Lecture Notes in Cryptography Engineering

Óscar Pereira[1]

1st December 2020

[1] oscar@di.uminho.pt.

Contents

5 RSA

5.1 A Gentle Introduction

Pierre de Fermat's so-called little theorem tells us that for any integer a, and prime p, we have $a^p \equiv a \pmod{p}$. Now, if $a \not\equiv 0 \pmod{p}$, or equivalently, if a is not a multiple of p (symbolically, $p \nmid a$, read "p does not divide a"), the extended Euclidean Algorithm tells us that a has a modular inverse modulo p—indeed, it shows us how to compute it. That is, there exists an integer b such that $ab \equiv 1 \pmod{p}$. And so, if we multiply both sides of the first congruence above by b, we obtain $a^{p-1} \equiv 1 \pmod{p}$—valid for any a that is **not** a multiple of p.

Now, as your favourite book on abstract algebra or number theory will happily explain to you, when the modulus is not prime, things get a bit more complicated. In particular, we have to use *Euler's ϕ function*, also called the *totient* function. For a positive integer n, $\phi(n)$ equals the number of integers i such that $1 \leq i < n$ and $gcd(i, n) = 1$. This allows to generalise Fermat's little theorem as follows:

Theorem 5.1 (Euler's theorem). *If a and n are integers such that $n > 0$ and $gcd(a, n) = 1$, then the following holds:*

$$a^{\phi(n)} \equiv 1 \pmod{n} \tag{5.1}$$

Remark 5.2. If n is prime, then $\phi(n) = n - 1$ and we get back Fermat's (little) theorem. △

Most books explain RSA with the totient, but we can simplify things somewhat.[1] This is because the modulus used in RSA, though not a prime, has the (relatively) simple form of a product of two primes: $n = pq$. Now, just as above, we are trying to find a value t such that $x^t \equiv 1 \pmod{n}$, for almost all values of x. We already know we can set $t = \phi(n)$, in which case the condition holds for all x relatively prime to n. But we can do better. For $x^t \equiv 1 \pmod{n}$ means that $n \mid (x^t - 1)$, and as $n = pq$, this implies that $p \mid (x^t - 1)$ and $q \mid (x^t - 1)$—or equivalently, $x^t \equiv 1 \pmod{p}$ and $x^t \equiv 1 \pmod{q}$ respectively. Note that, as p and q are both primes, and hence also co-prime, we have $lcm(p,\ q) = pq = n$.[2] And so the *converse* also holds: if $x^t \equiv 1 \pmod{p}$ and $x^t \equiv 1 \pmod{q}$, then also $x^t \equiv 1 \pmod{n}$.

Fermat's theorem shows that to have $x^t \equiv 1 \pmod{p}$, we must have $(p-1) \mid t$. And similarly, to have $x^t \equiv 1 \pmod{q}$, we must have $(q-1) \mid t$. The smallest t for which this holds is $t = lcm(p-1,\ q-1)$, which you learned back in high school to compute as:

$$lcm(p-1,\ q-1) = \frac{(p-1)(q-1)}{gcd(p-1,\ q-1)} \tag{5.2}$$

[1] See Ferguson and Schneier ([5], §13).

[2] lcm stands for *least common multiple*; gcd for *greatest common divisor*.

So we set $t = lcm(p-1,\ q-1)$, for which $x^t \equiv 1 \pmod{n}$ holds—for *almost* all values of x. Which values of x should be excluded? Those for which either $x^t \equiv 1 \pmod{p}$ or $x^t \equiv 1 \pmod{q}$ fails to hold. Fermat's theorem says that these are the multiples of p and q, and as here we are working with modulo n, we want those multiples that are between 0 and $n-1$. So we have the multiples of p: $p, 2p, \ldots, (q-1)p$—so we have $q-1$ multiples of p. For q, we have $q, 2q, \ldots (p-1)q$—so $p-1$ multiples of q. And there is 0 (multiple of all numbers), so in total we have $(q-1)+(p-1)+1 = p+q-1$. For a large enough n, this is a very small fraction of the total number of values $0, \ldots, n-1$.

Remark 5.3. For $n = pq$, $\phi(n)$ is equal to the number of values from 1 to $n-1$ (so $pq-1$ values), minus the number of multiples of either p or q in that range. As we are excluding the 0, this is just $(q-1)+(p-1) = p+q-2$. And so we have $\phi(n) = (pq-1)-(p+q-2) = pq-p-q+1 = (p-1)(q-1)$. Comparing with (5.2), we see that $t \mid \phi(n)$, and so $x^{\phi(n)} \equiv 1 \pmod{n}$ also holds. △

RSA. This allows us to compute e, d such that $ed \equiv 1 \pmod{t}$, which means that ed can be written as $ed = tk+1$, for some value of k. Thus, if we cipher a message m as m^e, and decipher the cryptogram doing $(m^e)^d$, we obtain $m^{ed} = m^{tk+1} = (m^t)^k m \equiv m \pmod{n}$.

References

1. **Arora**, Sanjeev and Boaz **Barak** (2009). *Complexity Theory: A Modern Approach*. Cambridge, UK: Cambridge University Press. ISBN: 978-0-521-42426-4. Not cited.

2. **Bellare**, Mihir, Ran **Canetti**, and Hugo **Krawczyk** (1996). *Keying Hash Functions for Message Authentication*. In Koblitz, N. (ed.), Advances in Cryptology — CRYPTO '96. Berlin, Heidelberg: Springer, pp. 1–15. Not cited.

3. **Bellovin**, Steve (2009). Email sent to `comp.risks` (Risks Digest), 25.71, June 6, 2009. It used to be availabe at https://groups.google.com/forum/#!topic/comp.risks/4V3cECtN_vQ (last checked at 2016-05-09), but that URL is no longer accessible. Not cited.

4. **Benoit**, Viguier (2016). https://gitlab.insa-rennes.fr/bviguier/mri/blob/fd9d822a6cb4eea2bce422701a2b011811d78cb1/MRI/tikz/CBC-MAC.tex Not cited.

5. **Ferguson**, Niels and Bruce **Schneier** (2003). *Practical Cryptography*. Indianapolis: Wiley Publishing. ISBN: 0-471-22357-3. Cited on page 25.

6. **Katz**, Jonathan and Yehuda **Lindell** (2015). *Introduction to Modern Cryptography*, 2nd edition. Boca Raton, FL, U.S.: CRC Press. ISBN: 978-1-4665-7027-6. Not cited.