Welcome to SOCAS

# A Summary of Cryptography & Security

Publicly available at *socas.perazzolo.ch*

## Contents

# 1. Systems of Linear Congruences.

A system of linear congruences is a set of simultaneous congruences of the form:

$$\begin{aligned} a_1 * x &\equiv b_1 (\mod m_1) \\ a_2 * x &\equiv b_2 (\mod m_2) \\ &\dots \\ a_k * x &\equiv b_k (\mod m_k) \end{aligned}$$

For a solution to exist the system must be constitent, meaning that for each pair of congruences, the following condition holds:

$$a_i \equiv a_j \big(\mod \gcd\big(m_i, m_j\big)\big)$$

if the moduli are coprime then gcd is always 1, so the system is always consistent, and a unique solution exists modulo the product of the moduli.

In the general case the solution is non-unique and is given modulo the least common multiple of the moduli $\operatorname{lcm}(m_1, m_2, ..., m_k)$.

## 1.1. Solving Non-Coprime Systems of Linear Congruences.

To solve a non-coprime system of linear congruences we can use the following algorithm:

1. Pre-reduce each $ax = b (\mod m)$ by first dividing by $\gcd(a, m)$ the coefficient, the modulus and the remainder, then multiply the remainder by the modular inverse of the reduced coefficient mod the new modulus.
2. Rewrite each moduli as a product of its prime factors.
3. Separate the system into subsystems for each prime factor.
4. Check consistency within each subsystem.
5. Keep only the congruences with the highest power of each prime factor.
6. Solve using chinese Remainder Theorem for coprime moduli.

## 1.2. Chinese Remainder Theorem for Non-Coprime Moduli.

1. Compute $N = m_1 * m_2 * ... * m_k$
2. For each $i$, compute $N_i = \frac{N}{m_i}$
3. For each $i$, compute the modular inverse of $\frac{N}{m_i} \mod m_i = y_i$
4. The solution is given by $x \equiv \Sigma(a_i * N_i * y_i) (\mod N)$

## 2. Extended Euclidean Algorithm.

*Mainly to find the modular inverse of a mod m.*

Bezout's identity states that for any integers $a$ and $b$, there exist integers $s$ and $t$ such that:

$$\gcd(a, b) = s * a + t * b$$

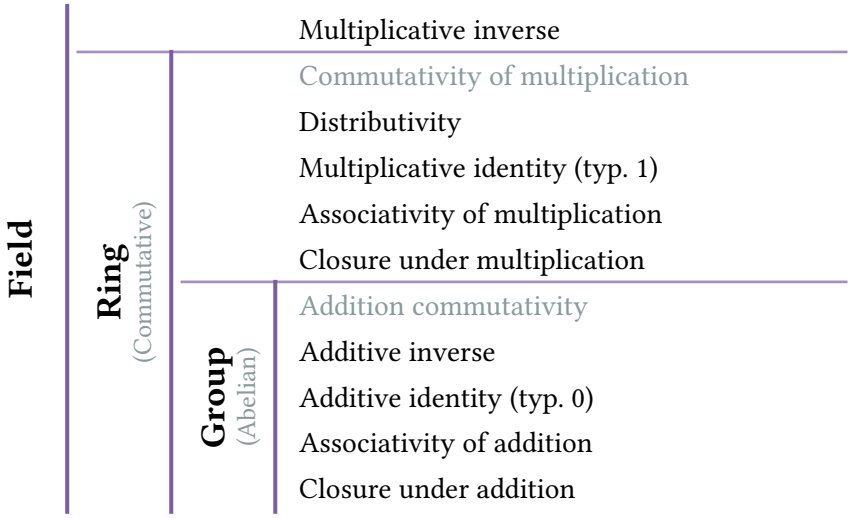
To find $s$ and $t$, we can use the Extended Euclidean Algorithm, which builds upon the standard Euclidean Algorithm for finding the gcd of two numbers.

| $q$ | $r$ | $s$ (coef of $a$) | $t$ (coef of $b$) | Equation |
|---|---|---|---|---|
| — | 240 | 1 | 0 | $240 = 1(240) + 0(46)$ |
| — | 46 | 0 | 1 | $46 = 0(240) + 1(46)$ |
| 5 | 10 | $1 - (5 \times 0) = 1$ | $0 - (5 \times 1) = -5$ | $10 = 1(240) - 5(46)$ |
| 4 | 6 | $0 - (4 \times 1) = -4$ | $1 - (4 \times -5) = 21$ | $6 = -4(240) + 21(46)$ |
| 1 | 4 | $1 - (1 \times -4) = 5$ | $-5 - (1 \times 21) = -26$ | $4 = 5(240) - 26(46)$ |
| 1 | 2 | $-4 - (1 \times 5) = -9$ | $21 - (1 \times -26) = 47$ | $2 = -9(240) + 47(46)$ |

To find new value of $s$ and $t$, remember that $r_i = r_{i-2} - q_{i-1} * r_{i-1}$, so we can substitute for $r_{i-2}$ and $r_{i-1}$ using their equations in terms of $a$ and $b$.

$$\begin{aligned} r_i &= (s_{i-2}a + t_{i-2}b) - q_{i-1} * (s_{i-1}a + t_{i-1}b) \\ &= (s_{i-2} - q_{i-1}s_{i-1})a + (t_{i-2} - q_{i-1}t_{i-1})b \end{aligned}$$

## 3. Group Theory.




| Set | Operation | Classification | Details |
|---|---|---|---|
| $\mathbb{Z}_n$ | Addition mod $n$ | Abelian Group | The "standard" clock arithmetic. Always a group for any $n > 1$. |
| $\mathbb{Z}_n$ | Add. & Mult. | Commutative Ring | Has zero divisors if $n$ is composite (e.g., $2 \cdot 3 = 0$ in $\mathbb{Z}_6$). Not a Field. |
| $\mathbb{Z}_p$ | Add. & Mult. | Field | $p$ must be prime. This ensures every non-zero element has a multiplicative inverse. |
| $\mathbb{Z}_n^*$ | Multiplication mod $n$ | Abelian Group | Contains only elements coprime to $n$. No additive properties (no zero, no additive closure). |
| $\mathbb{Z}_p^*$ | Multiplication mod $p$ | Abelian Group | Since $p$ is prime, this is just $\{1, 2, \ldots, p-1\}$. It is the "Multiplicative Group" of the field $\mathbb{Z}_p$. |

# 4. Cyclic Groups.

*A group G is cyclic if there exists an element g in G such that every element of G can be expressed as g raised to some integer power. Such an element g is called a generator of the group.*

A multiplicative group of integers modulo n is cyclic if and only if n is of the form:

$$1, 2, 4, p^k, 2p^k \text{ where } p \text{ is an odd prime and } k \geq 1 \text{ are cyclic groups.}$$

# 5. Euler's Totient Function.

*Euler's totient function $\varphi(n)$ counts the positive integers up to n that are relatively prime to n.*

## 5.1. Formula for Euler's Totient Function.

If the prime factorization of n is given by:

$$n = p_1^{e_1} * p_2^{e_2} * ... * p_k^{e_k}$$

Then Euler's totient function can be calculated using the formula:

$$\varphi(n) = n * \left(1 - \frac{1}{p_1}\right) * \left(1 - \frac{1}{p_2}\right) * ... * \left(1 - \frac{1}{p_k}\right)$$

or equivalently:

$$\varphi(n) = \left(p_1^{e_1} - p_1^{e_1 - 1}\right) * \left(p_2^{e_2} - p_2^{e_2 - 1}\right) * ... * \left(p_k^{e_k} - p_k^{e_k - 1}\right)$$

This gives the order of the multiplicative group of integers modulo n, denoted as $Z_n^*$.

# 6. Quadratic Residues and Non-Residues.

- **In any cyclic group of even order, exactly half of the elements are quadratic residues**

## 6.1. Euler criterion. (Checking quadratic residues modulo a prime).

Let p be an odd prime and a an integer not divisible by p. Then a is a quadratic residue modulo p if and only if

$$a^{\frac{p-1}{2}} \equiv 1 (\mathrm{mod}\, p)$$

**Checking quadratic residues modulo a composite modulus**:
For composite modulus $n = p_1^{e_1} * p_2^{e_2} * ...$, check every prime factor $p_1, p_2$

## 6.2. Number of quadratic residues in $Z_n^*$.

Let $n = 2^a * p_1^{e_1} * p_2^{e_2} * ... * p_k^{e_k}$ be the prime factorization of n, where $p_i$ are odd primes.

The number of quadratic residues in $Z_n^*$ is given by:

$$|\mathrm{QR}_n| = \frac{\varphi(n)}{2^m}$$

Where $\varphi(n)$ is Euler's totient function and $k$ is defined as follows:
- If $a \leq 1$, then $m = k$ (the number of odd prime factors).
- If $a = 2$, then $m = k + 1$.
- If $a \geq 3$, then $m = k + 2$.

An integer a is a quadratic residue modulo n if and only if it is a quadratic residue modulo every prime power factor $p_i^{e_i}$ in the factorization.
Any odd prime $p$ raised to any power contributes a factor of 2 to the count of quadratic residues, while the power of 2 in the factorization affects the count based on its exponent as described above.

# 7. Fermat's Little Theorem from Euler's Theorem.

Euler's theoem staes that for any integer a coprime to n:

$$a^{\varphi(n)} \equiv 1 (\mathrm{mod}\, n)$$

Fermat's little theorem is a special case of Euler's theorem for n prime = p.
In this case, $\varphi(p) = p - 1$, so we have, if a is not divisible by p:

$$a^{p-1} \equiv 1 (\mathrm{mod}\, p)$$

Multiplying both sides by a to get the general form of Fermat's little theorem:

$$a^{p} \equiv a (\mathrm{mod}\, p)$$

The general case of Fermat theorem can also be recovered if a and p are not coprime:
$p$ must divide $a$, so both sides are congruent to 0 modulo p.

$$a^{p} \equiv 0 (\mathrm{mod}\, p) \Rightarrow a^{p} \equiv a (\mathrm{mod}\, p)$$