COMP 8920: Cryptography	Winter 2025

Lecture 17 — March 11, 2025

Prof. Curtis Bright — *Scribe: Rajat Yadav*

1 Public Key Cryptography and RSA

The idea of public key cryptography was proposed by Diffie-Hellman in 1976, and in 1977 the RSA cryptosystem was proposed and is still in use today. Bob's encryption function e_k should be efficient to compute, but the inverse e_k^{-1} should only be efficient to compute for Bob, who has some secret information that enables him to compute it efficiently.

In RSA, encryption is done via $f : \mathbb{Z}_n \to \mathbb{Z}_n$ of the form $x \mapsto x^b \bmod n$ where $n = p \cdot q$ for primes p, q. Computing f^{-1} is difficult if you just know n, but if you also know p, q then the inverse can be computed efficiently.

2 Number Theory Background

The Euclidean Algorithm finds the greatest common divisor (gcd) of two numbers, and the extended Euclidean algorithm (EEA) allows us to compute the inverse of a number $a^{-1} \bmod n$. Recall that $a^{-1} \in \mathbb{Z}_n$ is defined so $a \cdot a^{-1} \equiv 1 \pmod n$, and it exists when $\gcd(a, n) = 1$.

$\mathbb{Z}_n$ is the numbers mod$n$. $\mathbb{Z}_n^*$ is the numbers mod$n$ with an inverse. $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n : \gcd(a, n) = 1\}$; its size $|\mathbb{Z}_n^*| = \phi(n)$ where ϕ is Euler's phi function.

2.1 Euclidean Algorithm

Input: $a, b \in \mathbb{Z}^+$

Set $r_0 := a, r_1 := b$, and write:

$$r_0 = q_1 r_1 + r_2 \quad \text{where } 0 \le r_2 < r_1 \tag{1}$$
$$r_1 = q_2 r_2 + r_3 \tag{2}$$
$$\vdots \tag{3}$$
$$r_{m-2} = q_{m-1} r_{m-1} + r_m \tag{4}$$
$$r_{m-1} = q_m r_m \tag{5}$$

Output: r_m

Note:

$$
\begin{aligned}
\gcd(a,b) &= \gcd(r_0, r_1) && (6)\\
&= \gcd(r_0 - q_1 r_1, r_1) && (7)\\
&= \gcd(r_2, r_1) && (8)\\
&= \gcd(r_1, r_2) && (9)\\
&= \gcd(r_1 - q_2 r_2, r_2) && (10)\\
&= \gcd(r_3, r_2) && (11)\\
&\vdots && (12)\\
&= \gcd(r_{m-1}, r_m) && (13)\\
&= \gcd(r_m, 0) = r_m && (14)
\end{aligned}
$$

2.2 Extended Euclidean Algorithm

We can now tell if $a^{-1} \bmod n$ exists by computing $\gcd(a, n)$, but how do we find a^{-1}? We run the same computation but store some extra information, storing the r_i as a weighted sum of a and b. Note it is easy to write r_0, r_1 as a sum of a and b:

$$
\begin{aligned}
r_0 &= 1 \cdot a + 0 \cdot b = s_0 \cdot a + t_0 \cdot b && (15)\\
r_1 &= 0 \cdot a + 1 \cdot b = s_1 \cdot a + t_1 \cdot b && (16)
\end{aligned}
$$

To form r_2, subtract $q_1 \cdot r_1$ from r_0:

$$r_2 = r_0 - q_1 r_1 = 1 \cdot a + (-q_1) \cdot b = s_2 \cdot a + t_2 \cdot b \qquad (17)$$

Proceeding in this way by subtracting $q_j \cdot r_{j-1}$ from r_{j-2}, we find $r_j = s_j a + t_j \cdot b$ where s_j and t_j are integers found via:

$$
\begin{aligned}
t_j &= t_{j-2} - q_{j-1} \cdot t_{j-1} && (18)\\
s_j &= s_{j-2} - q_{j-1} \cdot s_{j-1} && (19)
\end{aligned}
$$

with $(s_0, t_0) = (1, 0)$ and $(s_1, t_1) = (0, 1)$.

Then $\text{EEA}(a, b)$ returns (r_m, s_m, t_m).

Why is this useful? To compute $a^{-1} \bmod n$, run $\text{EEA}(a, n)$ which gives s, t with:

$$1 = s \cdot a + t \cdot n \equiv s \cdot a \pmod{n} \qquad (20)$$

So $a^{-1} \bmod n = s$.

2.3 Chinese Remainder Theorem (CRT)

The Chinese Remainder Theorem is a method of solving systems of congruences of the form:

$$x \equiv a_1 \pmod{m_1} \tag{21}$$
$$x \equiv a_2 \pmod{m_2} \tag{22}$$
$$\vdots \tag{23}$$
$$x \equiv a_r \pmod{m_r} \tag{24}$$

where all m_i's are pairwise coprime. The CRT says this has a unique solution modulo $M = m_1 \times \cdots \times m_r$, and CRT gives a formula for x.

Consider the "projection" $\chi : \mathbb{Z}_M \to \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_r}$:

$$\chi(x) = (x \bmod m_1, \ldots, x \bmod m_r) \tag{25}$$

CRT essentially says χ is a bijection and gives a formula for χ^{-1}.

For $1 \leq i \leq r$, define:

$$M_i = M/m_i = \prod_{j=1, j\neq i}^{r} m_j \in \mathbb{Z} \tag{26}$$

Also note $\gcd(M_i, m_i) = 1$ since $\gcd(m_j, m_i) = 1$ for all $j \neq i$. Thus, we can define $y_i = M_i^{-1} \bmod m_i$ and find y_i with $\text{EEA}(m_i, M_i)$.

Now define $\rho : \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_r} \to \mathbb{Z}_M$ by:

$$(a_1, \ldots, a_r) \mapsto \sum_{i=1}^{r} a_i M_i y_i \bmod M \tag{27}$$

The system of congruences is equivalent to solving $\chi(x) = (a_1, \ldots, a_r)$, and the system has the general solution $x = \rho(a_1, \ldots, a_r)$.

First, we'll show $X = \rho(a_1, \ldots, a_r)$ is a solution of all congruences $x \equiv a_i \pmod{m_i}$. For concreteness, we'll just show $X \equiv a_1 \pmod{m_1}$, but the argument is the same for all m_i.

Note:

$$X = (a_1 M_1 y_1 + \sum_{i=2}^{r} a_i M_i y_i) \bmod M \tag{28}$$

Note $m_1 | M_i$ for all $2 \leq i \leq r$ (here $|$ means "divides"), so $M_i \equiv 0 \pmod{m_1}$ and $\sum_{i=2}^{r} a_i M_i y_i \equiv 0 \pmod{m_1}$. Also $y_1 = M_1^{-1} \bmod m_1$, so $M_1 y_1 \equiv 1 \pmod{m_1}$, and thus $X \equiv a_1 \pmod{m_1}$.

Now we just need to show that X is the only solution of $\chi(x) = (a_1, \ldots, a_r)$. We just showed this map χ is surjective (onto) since χ maps X to $(a_1, \ldots, a_r)$ for any values of a_i's. But the domain $\mathbb{Z}_M$ and codomain $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_r}$ are the same size, so χ must also be injective and hence a bijection, with $\chi(\rho(a_1, \ldots, a_r)) = (a_1, \ldots, a_r)$, i.e., $\rho = \chi^{-1}$.

For RSA, we'll work in $\mathbb{Z}_n$ where $n = p \cdot q$ for primes p, q. Since $\gcd(p, q) = 1$, CRT says $\mathbb{Z}_n$ is isomorphic to $\mathbb{Z}_p \times \mathbb{Z}_q$.