Digital Signatures and Primality Testing

Schedule for today

Recap

Digital Signatures

1. The RSA-FDH signature scheme
2. Proving RSA-FDH secure

Primality Testing

1. Prime numbers and the prime number theorem
2. Trial division
3. The Fermat test and Carmichael numbers
4. Miller-Rabin test
5. The AKS test

What we did last time

Question 1

RSA-OAEP only allows us to encrypt a message m substantially shorter than $\log_2 N$. Why is this unavoidable if we want IND-CPA security and correctness?

Question 2

In the EUF-CMA security game, what if the attacker can come up with a fresh σ' on a message m that it has not seen before?

Question 3

Could an attack where an attacker finds a new σ' for a previously queried m work for RSA-FDH?

Digital Signatures using RSA: RSA-FDH

Signing key: secret d

Verification key N, e

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

Message m $\xrightarrow{\text{Hash } H}$ $h = H(m)$ $\xrightarrow{\text{Encrypt with private key}}$ $\sigma = h^d \bmod N$

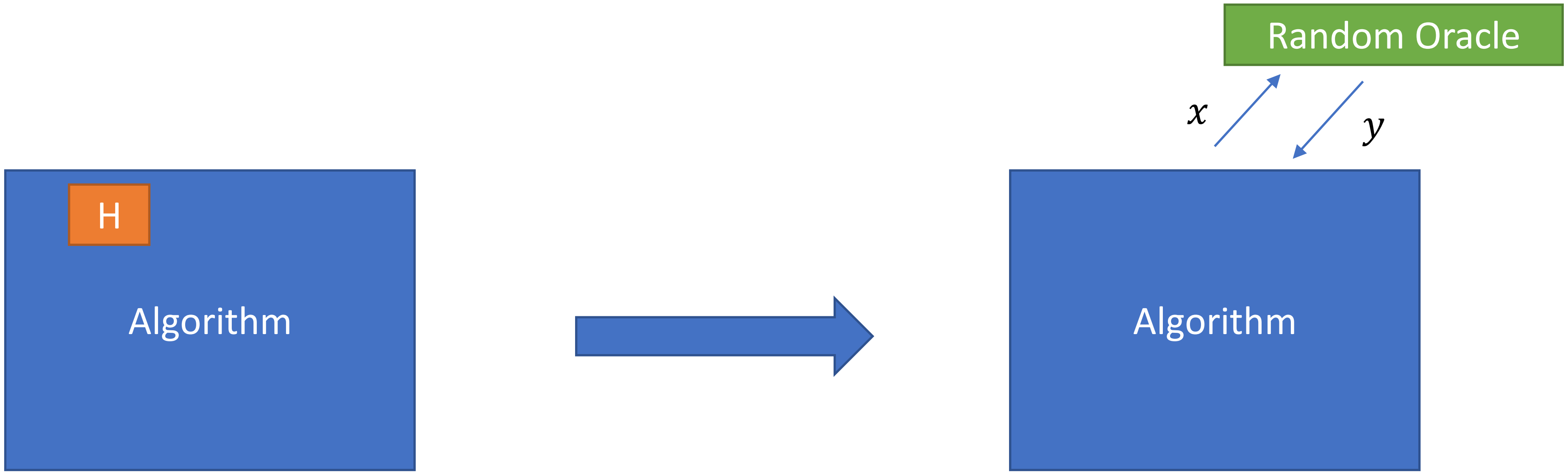
Verify m, σ:

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

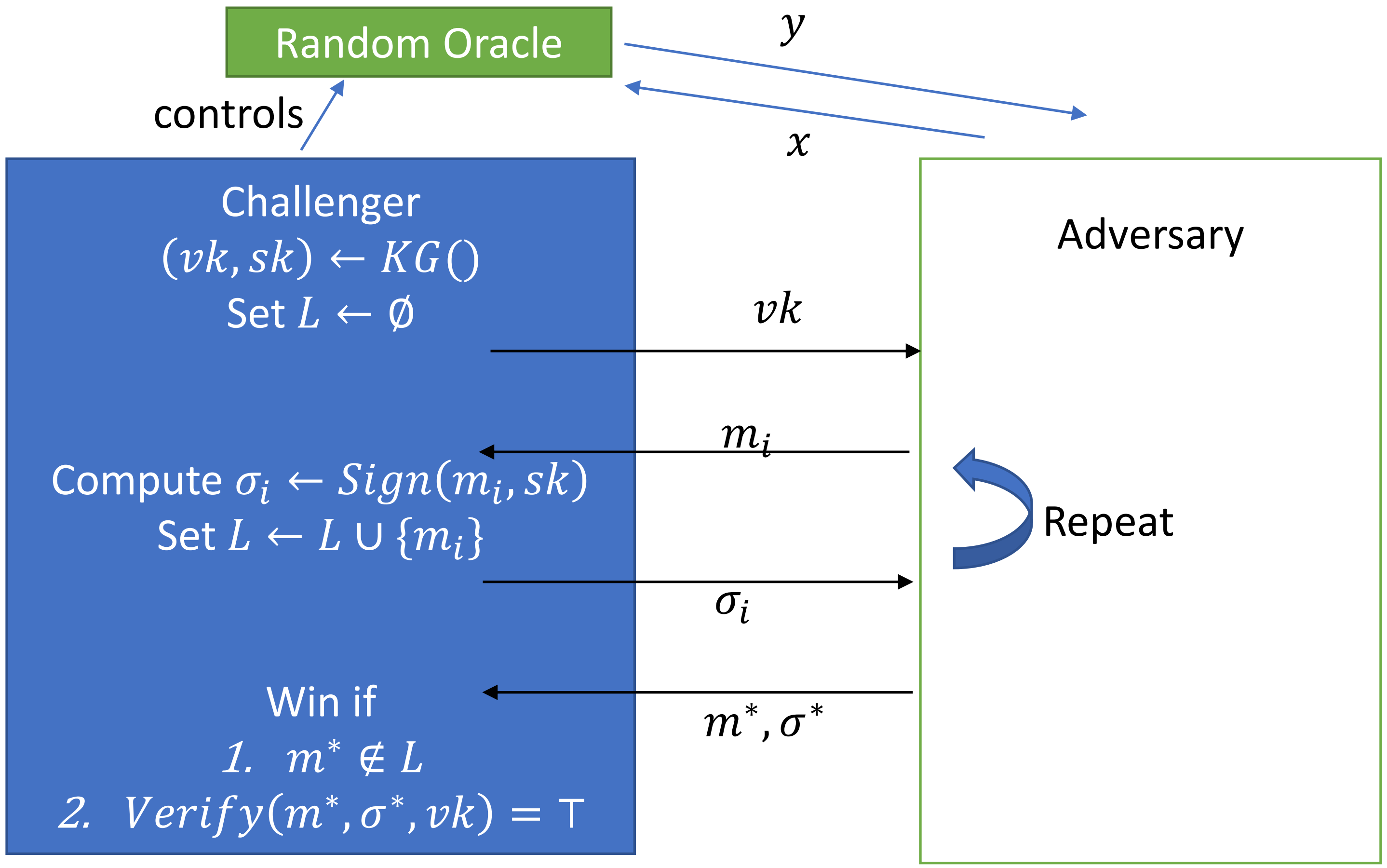
Any RSA instance for encryption can also be used for signing!

EUF-CMA security

Recap from Problem Sheet 5: the Random Oracle Model

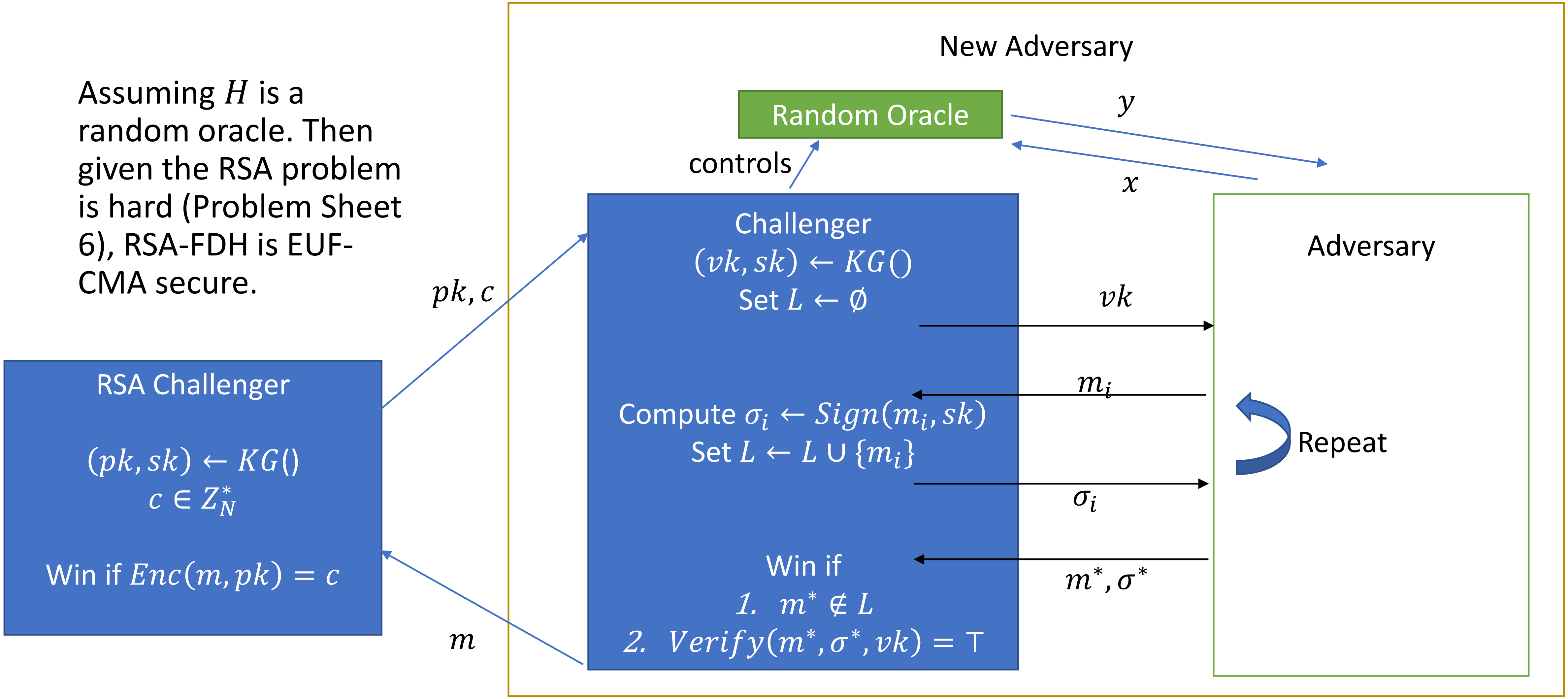


Looking at EUF-CMA



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.



Proof

Blackboard ☺

Primality testing

How many prime numbers are there?

Let $\pi(x) = |\{\mathrm{p\ prime} \mid \mathrm{p} < \mathrm{x}\}$. Then $\pi(x) \approx x/\ln(x)$

$\boldsymbol{x}$	$\boldsymbol{x/ln(x)}$	$\boldsymbol{\pi(x)}$
10^3	145	168
10^4	1,086	1,229
10^5	8,686	9,592
10^6	72,382	78,498
10^7	620,420	664,579

How many prime numbers are there?

Let $\pi(x) = |\{\mathrm{p\ prime} \mid \mathrm{p} \leq \mathrm{x}\}$. Then $\pi(x) \approx x/\ln(x)$

Assuming the primes are equally distributed in interval,

$$\Pr[p\ prime] \approx 1/\ln p$$

How to check that p is prime?

Idea 1: p prime iff only divisible by 1 and p

Trial-division by all numbers $k \in \{1, \ldots, \sqrt{p}\}$

Why is $\sqrt{p}$ sufficient?

<u>Runtime estimate</u>

1. Assume trial division by k each is one unit of time
2. $\sqrt{2^{1024}} = 2^{512}$ units of time needed
3. To break AES-128, we only need 2^{128} operations…

But!

Trial division is efficient for small numbers and to eradicate non-prime candidates early!

Any random number is divisible

1. by 2 with probability ½
2. by 3 with probability 1/3
3. by 5 with probability $1/5$
4. …

A random number is divisible by $2, 3$ or 5 with probability 0.73

Use to sieve before using ``the big guns''

Fermat's Test: idea

Fermat's little theorem

For any prime p,

$$a^{p-1} = 1 \; mod \; p$$

More generally: $a^{\phi(n)} = 1 \; mod \; n$ for $a \in Z_N^*$

Hope: if n not prime, then $\phi(n) \neq n - 1$ and very often $a^{n-1} \neq 1 \; mod \; n$

Fermat's Test

The algorithm for input n

1. For i $\in \{1, \dots, k\}$:
    1. Pick $a \in \{2, \dots, n-1\}$ uniformly at random
    2. Compute $b = a^{n-1} \; mod \; n$
    3. If $b \neq 1$ then output "Not prime"
2. Output "Probably prime"

How to choose k?

What test shows: if $a^{n-1} \neq 1 \; mod \; n$ then n not prime

What it doesn't show: n is prime

Example

$n = 17$:

- $3^{16} = 43046721 = 1\, mod\ 17$
- $2^{16} = 65536 = 1\, mod\ 17$

$n = 16$:

- $2^{15} = 32768 = 0\, mod\ 16$

More examples

$n = 561 = 3 \cdot 11 \cdot 17$:

- $5^{560} = 1 \, mod \, 561$
- $17^{560} = 1 \, mod \, 561$
- $235^{560} = 1 \, mod \, 561$

Carmichael Numbers

A composite n such that $\forall a \in Z_n^*: a^{n-1} = 1 \ mod\ n$

Examples:

- 561
- 1105
- 1729
- 2465
- ...

Theorem (Erdos): There are infinitely many Carmichael numbers ☹

Fixing Fermat's Test

Testing that $a^{n-1} = 1 \ mod\ n$ is necessary, but not sufficient

<u>Additional idea</u>: roots of unity

$$x^2 - 1 = 0\ mod\ n \leftrightarrow (x+1)(x-1) = 0\ mod\ n$$

If n is prime then ± 1 are only roots of 1 mod n

Fixing Fermat's Test

If n is odd, then $n - 1 = 2^s d$ where d is odd

Consider $a^d \; mod \; n, a^{2d} \; mod \; n, \ldots, a^{2^s d} \; mod \; n$ for $a \in Z_n^*$, then

- either $a^d = 1 \; mod \; n$
- or $a^{2^i d} = -1 \; mod \; n$

i.e. it cannot be that $a^{2^j d} \notin \{-1,1\} \; mod \; n$ but $a^{2^{j+1} d} = 1 \; mod \; n$

Miller-Rabin Test

The algorithm for input n

1. Let $n - 1 = 2^s d$ where d is odd
2. For i $\in \{1, \dots, k\}$:
    1. Pick $a \in \{2, \dots, n-1\}$ uniformly at random
    2. Compute $b = a^d \; mod \; n$
    3. If $b \notin \{-1,1\}$
        1. Set $i \leftarrow 1$
        2. While $i < s$ and $b \neq -1$
            1. $b \leftarrow b^2 \; mod \; n$
            2. If $b = 1$ return "Composite"
            3. $i \leftarrow i + 1$
        3. If $b \neq -1$ return "Composite"
3. Output "Probably prime"

Can we fool Miller-Rabin?

Short answer: No!

Less short answer

For every composite n there exist more than 2 roots of unity, which the test may choose!

Full answer

If n is composite, then $\geq 3/4$ of all a will make the test detect a composite! (e.g. https://shoup.net/ntb/ntb-v2.pdf Theorem 10.3)

Certificates of primality

Trial division: none

Fermat: well, Carmichael numbers…

Miller-Rabin: if a_i truly random, then yes*!

*repeating the test k times gives failure $\frac{1}{2^{2k}}$

Deterministic Poly-Time test of Primality

Long-standing open question: can we get exact primality test in polynomial time?

<u>Agrawal, Kayal, Saxena 2002: YES!</u>

Their approach: $n \geq 2$ is prime iff $(X - a)^n = X^n - a \ \ mod\ n$ for some integer a coprime to n

Their algorithm is accurate, but in practice slower than Miller-Rabin.

Summary

1. RSA-FDH is EUF-CMA secure
2. The Fermat primality test can be fooled
3. Miller-Rabin is more reliable