Recitation 7: RSA review

6.5610, Spring 2023

March 24, 2023

1 RSA

Recall the RSA trapdoor one way permutation, which has forward function F and an inversion function I.

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$.
  - Sample two random λ-bit primes p, q such that $p = q = 5 \mod 6$.
  - Output $\mathsf{sk} = (p, q)$ and $\mathsf{pk} = N$, where $N = p \cdot q$.
- $F(\mathsf{pk}, x) \to y$
  - The input space is $Z_N = \{0, 1, ..., N-1\}$
  - Output $y = x^3 \mod N$
- $I(\mathsf{sk}, y) \to x$
  - We want to solve for x such that $x^3 = y \mod N$
  - Which means that $x^3 = y \mod p$ and $x^3 = y \mod q$.
  - Find the cube roots x_p, x_q of y mod p and q, respectively.
  - Use the Chinese Remainder Theorem (CRT) to find $x \in Z_N$ such that $x = x_p \mod p$ and $x = x_q \mod q$.
  - Output x

1.1 Finding cube roots modulo a prime

If p is a prime congruent to $5 \mod 6$, then for all $a \in Z_p^*$, at least one element r of $a^{\frac{p+1}{6}}, -a^{\frac{p+1}{6}}$ is such that $r^3 = a \mod p$. The proof is in the lecture notes.

1.2 Chinese Remainder Theorem (CRT)

Let p and q be distinct primes. For all integers a and b, the pair of congruences $x = a \mod p$ and $x = b \mod q$ has a unique and efficiently computable solution modulo pq.

Proof idea: Let $p_1 = p^{-1} \mod q$ and $q_1 = q^{-1} \mod p$. Then the solution is:

$$x = aq_1q + bp_1p \mod pq$$

1.3 Example RSA encryption and decryption

We will walk through an example encryption and decryption of a message with real, small numbers.

- Let the secret key be $p = 5, q = 11$. Thus, the public key is $N = 55$.
- Let's encrypt message $m = 8$. We get $c = m^3 = 512 \mod 55 = 17$.
- To decrypt the ciphertext $c = 17$, we want to find x such that $x^3 = 17 \mod N$.
- This means that $x^3 = 17 \mod 5$ and $x^3 = 17 \mod 11$
- We want to find x_p such that $x_p^3 = 17 = 2 \mod 5$.
  - We use the modular cube root algorithm: $2^{\frac{p+1}{6}} = 2 \mod 5$.
  - Now, $2^3 = 8 = 3 \mod 5$. That's not right, so we try the negative. $-2 \mod 5 = 3$. We check that $3^3 = 27 = 2 \mod 5$.
  - So $x_p = 3$.
- Similarly, we want to find x_q such that $x_q^3 = 17 = 6 \mod 11$.
  - $6^{\frac{q+1}{6}} = 36 = 3 \mod 11$.
  - $3^3 = 27 = 5 \mod 11$, which is not right. We try the negative. $-3 \mod 11 = 8$. We check that $8^3 = 512 = 6 \mod 11$.
  - So $x_q = 8$.
- Now we want to find the original message $x \in Z_N$ such that $x = x_p \mod p$ and $x = x_q \mod q$.
- We use the Chinese Remainder Theorem, which says that $x = x_p q_1 q + x_q p_1 p \mod pq$, where $p_1 = p^{-1} \mod q$ and $q_1 = q^{-1} \mod p$.
- $p_1 = 5^{-1} \mod 11 = 9$ and $q_1 = 11^{-1} \mod 5 = 1$
- So $x = 3 \cdot 1 \cdot 11 + 8 \cdot 9 \cdot 5 = 8 \mod 55$, which is the original message!

2 Practice problems

2.1 Example: RSA variant

Bob extends RSA so that message m is encrypted as the pair $(r^e, h(r)m^e)$, where h is a hash function mapping inputs to Z_n^*. Argue that his new scheme is not CCA secure.

Solution: It is definitely malleable: $E(2m) = (r^e, h(r)(2m)^e)$, and so is not CCA secure.

2.2 Example: RSA digital signature

Consider the basic RSA signature scheme defined by

$$\text{Sign}(m) = m^d \pmod n,$$

and

$$\text{Verify}(m, \sigma) = 1 \text{ if and only if } \sigma^e = m \pmod n,$$

where the secret key is the pair (d, n), and the public key is the pair (n, e), where n is a product of two primes and $e \cdot d = 1 \mod \varphi(n)$.

(a) Is this signature scheme secure?

(a) Is the corresponding hash-and-sign scheme, where the signature algorithm is defined by $\text{Sign}(m) = H(m)^d \pmod n$, secure in the Random Oracle Model? Explain your answer (though you do not need to provide a formal proof).

Solution:

(a) This signature scheme is not secure. The adversary can easily sign a message by first (arbitrarily) choosing a signature $\sigma \in Z_n^*$, and then computing the message $m = \sigma^e \mod n$. Note that (m, σ) is a valid message/signature pair.

(b) Yes, this scheme is secure in the random oracle model. Intuitively the reason is the following: First note that the signing oracle is of no use to the adversary since it can easily be simulated by simulating the Random Oracle H, as follows: Whenever the adversary requests a signature of a message m_i, simply choose at random $\sigma_i \leftarrow Z_n^*$ and set $H(m_i) = \sigma_i^e \mod n$. Note that σ_i is a valid signature of m_i. Thus, the adversary is basically just getting random values of Z_n^* from the signing oracle; these values are of no use to him in forging signature for other messages.

Therefore, it suffices to argue that the adversary cannot generate a signature of any (new) message, assuming the hardness of the RSA problem. To this end, note that for any (new) adversarially chosen message m, the value $H(m)$ is truly random (and unknown before querying the oracle H). Therefore, generating a valid signature for m requires computing $H(m)^d \mod n$, where $H(m)$ is a truly random element, which is equivalent to solving the RSA problem.

2.3 Example: Randomized RSA digital signature

Suppose we are interested in developing a randomized digital signature scheme, where a message may have many signatures, and security now also requires that an Adversary is not able to produce a new but different signature for a message he has seen other signatures for already.

Consider the following randomized RSA-based signature proposal. We have $PK = (n, e, H)$ and $SK = (d)$ as usual for RSA, where H is a hash function modeled as a random oracle from messages to Z_n^*. The signature $\sigma(m)$ of a message m is defined

$$\sigma(m) = (H(r), (H(m) \cdot r)^d \pmod n)$$

where r is a fresh random value from Z_n^*.

Is σ secure (using the expanded definition of signature security given above)? Explain.

Solution: No, it is not secure.

Having seen one signature $\sigma(m)$ for a known message m, the Adversary can produce a signature for an arbitrary other message m' as follows. Note that the Adversary can compute $H(m)$ and $H(m')$, since m and m' are known and H is public. Also, the Adversary can compute $H(m) \cdot r = \left((H(m) \cdot r)^d\right)^e \pmod n$.

The Adversary can then compute a value

$$r' = (H(m) \cdot r)/H(m') \pmod n$$

so that

$$H(m) \cdot r = H(m') \cdot r' \pmod n \ .$$

The Adversary can then easily compute $H(r')$, which he can combine with the known signature for m to compute the signature for m':

$$\begin{aligned} \sigma(m') &= (H(r'), (H(m') \cdot r')^d \mod n) \\ &= (H(r'), (H(m) \cdot r)^d \mod n) \ . \end{aligned}$$

3 References

https://65610.csail.mit.edu/2023/lec/l13-rsa.pdf
6.857 past quizzes