CSC110 Lecture 19: Public-Key Cryptography and the RSA Cryptosystem

Hisbaan Noorani

October 27, 2020

Contents

For your reference, here is one key definition and the two main theorems about modular exponentiation that we'll use today.

(*Fermat's Little Theorem*) Let $p, a \in \mathbb{Z}$ and assume p is prime and that $p \nmid a$. Then $a^{p-1} \equiv 1 \pmod{p}$.
We define the function $\varphi : \mathbb{Z}^+ \to \mathbb{N}$, called the **Euler totient function** (or **Euler phi function**), as follows:

$$\varphi(n) = \big|\{a \mid a \in \{1, \ldots, n-1\}, \text{ and } \gcd(a, n) = 1\}\big|.$$

We have the following formulas for special cases of $\varphi(n)$:

- For all primes $p \in \mathbb{Z}$, $\varphi(p) = p - 1$.
- For all *distinct* primes $p, q \in \mathbb{Z}$, $\varphi(pq) = (p-1)(q-1)$.

(*Euler's Theorem*). For all $a \in \mathbb{Z}$ and $n \in \mathbb{Z}^+$, if $\gcd(a, n) = 1$ then $a^{\varphi(n)} \equiv 1 \pmod{n}$.

1 Exercise 1: Reviewing modular exponentiation

1. Let $a, p \in \mathbb{Z}$ and assume that p is prime that $\gcd(a, p) = 1$. Using Fermat's Little Theorem, simplify each of the following expressions modulo p by reducing it to 1 or an expression of the form a^e, where the exponent e is positive and as small as possible. We've done the first one for you.

   Note: start out with $a^{p-1} \equiv 1 \pmod{p}$ and try to manipulate it to become what you want.

Power of a	Simplified expression modulo p
a^{p-1}	1
a^p	a
a^{2p-2}	1
a^{2p}	a^{2p}
a^{p^2-1}	a^{p-1}
a^{p^2}	a^p

2. let $p = 23$ and $q = 5$.

   (a) What is $\varphi(pq)$?

$$\begin{aligned}\varphi(pq) &= (23-1)(5-1) \\ &= 88\end{aligned}$$

(b) Using euler's theorem, calculate each of the following remainders (modulo $pq = 115$). we have done the first row for you (note that $(p-1)(q-1) = 88$ – keep this number in mind).
Note: start out with $2^{88} \equiv 1 \pmod{115}$ and try to manipulate it to become what you want.

power of 2	remainder modulo $pq = 115$
2^{88}	1
2^{89}	2
2^{176}	1
2^{180}	16
2^{880}	1
2^{8801}	2

2 Exercise 2: Reviewing the RSA Cryptosystem

1. The following parts get you to manually trace through the steps of the RSA cryptosystem. The calculations themselves are pretty straightforward, we just want you to review the algorithm and practice all of the steps!

   (a) Suppose we start with the primes $p = 23$ and $q = 5$. What are n and $\varphi(n)$?
   $n = 115$

   $$\begin{aligned}\varphi(n) &= \varphi(pq) \\ &= (23-1)(5-1) \\ &= 88\end{aligned}$$

   (b) Suppose $e = 3$. Find the corresponding value for d such that $ed \equiv 1 \pmod{\varphi(n)}$. (You can just use trial and error here, or the `modular_inverse` function from last week!)
   $d = 59$

   (c) What are the RSA private and public keys for these choices of p, q, and e?
   RSA private $= (p, q, d) = (25, 5, 59)$
   RSA public $= (n, e) = (115, 3)$

   (d) Suppose you want to encrypt the number 77 using the public key. What is the resulting "ciphertext" (the encrypted number)? You can use Python as a calculator to answer this.

   $$\begin{aligned}c &= m^e \ \% \ n \\ &= 77^3 \ \% \ 115 \\ &= 98\end{aligned}$$

   (e) Verify that if you decrypt this ciphertext with the private key, you get back the original number 77.

   $$\begin{aligned}m' &= c^d \ \% \ n \\ &= 98^{59} \ \% \ 115 \\ &= 77\end{aligned}$$

2. The following are some conceptual questions about the RSA algorithm to check your understanding of this algorithm.

   (a) Why does the key generation phase require that $\gcd(e, \varphi(n)) = 1$?
   This gives us the number of numbers that are coprime to n. In order to find the modular inverse, d, the $\gcd(e, \varphi(n)$ must be 1.

(b) We know that picking $e = 1$ satisfies $\gcd(e, \varphi(n)) = 1$. Yet why is $e = 1$ not a good choice of e?
Any number d would satisfy $ed \equiv 1 \pmod{\varphi(n)}$.
Making $e = 1$ would make it very easy to compute d:
$ed \equiv 1 \pmod{\varphi(n)}$
$d = 1$

(c) When we discussed encrypting a message, we said that the message had to be in the range $\{1, 2, \ldots, n-1\}$ (where the n is from the public key) Why do we not allow numbers larger than $n - 1$ to be encrypted?
This would introduce ambiguity. Ther would be more than one possible solution when you're decrypting that would me mathematically correct but not actually correct.