Lecture 3 – Towards public key cryptography

Chloe Martindale

2023

In previous lectures we have seen how to set up secure communication *given a shared secret value.* In the modern world, you are attempting to communicate securely with many different parties: Servers on the other side of the world, family in another country, companies, governments, hospitals, the list goes on. So how can we use mathematics to share a secret value cheaply, easily, and without ever meeting? This is the premise of *public key cryptography*, and this lecture will build up the mathematical foundations necessary to understand some ways in which this is done in practise.

The most famous historical example of a shared secret value is the Caesar cipher. This was an encryption method used in Ancient Rome which just shifted all the letters of the alphabet by a given secret number. For example, if you shift LOVE CRYPTO by 5, you get QTAJ HWDUYT. This is not a great system as there are only 26 options for a shift (one of which is a shift of 0 and amounts to zero encryption). However, consider what would happen if I split my 10-letter message into two 5-letter blocks, and applied a Caesar shift on each (these can be different shifts). Then you already get 26^2 options - and of course we can split a long message into n blocks and get 26^n. This is the idea behind the modern term blockcipher that we saw in Week 1.

There are some other neat tricks hidden in this historical cipher. Look back at the example above: What do we do when we want to shift Y by 5? We cycle back around to A, so the intermediate letters look like Y–ZABC–D, so that every letter can be shifted. When working with computers, we are not given letters A-Z to encrypt but bit strings, but exactly the same idea applies. Consider shifting a message made up of the numbers $0, \ldots, 9$, by 5. If we want to shift, for example, number 8, then we get 3 via the intermediate values 8–9012–3. This should be a familiar concept: It is the concept behind a clock. It is also the concept behind *modular arithmetic.*

3.1 Modular arithmetic

Arithmetic should be thought of as the basic mathematical operations such as addition, subtraction, multiplication, and division. This is something with which you are very familiar with in the sets $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$, but there are many more ways of constructing sets of numbers on which there exist consistent arithmetic laws. The arithmetic of a clock is especially interesting because we can construct a consistent set of arithmetic laws on the *finite* set of hours.

Let us start by studying 'clock addition'. If you add 4 hours to 10 o'clock, then you get 2 o'clock (rather than 14 o'clock, since we will work here with the 12-hour clock). The way that we will write this is:

$$10 + 4 \equiv 2 \pmod{12}.$$

The $\equiv$ sign should be read as 'is equivalent to', and the notation $\pmod{12}$ tells us that we should reset when we get to the number 12, or if you like that our day is split into 12 hours.

'Clock subtraction' works in much the same way. If you subtract 6 hours from 1 o'clock, then you get 7 o'clock. The way that we will write this is:

$$1 - 6 \equiv 7 \pmod{12}.$$

'Clock multiplication' can be thought of just as repeated addition, so can as be defined in a natural way. For example,

$$5 \times 3 = 5 + 5 + 5 \equiv 3 \pmod{12}.$$

Division is a little more complicated, so we will return to that later.

A natural question that arises when studying clock arithmetic is: what if the day was not split into sets of 12 hours, but some other number, like 7? We can of course set up addition, subtraction, and multiplication $\pmod{7}$ in just the same way as $\pmod{12}$. Formally, we define the notation $\equiv$ and $\pmod{n}$ as follows.

Definition 3.1. Let $n \in \mathbb{Z}_{>1}$ and let $a, b \in \mathbb{Z}$. We say that

$$a \equiv b \pmod{n}$$

if there exists $k \in \mathbb{Z}$ such that $a = b + kn$.

We refer to basic arithmetic $\pmod{n}$ as *modular arithmetic*. Suppose now that we want to compute $10 \times 11 \pmod{12}$. We would like to find $a \in \mathbb{Z}$ such that $1 \leq a \leq 12$ and $10 \times 11 \equiv a \pmod{12}$. One way to do this is to first compute $10 \times 11 = 110$, and then divide 110 by 12 and take a to be the remainder. Try to prove for yourself that this will give the right answer.

Finally, let us turn to division. Suppose that you want to divide 3 by 4 on our 7-hour clock. It turns out that the best way to think of this is as 3×4^{-1}– we already have a notion of multiplication (and of 3), so it remains to understand the notion of inverses:

Definition 3.2. Let $a \in \mathbb{Z}$ and $n \in \mathbb{Z}_{>1}$. If there exists $b \in \mathbb{Z}$ such that

$$ab \equiv 1 \pmod{n}$$

then we say that $b \pmod{n}$ is the *inverse* of $a \pmod{n}$.

Notice the 'if there exists' part of this definition. Consider $a = n = 12$. No matter how many multiples of 12 you take, you are always going to land back at the 12 o'clock position on the clock, or more formally, for every $b \in \mathbb{Z}$ we have that $12b \equiv 12 \pmod{12}$, so in particular 12 has no inverse mod 12. In fact, since $12 \equiv 0 \pmod{12}$, this isn't so surprising, since we are used to the idea of 0 having no inverse. There are however other numbers by which we cannot divide $\pmod{12}$. Consider $a = 6$ and $n = 12$. For every $b \in \mathbb{Z}$ we have that either $6b \equiv 6 \pmod{12}$ or $6b \equiv 0 \pmod{12}$. So $6 \pmod{12}$ also has no inverse. When does an integer mod n have an inverse?

To understand when the inverse exists, we first need to understand in which situations the inverse of $a \pmod{b}$ exist for any a and b. Let's look at a couple of examples.

Examples

- The inverse of $4 \pmod 7$ is $2 \pmod 7$ because $4 \cdot 2 \pmod 7 \equiv 1 \pmod 7$.
- $4 \pmod 8$ has no inverse because for every $n \in \mathbb{Z}$ we know that
  $$4 \cdot n \pmod 8 \in \{0 \pmod 8, 4 \pmod 8\},$$
  so in particular there does not exist any $n \pmod 8$ such that $4 \cdot n \equiv 1 \pmod 8$.
- Exercise: generalize the above example. That is, show that if m and n are not coprime then m does not have an inverse mod n.

In fact, the above exercise is also true in the reverse. That is, $a \pmod b$ is invertible if and only if a and b are coprime. The exercise above gives the 'only if', but what about the 'if'? For this we need *Euclid's algorithm.*[1]

3.2 Euclid's algorithm

Euclid's Algorithm

Require: a and $b \in \mathbb{Z}_{>0}$; without loss of generality suppose that $a \geq b$.
Ensure: $d = \gcd(a, b)$.

```
Set $r_0 = a$, $r_1 = b$, and $i = 1$.
while $r_i \neq 0$ do
    $i \leftarrow i + 1$.
    Compute[2] the unique $m_i$ and $r_i \in \mathbb{Z}$ such that $0 \leq r_i < r_{i-1}$ and
        $$r_{i-2} = m_i \cdot r_{i-1} + r_i.$$
end while
return $r_i$
```

Corollary 3.1 (Euclid's corollary[3]). *Let a and b be integers. If $d = \gcd(a, b)$ then there exist $m, n \in \mathbb{Z}$ such that*

$$am + bn = d.$$

[1] Most likely not due to Euclid, but Euclid wrote about it.

[2] This is called *division-with-remainder.*

[3] Often referred to just as Euclid's algorithm.

Proof. This follows from Euclid's algorithm just by solving the series

$$\{r_{i-2} = m_i \cdot r_{i-1} + r_i\}_{2 \leq i \leq k}$$

of simultaneous equations occuring in Euclid's algorithm for $r_0 = a$, $r_1 = b$, and $r_k = d$. □

Now we have a new method to compute the inverse of a mod b, as long as a and b are coprime: Use Euclid's algorithm to compute m and n such that $am + bn = 1$. Then, modulo b, we have

$$am \equiv 1 \pmod{b},$$

or in other words m is the inverse of a mod b.

Example.
Let's see an example of how to use Euclid's algorithm to compute an inverse. Suppose that you want to compute the inverse of 11 $\pmod{17}$.

Run Euclid's algorithm with $r_0 = 17$ and $r_1 = 11$:

$$\begin{aligned} r_0 &= 17 \\ r_1 &= 11 \\ r_2 &= 17 - 1 \cdot 11 = 6 \\ r_3 &= 11 - 1 \cdot 6 = 5 \\ r_4 &= 6 - 1 \cdot 5 = 1. \end{aligned}$$

Then reverse engineer the algorithm to get:

$$1 = r_4 = r_2 - r_3 = (r_0 - r_1) - (r_1 - r_2) = r_0 - 2r_1 + r_2 = 2r_0 - 3r_1.$$

In other words,

$$2 \cdot 17 - 3 \cdot 11 = 1,$$

so in particular

$$-3 \cdot 11 \equiv 1 \pmod{17},$$

so the inverse of 11 $\pmod{17}$ is $-3 \equiv 14 \pmod{17}$.

3.3 Groups

Sets of integers equipped with addition modulo n are examples of *groups*. Groups are central to the construction of public-key cryptography–next week we'll see how to define a secure key exchange based on a group with certain properties. This key exchange (the Diffie-Hellman key exchange) is fundamental in every widely used protocol on the internet today (TLS 1.3, Signal, etc). Here we just give the definition and some examples of groups to familiarise ourselves with the concept.

Definition 3.3. Let G be a set and $* : G \times G \to G$ a map that takes pairs of elements in G to a single element of G. We say that $(G, *)$ is a *group* or that G *defines a group under* $*$ if the following *group axioms* are satisfied:

(G1) There exists $e \in G$ such that for every $g \in G$, $e * g = g * e = g$. *(G has an identity).*

(G2) For every $g \in G$, there exists $h \in G$ such that $g * h = h * g = e$. *(every element has an inverse).*

(G3) For every $a, b, c \in G$, $(a * b) * c = a * (b * c)$. *($*$ is associative).*

We often just say 'G is a group' instead of '$(G, *)$ is a group' if the author considers it 'obvious' which operation $*$ should be.

Examples

- For any integer $n \geq 2$, the set $\{0 \pmod{n}, 1 \pmod{n}, \ldots, n-1 \pmod{n}\}$ is a group under $+ \pmod{n}$.
- For any integer $n \geq 2$, the set $\{0 \pmod{n}, 2 \pmod{n}, \ldots, n-1 \pmod{n}\}$ is *not* a group under multiplication $\pmod{n}$. Reason: $0 \pmod{n}$ has no inverse (c.f. (G2)).
- For any composite integer $n \geq 2$, the set $\{1 \pmod{n}, 2 \pmod{n}, \ldots, n-1 \pmod{n}$ is *not* a group under multiplication $\pmod{n}$. Reason: n is composite, so there exists $0 \neq a \pmod{n}$ such that $\gcd(a, n) \neq 1$, which we proved above was not invertible.
- For any prime p, the set $\{1 \pmod{p}, \ldots, p-1 \pmod{p}\}$ is a group under multiplication $\pmod{p}$.

Sets of integers $\pmod{p}$ and $\pmod{n}$ will return again and again, so let us introduce some notation for this. From now on, we will write

$$\mathbb{Z}/n\mathbb{Z} = \{0 \pmod{n}, \ldots, n-1 \pmod{n}\}$$

and $(\mathbb{Z}/n\mathbb{Z})^*$ for the set of invertible elements of $\mathbb{Z}/n\mathbb{Z}$.

Note that for a prime p, that means that

$$(\mathbb{Z}/p\mathbb{Z})^* = \{1 \pmod{p}, \ldots, p-1 \pmod{p}\};$$

we saw in our examples above that $(Z/p\mathbb{Z})^*$ is a group under multiplication $\pmod{p}$. This group turns out to be very useful for us, partly because it is *cyclic* for any prime p. That is, there exists a $g \pmod{p}$ such that

$$(\mathbb{Z}/p\mathbb{Z})^* = \{g \pmod{p}, g^2 \pmod{p}, \ldots, g^{p-1} \pmod{p}\}.$$

Another word for this is to say that g *generates* $(\mathbb{Z}/p\mathbb{Z})^*$.

Definition 3.4. Let $(G, *)$ be a group. We say that $g \in G$ *generates* G if

$$G = \{g, g * g, g \underbrace{* \cdots *}_{|G| \text{ times}} g\}.$$

We then call g a *generator.*

For example, if $p = 5$ it turns out that we can choose $g = 2$:

$$(\mathbb{Z}/5\mathbb{Z})^* = \{2 \pmod 5, 4 \equiv 2^2 \pmod 5, 3 \equiv 2^3 \pmod 5, 1 \equiv 2^4 \pmod 5\}.$$

In this example, you see that the last element in the list, $g^{p-1} \pmod p$, is $1 \pmod p$. In fact, this is not a coincidence: This follows from *Fermat's Little Theorem.*

3.4 Fermat's Little Theorem

Fermat's Little Theorem comes up again and again when dealing with modular arithmetic. It is a useful identity in its own right, but it is also another way of computing inverses mod n, as well as fundamental in the construction of RSA.

Definition 3.5. Let $n \in \mathbb{Z}_{>0}$. The *Euler φ-function* of n is

$$\varphi(n) := \#\{m \in \mathbb{Z} : 0 < m < n, \gcd(m, n) = 1\}.$$

1. $\varphi(7) = \#\{1, 2, 3, 4, 5, 6\} = 6.$
2. $\varphi(8) = \#\{1, 3, 5, 7\} = 4.$

Exercise: prove that for $p \neq q$ prime,

$$\varphi(p) = p - 1$$

and

$$\varphi(pq) = (p - 1)(q - 1).$$

Theorem 3.2 (Fermat's Little Theorem). *For every $a \in \mathbb{Z}$ and squarefree $n \in \mathbb{Z}_{>1}$,*

$$a^{\varphi(n)+1} \equiv a \pmod n.$$

Note in particular that is $n = p$ is prime, then this identity becomes

$$a^{p-1} \equiv 1 \pmod p,$$

which is what we observed in the example above. This also means that for any a coprime to p, the inverse of a can be computed by repeated exponentiation as $a^{p-2} \pmod p$.

3.5 Sun-Tzu's Remainder Theorem

There are many surprising constructions and consequences of modular arithmetic, and we end this chapter with a seminal theorem in modular arithmetic which turns out to be a key tool in cryptanalysis.

Theorem 3.3 (Sun-Tzu's Remainder Theorem (SRT)[4])**.** *Given coprime* $n, m \in \mathbb{Z}_{>1}$ *and* $a, b \in \mathbb{Z}$ *there exist* $c, d \in \mathbb{Z}$ *such that*

$$cm + dn = 1 \tag{3.1}$$

and

$$x = bcm + adn \pmod{mn}$$

is the only number $\pmod{mn}$ *such that both*

$$x \equiv a \pmod{m} \quad \textit{and} \quad x \equiv b \pmod{n}.$$

You may have seen this before in a basic number theory course or a group theory course for example: with some mathematical machinery it is quick to prove. We won't prove uniqueness now but we will check that the given construction is valid.

Proof of existence (constructive). As $\gcd(n, m) = 1$, by Euclid's algorithm there exist $c, d \in \mathbb{Z}$ such that

$$cm + dn = 1. \tag{3.2}$$

We claim that

$$x = bcm + adn \pmod{mn}$$

will work. We first check mod n. Note that $cm = 1 - dn$ by (3.2). So

$$x = b(1 - dn) + adn \equiv b \pmod{n}.$$

Similarly

$$x = bcm + a(1 - cm) \equiv a \pmod{m}.$$

□

Example
Now suppose that you are given the equations

$$x \equiv 4 \pmod{17}$$

and

$$x \equiv 3 \pmod{11}$$

and you want to find the $x \pmod{17 \cdot 11}$ that reduces mod 17 and 11 to these values. We already saw in our example of computing inverses using Euclid's algorithm that

$$2 \cdot 17 - 3 \cdot 11 = 1.$$

[4] Most textbooks refer to this as the 'Chinese Remainder Theorem'. It is most likely not due to Sun-Tzu, but Sun-Tzu wrote about it.

Now using SRT, we get

$$x = 2 \cdot 17 \cdot 3 - 3 \cdot 11 \cdot 4 = 2 \cdot 3(17 - 2 \cdot 11) = -30.$$

To get a positive representative, we can just add $17 \cdot 11 = 187$, so

$$x \equiv 157 \pmod{17 \cdot 11}.$$