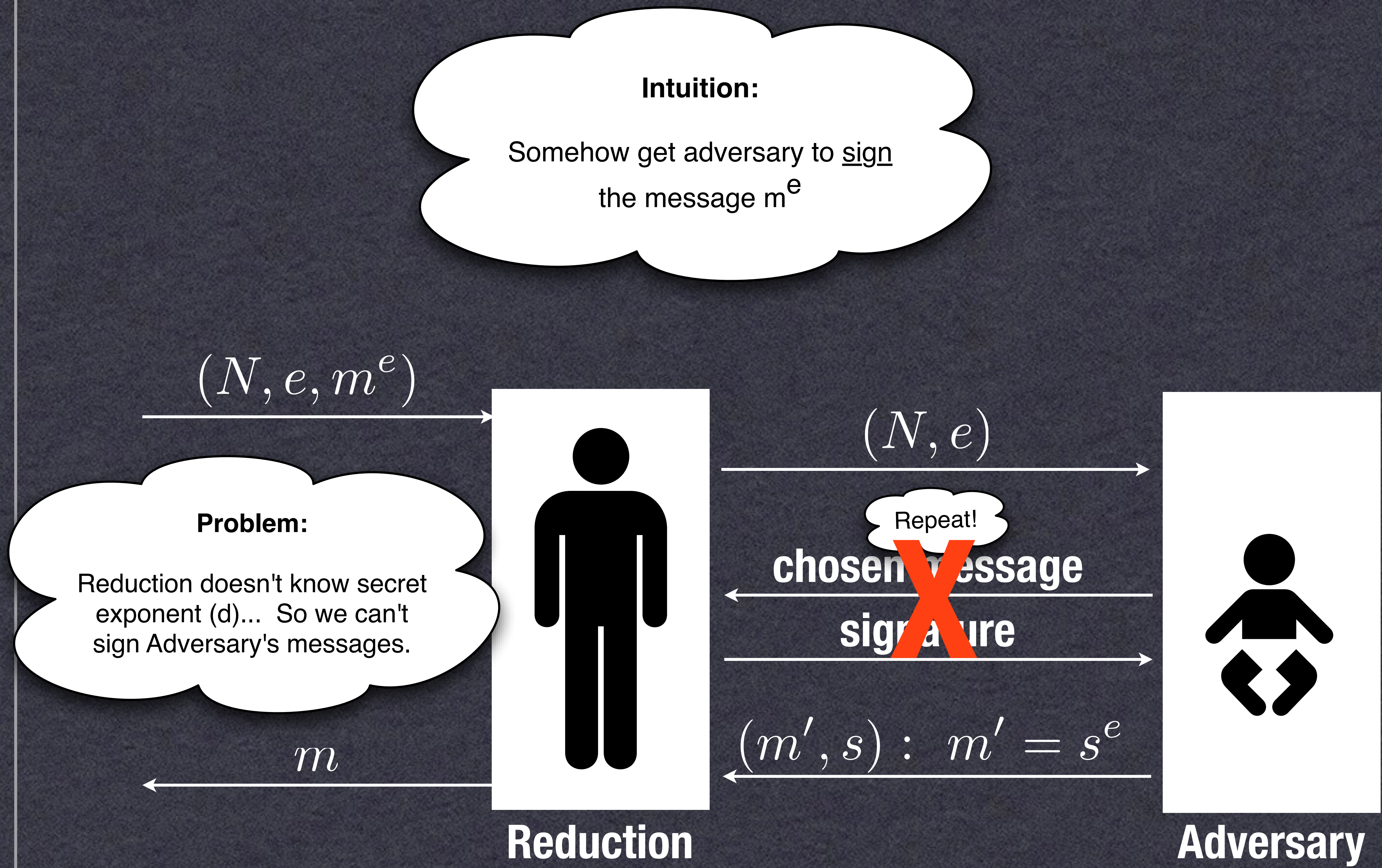
PKCS #1 v1.5 Signature

Intuition:

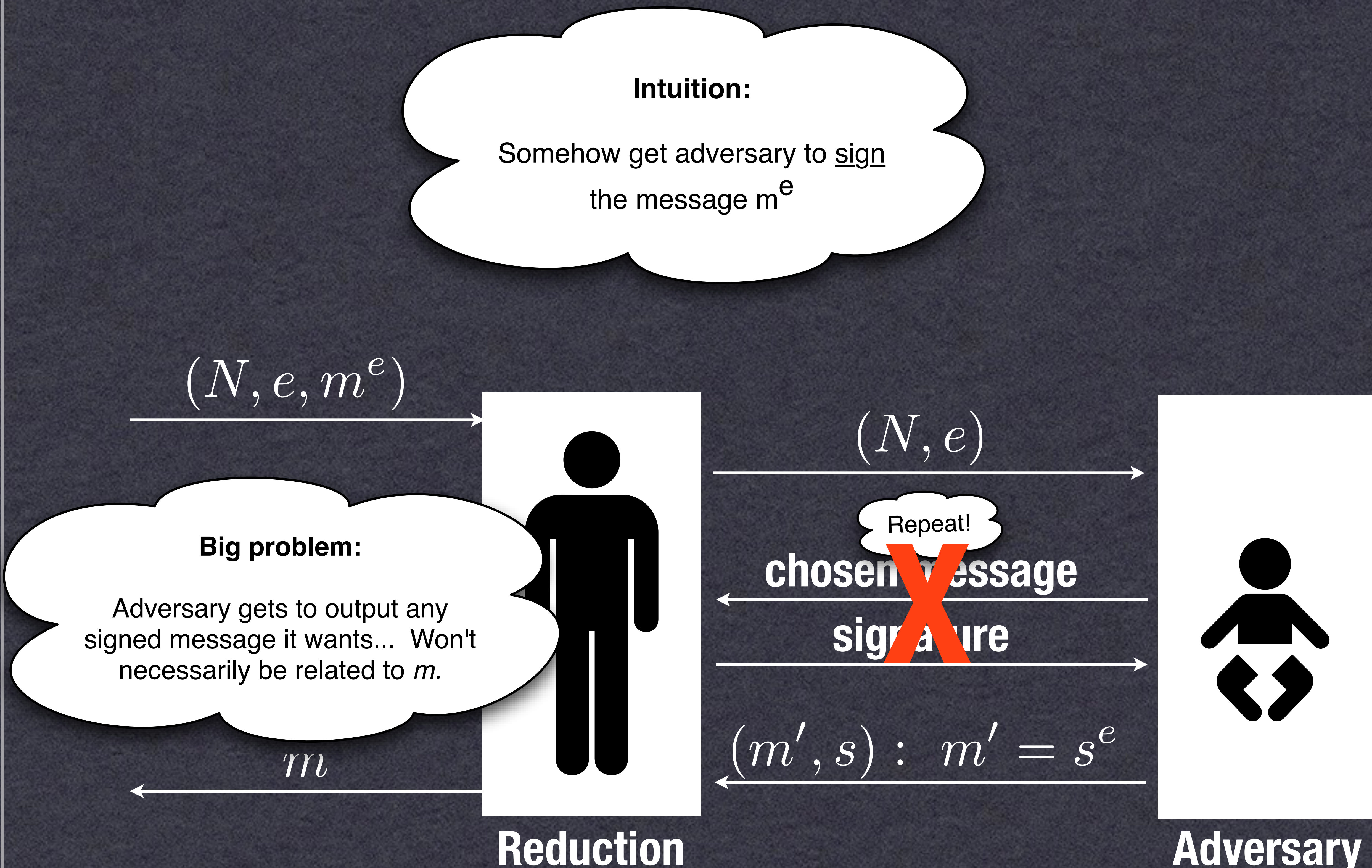
Somehow get adversary to <u>sign</u> the message m^e



PKCS #1 v1.5 Signature



PKCS #1 v1.5 Signature

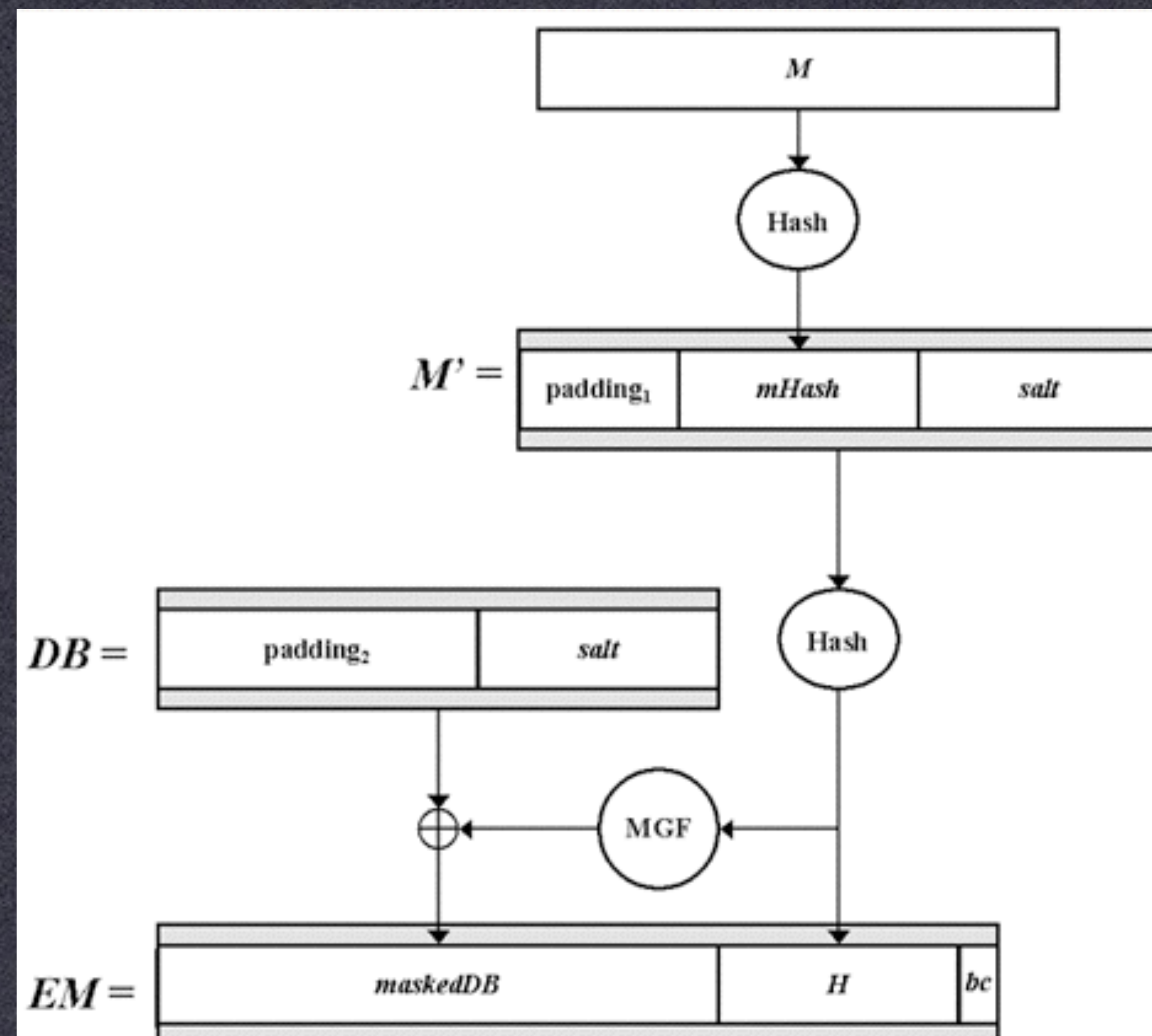


PKCS#1 v1.5

- Issues:
  - We want the adversary to sign a message of our choice -- but the EU-CMA game lets them sign any message
  - We need to give the adversary signatures on chosen messages... But we don't know the secret key!

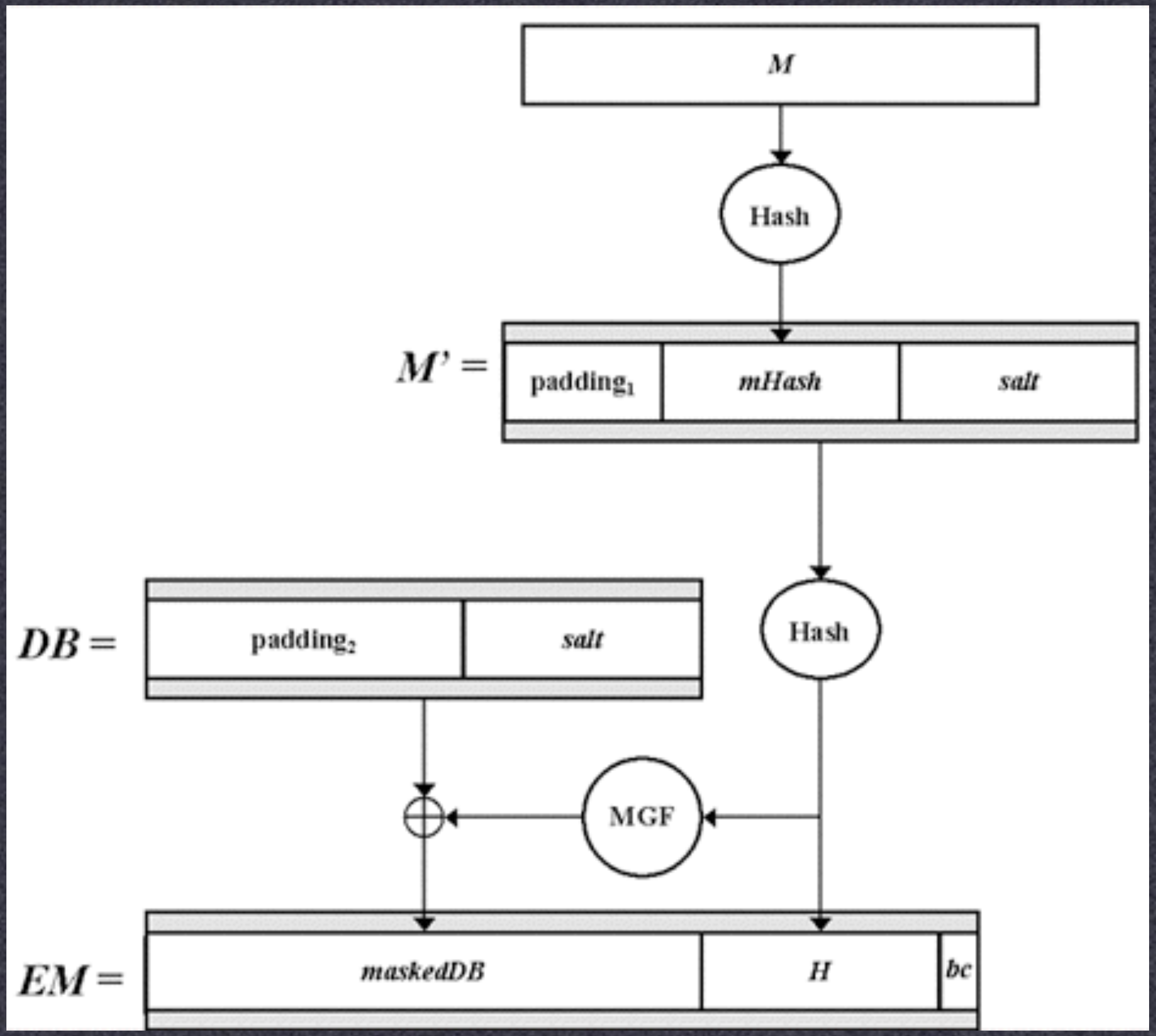
RSA-PSS

- Bellare/Rogaway (Eurocrypt '96):
  - Submitted to p1363
  - Adopted by RSA Security (PKCS)
  - Accepted by NIST



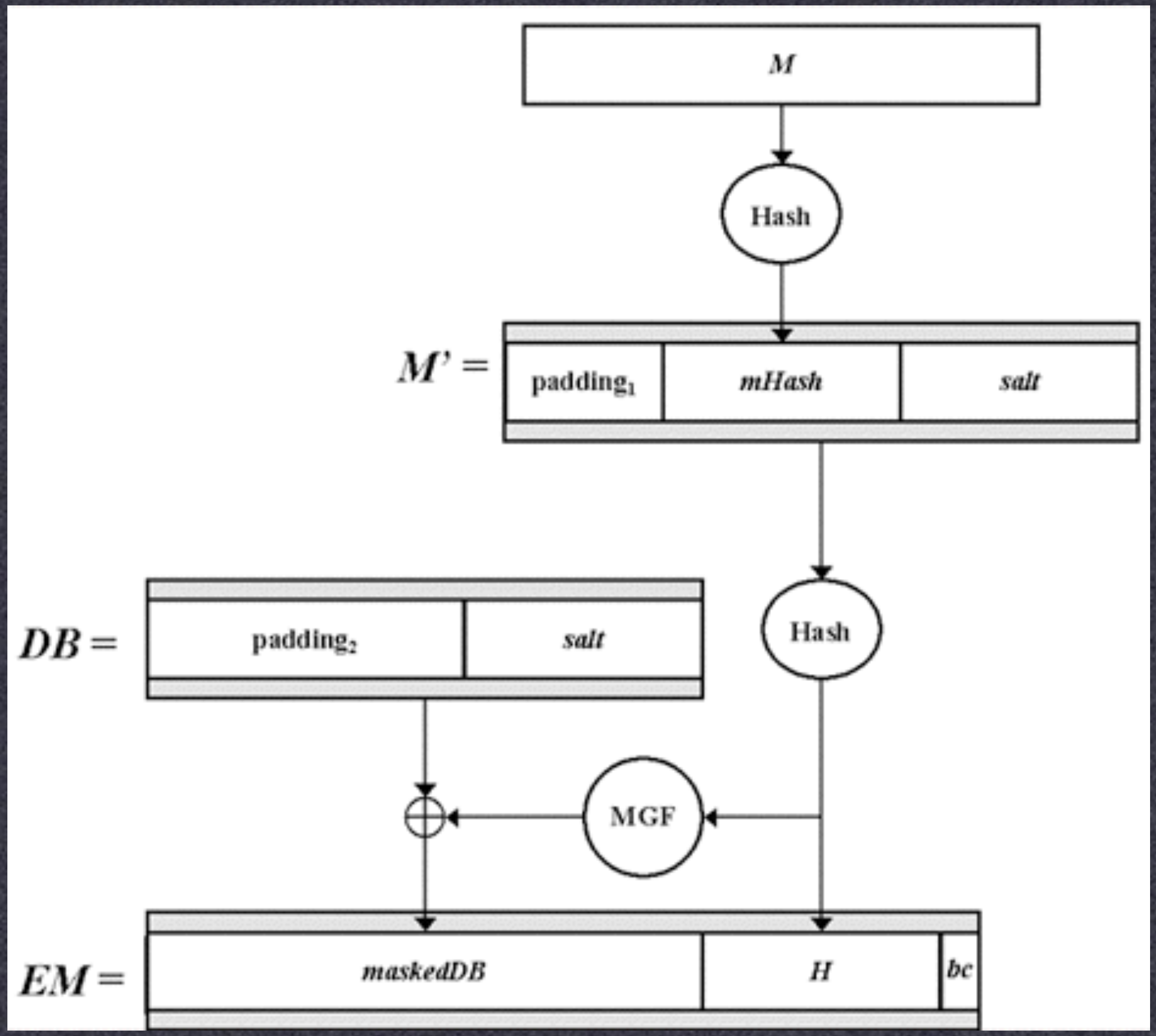
RSA-PSS

- Bellare/Rogaway (Eurocrypt '96):
  - Padding scheme
  - Applied to message M, to produce padded message EM
  - RSA sign EM
  - padding1/padding2 are fixed values

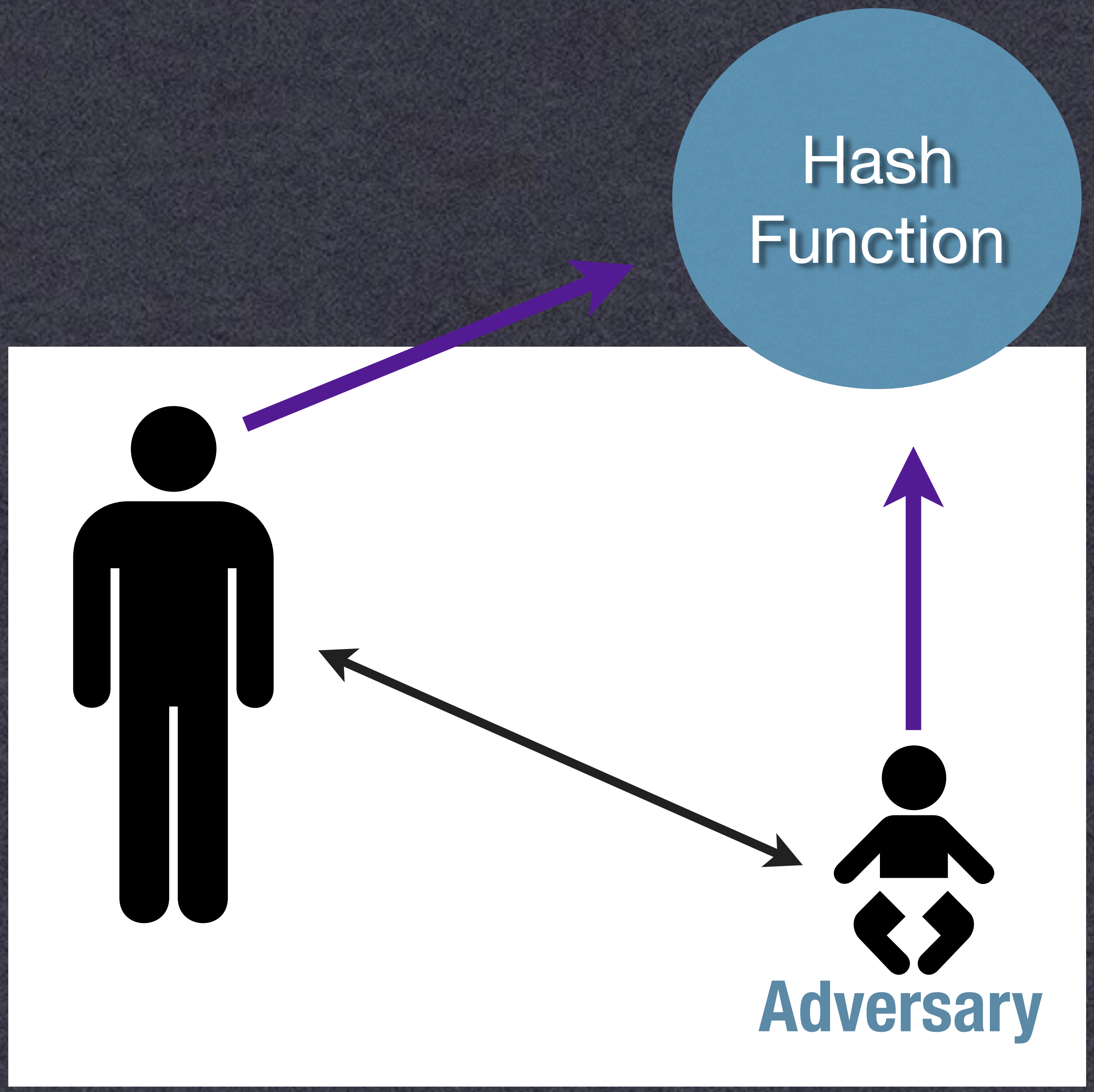


RSA-PSS

- **What are these hash functions (Hash, MGF)?**
  - Ideal hash functions, aka Random Oracles

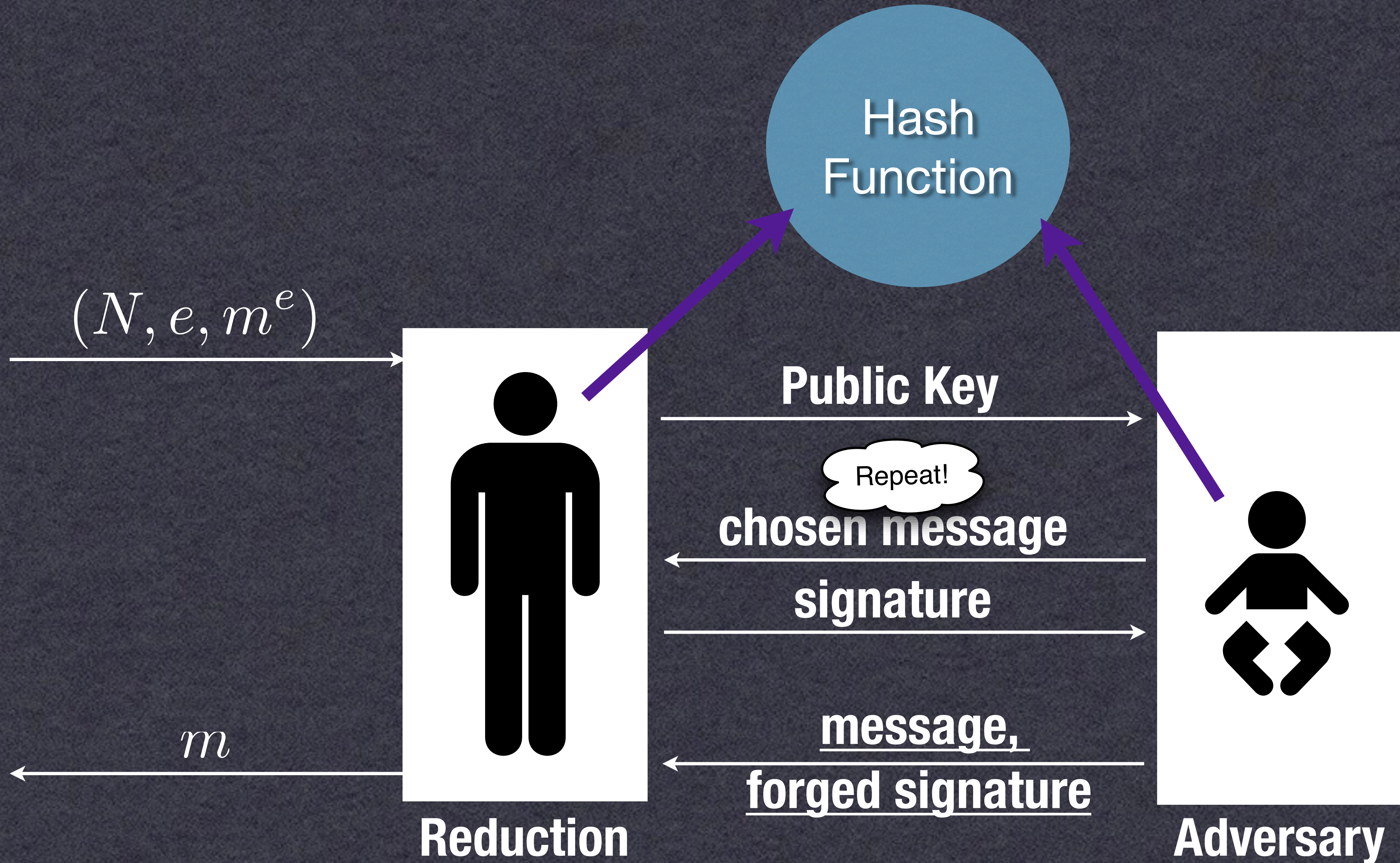


Random Oracles



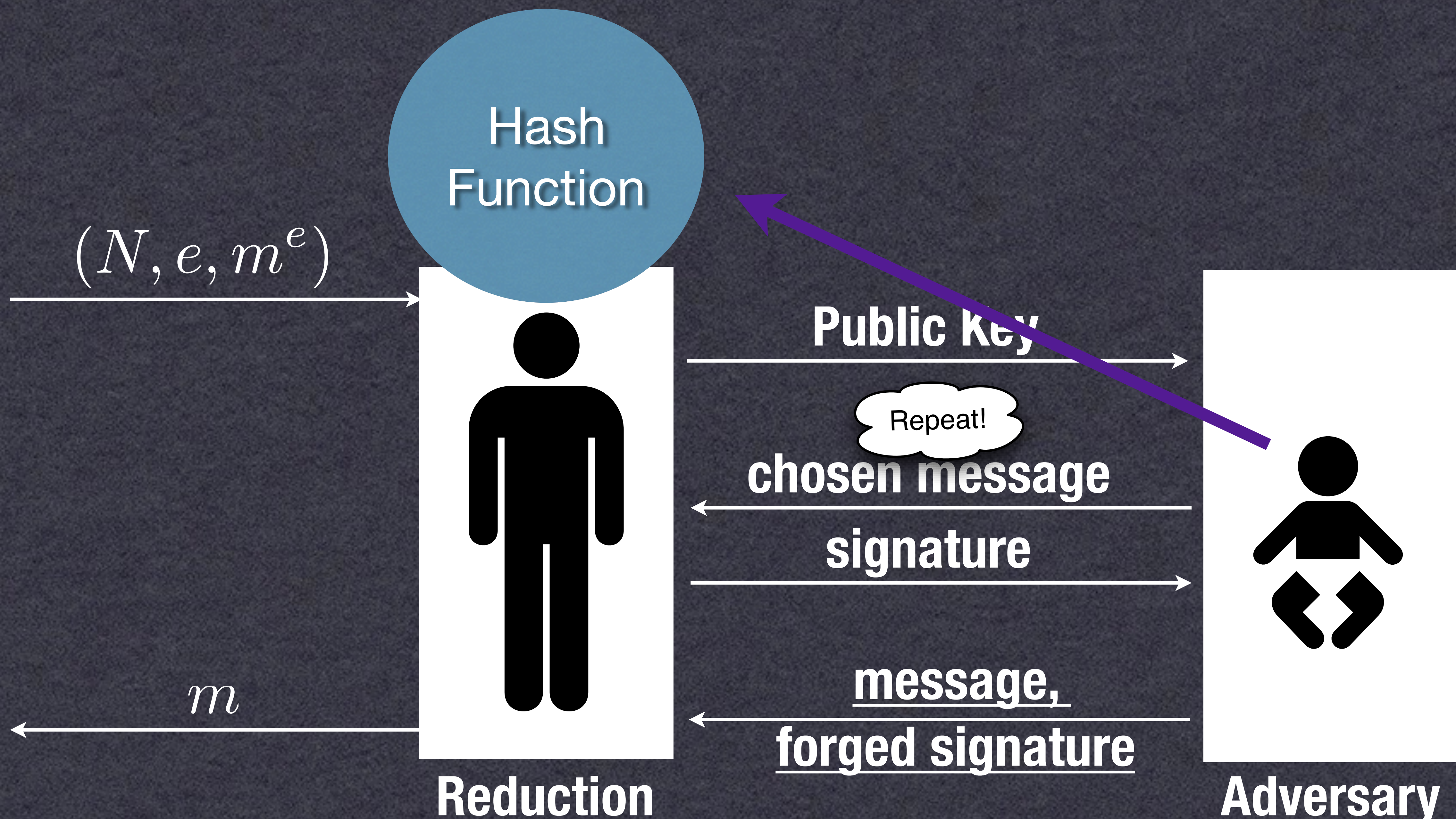
PKCS #1 v1.5 Signature

- In principle, Random Oracle is a "trusted third party"



PKCS #1 v1.5 Signature

- **But... our reduction can trick the adversary, and control the oracle**



RSA-PSS

- **Proof (board)**