COMP 8920: Cryptography Winter 2025

Lecture 18 — March 13, 2025

Prof. Curtis Bright *Scribe: Rajat Yadav*

1 Group Theory and RSA

A group is an algebraic structure with a binary operation (e.g. multiplication) following:

- An identity exists (say 1)
- Every element x has an inverse x^{-1}
- Associativity holds $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

Examples include $(\mathbb{Z}, +)$, $(\mathbb{Q} \setminus \{0\}, \times)$, $(\mathbb{Z}_m^*, \times)$.

The order of a group element g is the smallest positive integer m for which $g^m = 1$.

Lagrange's Theorem

For a group G with n elements, the order of g divides n (written $g \mid n$) and g divides n means that n is a multiple of g, so there exists $k \in \mathbb{Z}$ with $g \cdot k = n$.

Note: $\mathbb{Z}_m^* = \{a \in \mathbb{Z}_m : \gcd(a, m) = 1\}$ is a group since $a^{-1} \bmod m$ exists when a and m are coprime and a^{-1} is found by $\mathrm{EEA}(a, m)$.

Note: $|\mathbb{Z}_m^*| = \phi(m)$, so Lagrange's Theorem says for all $b \in \mathbb{Z}_m^*$

$$b^{\phi(m)} \equiv 1 \pmod{m} \tag{1}$$

A special case is Fermat's Little Theorem: for all $b \in \mathbb{Z}_p^*$ for prime p

$$b^{p-1} \equiv 1 \pmod{p} \tag{2}$$

Also, $\mathbb{Z}_p^*$ is known as a cyclic group, meaning that all elements in $\mathbb{Z}_p^*$ can be written as a power of a generator α known as a primitive element having order $p - 1$, i.e.,

$$\mathbb{Z}_p^* = \{\alpha^i : 1 \leq i \leq p - 1\} \tag{3}$$

If $\beta \in \mathbb{Z}_p^*$ then $\mathrm{ord}(\beta) = (p - 1)/\gcd(p - 1, i)$ where $\beta = \alpha^i$. Thus, β is a primitive element when $i = \log_\alpha \beta$ is prime to $p - 1$. This means there are $\phi(p - 1)$ primitive elements in $\mathbb{Z}_p^*$.

There is no provably deterministic way to find a primitive element, but $\phi(p-1) \approx (p-1)/\log\log(p-1)$, so you can usually find one just by trying random $\beta \in \mathbb{Z}_p^*$. However, how to check if β is primitive? Computing $\beta^2, \beta^3, \beta^4, \ldots$ until you reach 1 would be very slow when p is large.

Theorem

If $p > 2$ is prime then $\alpha \in \mathbb{Z}_p^*$ is primitive if and only if $\alpha^{(p-1)/q} \not\equiv 1 \pmod p$ for all primes $q \mid (p-1)$. If we know the prime divisors of $(p-1)$, this provides an efficient test if α is primitive.

Proof

($\Rightarrow$): If α is primitive then $\alpha^i \bmod p \neq 1$ for any $i \in \{1, \ldots, p-2\}$ and $(p-1)/q$ is in $\{1, \ldots, p-2\}$ as $q > 1$ and $q < p-1$.

($\Leftarrow$): Suppose $\alpha^{(p-1)/q} \not\equiv 1 \pmod p$ for all primes $q \mid (p-1)$. Suppose α is not primitive, so it has order $d < p-1$. So, $\alpha^d \equiv 1 \pmod p$ and $d \mid (p-1)$ by Lagrange's Theorem. Since $d \mid (p-1)$ but $d \neq p-1$, $(p-1)/d$ is an integer > 1.

So $(p-1)/d$ has some prime divisor, say $q \mid (p-1)/d$. Rewriting this, we have $\exists k \in \mathbb{Z}$ such that $q \cdot k = (p-1)/d$ which implies $d \cdot k = (p-1)/q$. Since $\alpha^d \equiv 1 \pmod p$, raising this to the power k, we get $\alpha^{d \cdot k} = \alpha^{(p-1)/q} \equiv 1 \pmod p$, a contradiction. $\square$

The RSA Cryptosystem

Let $n = pq$ where p, q are distinct primes. Let $e = P = \mathbb{Z}_n$.

In practice, for security, p and q will be say 1024-bit primes.

Define the key set:

$$\mathcal{K} = \{(n, p, q, a, b) : ab \equiv 1 \mod \varphi(n)\}$$

For a key $k = (n, p, q, a, b)$, define the encryption and decryption functions as:

$$e_k(x) = x^b \mod n$$
$$d_k(x) = x^a \mod n$$

The public key is (n, b), while the private key is (p, q, a). None of these values, nor $\varphi(n)$, should be revealed, or RSA will be insecure.

An adversary does not know p, q, or $\varphi(n)$, so they cannot compute $a = b^{-1} \mod \varphi(n)$ using the Extended Euclidean Algorithm (EEA).

To compute an RSA key, you would select two large random primes p and q first, then multiply $n = pq$. Note this also allows us to find $\varphi(n)$, Since $\varphi(p, q) = \varphi(p)\varphi(q) = (p-1)(q-1)$. Select a random $b \in \mathbb{Z}_{\varphi(n)}^*$) then compute $a = b^{-1} \mod \varphi(n)$ using the EEA.

Why are d_k and e_k inverse functions?

We need to show that:

$$x^{ab} \equiv x \mod n$$

for all $x \in \mathbb{Z}_n^*$, which is the only case that is important in practice. This also holds for all $x \in \mathbb{Z}_n$ (exercise).

Since $ab \equiv 1 \mod \varphi(n)$, we can write:

$$ab = 1 + t\varphi(n) \text{ for some integer } t.$$

Thus,

$$x^{ab} = x^{1+t\varphi(n)} \equiv x(x^{\varphi(n)})^t \mod n.$$

By Lagrange's theorem, we know that:

$$x^{\varphi(n)} \equiv 1 \mod n.$$

Therefore,

$$x(x^{\varphi(n)})^t \equiv x \cdot 1^t \equiv x \mod n.$$

Security Considerations

The best factoring algorithms can currently factor RSA moduli up to around 768 bits. This means that p, q should each be at least 384 bits long.

Multiprecision arithmetic is required to perform computations with numbers of this size, as n will not fit within a single CPU word (typically 64-bit words are used in modern CPUs).