Notes on Chinese Remainder Theorem

Davis Yang Email:yxyang@berkeley.edu

September 29, 2016

In my discussion today we tried to go over Chinese Remainder Theorem and how it can be applied in proving the correctness of RSA. I went over time a little bit and didn't finish, so I decided to write it up. This note should provide an introduction to Chinese Remainder Theorem, as well as the basic proof.

1 Let's start from RSA

Let's first do a brief recap of RSA to motivate the need for Chinese Remainder Theorem. Feel free to skip this section and jump to the theorem itself.

In RSA encryption, we have $N = pq$ where p and q are two prime numbers. Then, the public key e and private key d are selected such that $e \equiv d^{-1} \pmod{(p-1)(q-1)}$. Now, suppose Alice wants to send Bob a secure message, Bob first sent the public key (e, N) to Alice. Then, for message x, Alice sends $x^e \pmod{N}$ to Bob, and once Bob received the encrypted message y, he decrypts the message by computing $y^d \pmod{N}$.

An obvious question now is, is this algorithm even correct? Does raising the received message y to the power of d guarantee you the original message back? This led us to the proof of RSA's correctness. That is, we want to show $(x^e)^d \equiv x^{de} \equiv x \pmod{N}$.

Note that since e and d are inverses modulo $(p-1)(q-1)$, $de \equiv 1 \pmod{(p-1)(q-1)}$. Therefore, we have:

$$x^{de} \equiv x^{k(p-1)(q-1)+1} \pmod{pq}$$

Now, according to Fermat's Little Theorem, since p is prime and $1 \leq x \leq p$, we have $x^{p-1} \equiv 1 \pmod{p}$. Therefore, we have:

$$\begin{aligned} x^{k(p-1)(q-1)+1} &\equiv x \cdot x^{k(p-1)(q-1)} \\ &\equiv x \cdot 1^{k(q-1)} \\ &\equiv x \pmod{p} \end{aligned}$$

And similarly:

$$\begin{aligned} x^{k(p-1)(q-1)+1} &\equiv x \cdot x^{k(p-1)(q-1)} \\ &\equiv x \cdot 1^{k(p-1)} \\ &\equiv x \pmod{q} \end{aligned}$$

But we are not done yet! We want to show that $x^{(p-1)(q-1)+1} \equiv x \pmod{pq}$, but we only show that it works for $\pmod{p}$ **and** $\pmod{q}$. How do we proceed? The proof in lecture note proceeded by subtracting x from both sides of the equation, which is actually the proof of a special case of the **Chinese Remainder Theorem**, which will be described in the next section.

2 The Chinese Remainder Theorem

Let's start with the theorem itself. An example is given at the end.

Chinese Remainder Theorem. if $n_1, n_2...n_k$ are pairwise coprime, there exists a unique $x \pmod N$ s.t.

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &..... \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

where $N = n_1 n_2 ... n_k$

Proof. The proof of the theorem consists of 2 part: existence and uniqueness

Uniqueness: Proof by contradiction. Suppose that for all any $\exists x_1 \neq x_2$ s.t. $\forall 1 \leq i \leq k, x_1 \equiv x_2 \equiv a_i \pmod{n_i}$. Then $\forall 1 \leq i \leq k, x_1 - x_2 \equiv 0 \pmod{n_i}$. Then since $n_1, n_2...n_k$ are coprime, they don't share any common factor together, which means $x_1 - x_2 \equiv 0 \pmod{n_1 n_2 ... n_k}$. This implies that x_1 and x_2 differ by at least N, contradiction.

Note that the uniqueness proof is very similar to the bijection proof in proving Fermat's Little Theorem!

Existence: We give a constructive proof. We start with $k = 2$, i.e. there are only 2 numbers, n_1, n_2, and we want to find an x s.t. $x \equiv a_1 \pmod{n_1}$ and $x \equiv a_2 \pmod{n_2}$.

Since we know n_1 and n_2 are coprime, running extended-gcd algorithm on n_1 and n_2 will give us m_1 and m_2 [1] such that:

$$m_1 n_1 + m_2 n_2 = 1 \tag{1}$$

Now, construct x s.t.:

$$x \equiv a_1 m_2 n_2 + a_2 m_1 n_1 \tag{2}$$

To find the remainder of x modulo n_1, we have:

$$\begin{aligned} x &\equiv a_1 m_2 n_2 + a_2 m_1 n_1 \\ &\equiv a_1(1 - m_1 n_1) + a_2 m_1 n_1 \\ &\equiv a_1 - a_1 m_1 n_1 + a_2 m_1 n_1 \\ &\equiv a_1 \pmod{n_1} \end{aligned}$$

Where the first step follows from equation (1). Similarly, we have:

$$\begin{aligned} x &\equiv a_1 m_2 n_2 + a_2 m_1 n_1 \\ &\equiv a_1 m_2 n_2 + a_2(1 - m_2 n_2) \\ &\equiv a_1 m_2 n_2 + a_2 - a_2 m_2 n_2 \\ &\equiv a_2 \pmod{n_2} \end{aligned}$$

[1] Review the lecture note on extended-gcd algorithm if this doesn't make sense.

Which shows that x indeed satisfies the conditions! To extend the case from $k = 2$ to any arbitrary k, you can either follow a proof by induction, or generalize the above construction process. [2] □

So, suppose we want to find an x such that[3]:

$$\begin{aligned} x &\equiv 3 \pmod 5 \\ x &\equiv 4 \pmod 8 \end{aligned}$$

The extended gcd algorithm gives us:

$$5 \cdot 5 + 8 \cdot (-3) = 1$$

So according to the construction, we have:

$$x = 3 \cdot 8 \cdot (-3) + 4 \cdot 5 \cdot 5 = -72 + 100 = 28$$

Which does satisfy the condition.

3 Back To RSA

Let's go back to where we started from. In the proving RSA's correctness, we found that

$$\begin{aligned} x^{k(p-1)(q-1)+1} &\equiv x \pmod p \\ x^{k(p-1)(q-1)+1} &\equiv x \pmod q \end{aligned} \tag{3}$$

Applying the constructive process, assuming that extended GCD algorithm gives us $m_1p + m_2q = 1$, then the number that satisfies (3) would be $xm_2q + xm_1p \equiv x(m_1p + m_2q) \equiv x$. This means that $x^{k(p-1)(q-1)+1} \pmod{pq}$ is indeed x. That completes our proof about RSA's correctness.

[2] See `https://en.wikipedia.org/wiki/Chinese_remainder_theorem` for a complete discussion of the theorem and its proof.

[3] actually, you saw this example back in discussion 4(b) problem 5