Lecture 5.2: Public-key cryptography and RSA

Matthew Macauley

Department of Mathematical Sciences
Clemson University
http://www.math.clemson.edu/~macaule/

Math 4190, Discrete Mathematical Structures

RSA: a different type of cryptosystem

The RSA cryptosystem was develeoped in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman.

It allows two people to exchange messages "in plain sight".

Suppose I want to send *you* a secret message, e.g., your midterm exam score.

For privacy reasons, I cannot just email it to you in plain text. What if somebody snoops?

Instead, you create a publicaly available encryption function $e(x)$.

I compute $e(\text{score})$, and email this to you.

You have secret information that allows you to easily compute the inverse (decryption) function, $d = e^{-1}\colon X \to X$.

However, for everybody else, this is basically impossible.

RSA is an example of a public-key cryptosystem, and these are widely used today.

All of these are characterized by an encryption function $e\colon X \to X$ that is easy to compute but almost impossible to invert, unless you have the "secret key".

Unlike the methods in the previous lecture, public-key systems are asymmetric cryptosystems.

How RSA works

As the intended recipient of encrypted messages, you need to take the following steps:

1. Choose 2 (large) primes, e.g., $p = 17$, $q = 19$.

   Normally, these would be several hundred digits in length.

2. Let $n = pq = 17 \cdot 19 = 323$.

   Factoring such a large n is basically impossible. Only you know p and q!

3. Let $A = \varphi(n) = (p-1)(q-1) = 16 \cdot 18 = 288$.

   Without knowing how to factor n, computing $\varphi(n)$ is basically impossible.

4. Pick $E < \varphi(n)$ such that $\gcd(E, \varphi(n)) = 1$. [Let's pick $E = 95$].

   We'll learn how to efficiently find such an E.

   Your public key is $(n, E) = (323, 95)$, and your (public) encryption function is

   $$e(x) = x^E \pmod{n}, \qquad \big[e(x) = x^{95} \pmod{323}\big].$$

5. Compute your private key, $D = E^{-1} \pmod{A}$, i.e., the solution to $Ex \equiv 1 \pmod{A}$.

   The decryption function, known only to you, is (modulo n)

   $$d(y) = y^D = (x^E)^D = x^{ED} \equiv x \pmod{n}, \qquad \big[d(y) = y^{191} \pmod{323}\big].$$

Example: How I can send you your exam score using RSA

You choose $p = 17$, $q = 19$, and publish your public key $(n, E) = (323, 95)$.

You compute your private key $D = E^{-1} = 191$. (We'll learn how to do this.)

I use your public encryption function to compute

$$e(\text{score}) = (\text{score})^{95} \equiv 307 \pmod{323},$$

I email you **307**, and then you use your private key to decrypt this message:

$$d(y) = y^{191} \pmod{323}, \qquad d(307) = 307^{191} \pmod{323}$$
$$\equiv \mathbf{86} \pmod{323}.$$

We need to learn how to do the following

1. Find $E \in \mathbb{N}$ such that $\gcd(E, \varphi(n)) = 1$. [e.g., $\gcd(288, E) = 1$.]
   Most systems use $E = 65537$.
2. Solve $Ex \equiv 1 \pmod{\varphi(n)}$. [e.g., solve $95x \equiv 1 \pmod{288}$.]
   Extended Euclidean algorithm.
3. Compute x^E and y^D modulo n. [e.g., $86^{95} \pmod{n}$ and $307^{191} \pmod{n}$.]
   "Fast modular exponentation", uses method of repeated squaring.

1. How to find E such that $\gcd(E, \varphi(n)) = 1$

In our example:

$$n = pq = 17 \cdot 19 = 323, \qquad \varphi(n) = 16 \cdot 18 = 288,$$

and as the message recepient, you needed to find E such that $\gcd(288, E) = 1$.

For small n, this is easy: factor 288 and pick a number with no common prime factors.

In practice, n is too large to factor. But *any* prime that does not divide $\varphi(n) = (p-1)(q-1)$ will work.

Guessing and checking will yield a prime rather quickly.

A particularly nice choice of E would be:

- prime [makes it easier to verify that $\gcd(E, \varphi(n)) = 1$],
- of the form $2^n + 1$, because this is $1000 \cdots 001$ in binary.

The only primes of the form $2^n + 1$ also have the form $2^{2^k} + 1$, called Fermat primes.

The only known Fermat primes are 3, 5, 17, 257, 65537.

As such, in practice, $E = 2^{2^4} + 1 = 65537$ is usually used for encryption.

In the very slim chance that 65537 divides $\phi(n) = (p-1)(q-1)$, then go back and pick a new p and q.

2. How to solve $Ex \equiv 1 \pmod{\varphi(n)}$

Recall that we can solve an equation such as $Ex \equiv 1 \pmod{\varphi(n)}$ using the extended Euclidean algorithm.

Let's solve $95x \equiv 1 \pmod{288}$.

		288	95
	$288 = 1 \cdot 288 + 0 \cdot 95$	1	0
	$95 = 0 \cdot 288 + 1 \cdot 95$	0	1
$288 = 95 \cdot 3 + 3$	$3 = 1 \cdot 288 - 3 \cdot 95$	1	-3
$95 = 3 \cdot 31 + 2$	$2 = 1 \cdot 95 - 31 \cdot 3$	-31	94
$3 = 2 \cdot 1 + 1$	$1 = 1 \cdot 3 - 1 \cdot 2$	32	-97

We conclude that:

$$\gcd(288, 95) = 1 = 288(32) + 95(-97).$$

From this, we can solve

$$95x \equiv 1 \mod 288, \qquad \Longrightarrow \qquad x = -97 \equiv 191 \pmod{288}.$$

The Euclidean algoritm take at most $2 \log_2 x$ steps (rows).

So even for numbers $x \approx 10^{200}$, this is only ≤ 1329 steps.

3. Computing x^E and y^D modulo $n = pq$.

Even for our small example, we encountered $307^{191} \approx 1.101 \times 10^{475}$.

Though a computer can easily handle this, and reduce it modulo 323, this quickly becomes infeasible for y^D when $y, D \approx 10^{200}$.

If $x = \left\lfloor \sqrt{2} \cdot 10^{185} \right\rfloor$ and $E = \left\lfloor \sqrt{3} \cdot 10^{180} \right\rfloor$, then computing x^E requires over 10^{180} multiplications.

The universe is only $\approx 4.4 \times 10^{17}$ seconds old.

Goal

Compute $x^E \pmod{n}$ is at most $2 \log_2 E$ steps.

For the example above, this would require $2 \log_2 E \approx 1198$ steps.

3. Fast modular exponentiation

Let's compute $86^{95} \pmod{323}$. First, we write the exponent in base 2:

$$95 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^3 + 1 \cdot 2^3 + 1 \cdot 2^0 = 1011111_2.$$

Next, we can write

$$86^{95} = 86^{64+16+8+4+2+1} = 86^{64} 86^{16} 86^{8} 86^{4} 86^{2} 86^{1}.$$

Note that $86^2 \equiv 290 \pmod{323}$, and successive powers are:

4. $86^4 = (86^2)^2 \equiv 290^2 \equiv 120 \pmod{323}$,
8. $86^8 = (86^4)^2 \equiv 120^2 \equiv 188 \pmod{323}$,
16. $86^{16} = (86^8)^2 \equiv 188^2 \equiv 137 \pmod{323}$,
32. $86^{32} = (86^{16})^2 \equiv 137^2 \equiv 35 \pmod{323}$,
64. $86^{64} = (86^{32})^2 \equiv 35^2 \equiv 256 \pmod{323}$,

$$86^{95} = 86^{64} 86^{16} 86^{8} 86^{4} 86^{2} 86^{1} = 256 \cdot \overbrace{137 \cdot \underbrace{188 \cdot \overbrace{120 \cdot \underbrace{290 \cdot 86}_{=69}}^{=205}}_{=103}}^{=222} \equiv 307 \pmod{323}.$$

This is called the method of repeated squaring, and requires *at most* $2\log_2(E)$ steps. Clearly, things are (slighly) easier using $E = 65537 = 1000\cdots0001_2$.