Security & Cryptography Class Notes

Anna Visman

Academic Year 2024-2025

Contents

1 Lecture 1 **2**
- 1.1 Security Overview . . . 2
- 1.2 Defense Overview . . . 2
- 1.3 Cryptography Overview . . . 3
- 1.4 Topics Covered in Course . . . 4

2 Lecture 2 **4**
- 2.1 Number Theory . . . 4
  - 2.1.1 Modular Arithmetic . . . 4
  - 2.1.2 Modular Exponentiation . . . 5
  - 2.1.3 Groups and Rings . . . 6
  - 2.1.4 Primes and Divisibility . . . 7
  - 2.1.5 Linear Congruences . . . 7
  - 2.1.6 Fields . . . 8
  - 2.1.7 Lagrange's Theorem . . . 8
  - 2.1.8 Fermat's Little Theorem . . . 8
- 2.2 Basic Algorithms . . . 9
  - 2.2.1 Euclid's GCD Algorithm . . . 9
  - 2.2.2 The Extended Euclidean Algorithm . . . 9
  - 2.2.3 The Chinese Remainder Theorem . . . 10
  - 2.2.4 Computing Legendre and Jacobi Symbols . . . 11
- 2.3 Primality Tests . . . 12
  - 2.3.1 Fermat's Primality Test . . . 13
  - 2.3.2 Miller-Rabin Primality Test . . . 13
- 2.4 Elliptic Curves . . . 13

3 Lecture 3 **15**
- 3.1 The Syntax of Encryption . . . 15
- 3.2 Classical Ciphers . . . 15
  - 3.2.1 Caesar Cipher . . . 15
  - 3.2.2 Shift Cipher . . . 15
  - 3.2.3 Substitution Cipher . . . 15
  - 3.2.4 Vigenère Cipher . . . 15
  - 3.2.5 Permutation Cipher . . . 15

4 Lecture 4 **15**
- 4.1 Security Definitions . . . 15
- 4.2 Probability and Ciphers . . . 15
- 4.3 Perfect Secrecy . . . 15
- 4.4 One-Time Pad . . . 16
- 4.5 Entropy . . . 16
- 4.6 Spurious Keys and Unicity Distance . . . 16

1 Lecture 1

1.1 Security Overview

A computer system is said to be secure if it satisfies the following properties:

- **Confidentiality**: Unauthorized entities cannot access the system or its data
- **Integrity**: When you receive data, it is the right one
- **Availability**: The system or data is there when you need it

Remark 1. *The mere presence of these properties does not necessarily mean that the system is fully secure in practice.*

A secure system is reliable:

- Keep your personal data confidential
- Allow only authorised access or modifications to resources
- Ensure that any produced results are correct
- Give you correct and meaningful results whenever you want them

Terminology:

- **Assets**: Things we want to protect (hardware, software, data)
- **Vulnerabilities**: Weaknesses in a system that may be exploited in order to cause loss and harm
- **Threats**: A loss or harm that might befall a system (interception, interruption, modification, fabrication)
- **Attack**: An action which exploits a vulnerability to execute a threat
- **Control/Defence**: Removing/reducing a vulnerability. You control a vulnerability to prevent an attack and defend against a threat

Methods of Defence:

- Prevent it
- Deter it: make the attack harder or more expensive
- Deflect it: make yourself less attractive to attacker
- Detect it: notice that the attack is occurring
- Recover from it: mitigate the effects of the attack

Principle of Easiest Penetration: A system is only as secure as its weakest link. An attacker will go after whatever part of the system is easiest for them, not most convenient for you. In order to build secure systems, we need to learn how to think like an attacker!

1.2 Defense Overview

Software controls:

- Passwords and other forms of access control
- Operating systems separate users' actions from each other
- Virus scanner watch for malware
- Development controls enforce quality measures on the original source code
- Personal firewalls that run on your desktop

Hardware controls:

- Not usually protection of the hardware itself, but rather using separate hardware to protect the system as a whole
- Fingerprint readers
- Smart tokens
- Firewalls
- Intrusion detection systems

Physical Systems:

- Protection of the hardwell itself, as well as physical access to the console, storage media, etc.
- Locks
- Guards
- Off-site backups

Policies and Procedures:

- Non-technical means can be used to protect against some classes of attack (e.g. VPNs for accessing interal company network)
- Rules about choosing Passwords
- Training in best security practices

1.3 Cryptography Overview

Objectives of Cryptography:

- Protecting data privacy
- Authenticaion (message, data origin, entity)
- Non-repudiation: preventing the sender from later denying that they sent the message

Definition 1. *Kerkhoff's Principle: The adversary knows all details about a crypto system except the secret key.*

Definition 2. *Cipher: A method or algorithm used to transform readable data (called plaintext) into an unreadable format (called ciphertext) to protect its confidentiality.*

Encryption is the process of converting plaintext into ciphertext. Decryption is the reverse process. Encryption uses the key k, decryption uses the key k'. If k = k', the system is symmetric. If k ≠ k', the system is asymmetric. Decryption(Encryption(m)) = m.

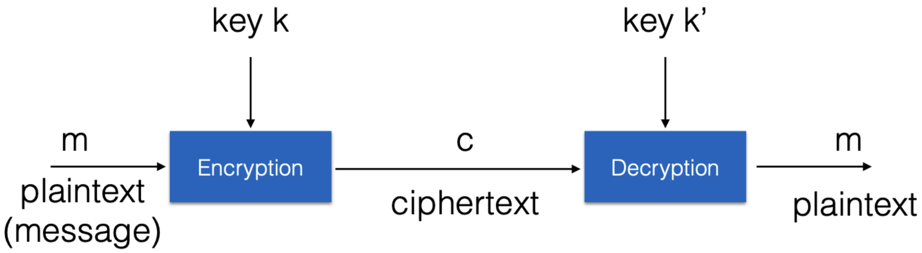


Figure 1: Encryption

Feature	**Private Key Encryption**	**Public Key Encryption**
Keys	Same key for encryption & decryption	Two keys: public and private
Speed	Faster	Slower
Key sharing	Must be kept secret	Only the private key is secret
Use cases	Encrypting large data, e.g., files	Secure key exchange, digital signatures

Table 1: Summary of Differences Between Private and Public Key Encryption

1.4 Topics Covered in Course

- Classical systems: simple ciphers, substitution, permutation, transposition, Caesar, Vigenere
- Information Theoretic Security
- Defining security: pseudorandomness, one-way functions, trapdoor functions
- Notions of security: perfect secrecy, semantic security, IND security
- Attacks on encryption schemes: objective, levels of computing power, amount of information available
- Types attacks: ciphertext-only, known plaintext, chosen plaintext, chosen ciphertext, adaptive
- Different types of adversaries: unbounded/polynomial computing power
- Security: unconditionally secure, computationally secure
- and more... see slides

2 Lecture 2

2.1 Number Theory

2.1.1 Modular Arithmetic

Definition 3. *A positive integer N is called the* modulus*. Two integers a and b are said to be congruent modulo N, written $a \equiv b \pmod{N}$, if N divides $b - a$.*

Examples:

$$18 \equiv 4 \pmod{7}, \quad -18 \equiv 3 \pmod{7}.$$

The set of integers modulo N is denoted by $\mathbb{Z}/N\mathbb{Z}$ or $\mathbb{Z}_N$:

$$\mathbb{Z}/N\mathbb{Z} = \{0, 1, \ldots, N-1\}, \quad \#(\mathbb{Z}/N\mathbb{Z}) = N.$$

Properties of Modular Arithmetic:

1. Addition is closed: $\forall a, b \in \mathbb{Z}/N\mathbb{Z} : a + b \in \mathbb{Z}/N\mathbb{Z}$.
2. Addition is associative: $\forall a, b, c \in \mathbb{Z}/N\mathbb{Z} : (a + b) + c = a + (b + c)$.
3. 0 is an additive identity: $\forall a \in \mathbb{Z}/N\mathbb{Z} : a + 0 = 0 + a = a$.
4. The additive inverse always exists: $\forall a \in \mathbb{Z}/N\mathbb{Z} : a + (N - a) = (N - a) + a = 0$.
5. Addition is commutative: $\forall a, b \in \mathbb{Z}/N\mathbb{Z} : a + b = b + a$.
6. Multiplication is closed: $\forall a, b \in \mathbb{Z}/N\mathbb{Z} : a \cdot b \in \mathbb{Z}/N\mathbb{Z}$.
7. Multiplication is associative: $\forall a, b, c \in \mathbb{Z}/N\mathbb{Z} : (a \cdot b) \cdot c = a \cdot (b \cdot c)$.
8. 1 is a multiplicative identity: $\forall a \in \mathbb{Z}/N\mathbb{Z} : a \cdot 1 = 1 \cdot a = a$.
9. Multiplication and addition satisfy the distributive law: $\forall a, b, c \in \mathbb{Z}/N\mathbb{Z} : (a + b) \cdot c = a \cdot c + b \cdot c$.
10. Multiplication is commutative: $\forall a, b \in \mathbb{Z}/N\mathbb{Z} : a \cdot b = b \cdot a$.

2.1.2 Modular Exponentiation

Modular exponentiation is a technique used to efficiently compute expressions of the form $a^b \mod m$, especially for large b. The key idea is to repeatedly square the base a, reduce modulo m at each step, and combine results as needed.

Example: Compute $3^4 \mod 11$

1. Write the problem:
$$3^4 \mod 11$$
2. Break it into smaller steps using properties of modular arithmetic:
   (a) First, compute $3^2 \mod 11$:
   $$3^2 = 9 \quad \Rightarrow \quad 9 \mod 11 = 9$$
   (b) Then, square the result to get $3^4 \mod 11$:
   $$3^4 = (3^2)^2 = 9^2 = 81 \quad \Rightarrow \quad 81 \mod 11 = 4$$
3. Final result:
$$3^4 \mod 11 = 4$$

General Algorithm: Exponentiation by Squaring

1. If b is even:
$$a^b \mod m = \left(a^{b/2} \mod m\right)^2 \mod m$$
2. If b is odd:
$$a^b \mod m = \left(a \cdot a^{b-1} \mod m\right) \mod m$$

You can also simplify the problem by reducing the base modulo:

Definition 4. *For any a, b, n, if $a \equiv b \pmod{n}$, then $a^k \equiv b^k \pmod{n}$ for any positive integer k.*

See an example of this in practice session 1 exercise 1e.

2.1.3 Groups and Rings

Definition 5. *A* group *is a set with an operation that is:*

- *Closed,*
- *Has an identity element,*
- *Associative, and*
- *Each element has an inverse.*

Definition 6. *A group is* abelian *if it is also commutative.*

Examples:

- The integers under addition $(\mathbb{Z}, +)$, where the identity is 0 and the inverse of x is $-x$.
- The nonzero rationals under multiplication $(\mathbb{Q}^*, \cdot)$, where the identity is 1 and the inverse of x is $1/x$.

Group types:

- Multiplicative group: operation is multiplication.
- Additive group: operation is addition.
- Cyclic abelian group: generated by a single element.

Definition 7. *An abelian group G is called* cyclic *if there exists an element in the group, called the* generator*, from which every other element in G can be obtained either by repeated application of the group operation to the generator, or by the use of the inverse operation.*

- *If the group operation is multiplication ($(G, \cdot)$), a generator g produces all elements by repeated multiplication or division: $h = g^x$, where h is an arbitrary element in the group.*
- *In modular arithmetic, g is a generator if $g^x \mod m$ produces all nonzero elements of the group as x varies.*

Example: The group $\mathbb{Z}_7^*$ (the multiplicative group of integers modulo 7) consists of the nonzero integers modulo 7 under multiplication. The elements of the group are:

$$\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}.$$

An element $g \in \mathbb{Z}_7^*$ is a generator if the powers $g^x \mod 7$ (for $x = 1, 2, 3, \ldots, 6$) produce **all elements** of $\mathbb{Z}_7^*$ exactly once. Let's test whether 3 is a generator:

1. Compute the powers of 3 modulo 7:
$$\begin{gathered} 3^1 \mod 7 = 3, \\ 3^2 \mod 7 = 9 \mod 7 = 2, \\ 3^3 \mod 7 = 27 \mod 7 = 6, \\ 3^4 \mod 7 = 81 \mod 7 = 4, \\ 3^5 \mod 7 = 243 \mod 7 = 5, \\ 3^6 \mod 7 = 729 \mod 7 = 1. \end{gathered}$$

2. The results are:
$$\{3, 2, 6, 4, 5, 1\}.$$

Since this list contains all elements of $\mathbb{Z}_7^*$, 3 is a generator of $\mathbb{Z}_7^*$. Other generators of $\mathbb{Z}_7^*$ include 5. You can verify this by computing $5^x \mod 7$ for $x = 1, 2, \ldots, 6$.

Definition 8. *A* ring *is a set with two operations ($+$, $\cdot$) satisfying:*

- *The set is an abelian group under addition.*
- *Multiplication is associative and closed.*
- *Distributive laws hold.*

If multiplication is commutative, the ring is called *commutative*. Examples:

- Integers, real numbers, and complex numbers form infinite rings.
- $\mathbb{Z}/N\mathbb{Z}$ forms a finite ring.

2.1.4 Primes and Divisibility

Definition 9. *An integer a divides another integer b, denoted $a \mid b$, if $b = k \cdot a$ for some integer k.*

Definition 10. *A number p is* prime *if its only divisors are* 1 *and p.*

Examples of primes: $2, 3, 5, 7, 11, \ldots$.

Definition 11. *Greatest Common Divisor: $c = \gcd(a, b)$ if and only if c is the largest number that divides both a and b.*

Theorem 1. *Every positive integer can be written as a product of primes in a unique way.*

Definition 12. *Two integers a and b are coprime, relatively prime or mutually prime if the only positive integer that is a divisor of both of them is 1.*

Definition 13. *Euler's Totient Function: $\phi(p)$ is the number of integers less than p that are relatively prime to p.*

- *If N is a prime then $\phi(N) = N - 1$.*
- *If p and q are both prime and $p \neq q$, then $\phi(pq) = (p - 1)(q - 1)$*

$$\phi(N) = N\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\ldots\left(1 - \frac{1}{p_k}\right) = n\prod_{p|n}(1 - \frac{1}{p})$$

where $p_1, p_2, \ldots, p_k$ are the prime factors of N.

Euler's Totient function counts the number of positive up to a given integer N that are relatively prime to N.

2.1.5 Linear Congruences

Finding the solution to the linear congruence equation:

$$a \cdot x \equiv b \pmod{N}$$

We want to know how many solutions exist for x modulo N given the coefficients a, b, and the modulus N.

1. Compute the greatest common divisor (gcd) of a and N, denoted as $\gcd(a, N) = g$.
2. The following cases determine the number of solutions:
   (a) **If $g = 1$:**
   - When a and N are coprime ($\gcd(a, N) = 1$), the equation has **exactly one solution** modulo N. This is because a has a multiplicative inverse modulo N.

   (b) **If $g > 1$ and $g \mid b$:**
   - If $\gcd(a, N) = g > 1$ and g divides b, then there are **exactly g solutions** modulo N.
   - These solutions can be determined by reducing the equation to a simpler congruence modulo N/g.

   (c) **If $g > 1$ and $g \nmid b$:**
   - If g does not divide b, then the equation has **no solution**. This is because b is not in the span of a modulo N.

Definition 14. *Multiplicative Inverse Modulo N: A number that, when multiplied by a given number a, gives a result of 1 modulo N. In other words, the multiplicative inverse of a modulo N is a number x such that:*

$$a \cdot x \equiv 1 \pmod{N}$$

- *The multiplicative inverse of a modulo N is denoted as a^{-1}.*
- *A multiplicative inverse of a modulo N exists only if a and N are coprime, i.e., $\gcd(a, N) = 1$.*
- *If a and N are not coprime, it's impossible to find x such that $a \cdot x \equiv 1 \pmod{N}$.*
- *When N is a prime p, then for all non-zero values of $a \in \mathbb{Z}/p\mathbb{Z}$ we always obtain a unique solution to the equation $a \cdot x \equiv 1 \pmod{p}$.*

Inverse in this case means that the two numbers multiply to 1 modulo N. Think about regular numbers: the inverse of 2 is $\frac{1}{2}$ under multiplication, because $2 * \frac{1}{2} = 1$.

2.1.6 Fields

Definition 15. *A field is a set G with two operations $(G, \cdot, +)$. It satisfies the following properties:*

- $(G, +)$ *is an abelian group with identity element 0 (G is a commutative group under addition).*
- $(G \backslash \{0\}, \cdot)$ *is an abelian group ($G \backslash \{0\}$ is a commutative group under multiplicatio).*
- *Multiplication distributes over addition, i.e., $(G, \cdot, +)$ satisfies the distributive law.*

A field is like the "ideal playground" for numbers: You can add, subtract, multiply, and divide (except by 0). Both addition and multiplication behave nicely (associative, commutative, etc.). Examples of fields include familiar systems like real numbers and rational numbers. The key difference between rings and fields is that in a ring, division is not always possible. In a field, division (except by 0) is always possible, because every nonzero element has a multiplicative inverse.

$$\mathbb{Z}/N\mathbb{Z}$$

is a field if and only if N is prime (because then every nonzero element has a multiplicative inverse). Else, it is a ring.
Think of $\mathbb{Z}/N\mathbb{Z}$ as a "clock" with N hours. Once you pass $N-1$, you wrap around back to 0. Arithmetic in $\mathbb{Z}/N\mathbb{Z}$ always "cycles" within the set $\{0, 1, \ldots, N-1\}$.

$(\mathbb{Z}/N\mathbb{Z})^*$ is the set of all elements that are invertible (the set of elements that are coprime to N).

$$(\mathbb{Z}/N\mathbb{Z})^* = \{x \in \mathbb{Z}/N\mathbb{Z} : \gcd(x, N) = 1\}$$

The size of $(\mathbb{Z}/N\mathbb{Z})^*$ is given by Euler's Totient function: $\phi(N)$. If N is a prime p, then $(\mathbb{Z}/N\mathbb{Z})^* = \{1, ..., p-1\}$.

2.1.7 Lagrange's Theorem

Lagrange's Theorem states that if $(G, \cdot)$ is a finite group with order (size) $n = \#G$, then for any element $a \in G$, the order of a (the smallest positive integer k such that $a^k = 1$) divides n. In particular, it follows that:

$$a^n = 1 \quad \text{for all } a \in G.$$

Application in Modular Arithmetic: In the context of modular arithmetic, consider the group of units $\mathbb{Z}/N\mathbb{Z}^*$ (the set of integers modulo N that are coprime to N, with multiplication as the group operation). If $x \in \mathbb{Z}/N\mathbb{Z}^*$, then the group has size $\phi(N)$, where $\phi(N)$ is Euler's totient function (the count of integers less than N that are coprime to N). Therefore:

$$x^{\phi(N)} \equiv 1 \pmod{N}.$$

2.1.8 Fermat's Little Theorem

Fermat's Little Theorem states that if p is a prime number and a is any integer, then:

$$a^p \equiv a \pmod{p}.$$

If a is not divisible by p, then this can be rewritten as:

$$a^{p-1} \equiv 1 \pmod{p}.$$

Explanation: This theorem tells us that raising a to the power of $p-1$ gives a remainder of 1 when divided by p, provided a and p are coprime. Fermat's Little Theorem is useful for simplifying modular exponentiation and serves as a foundation for more advanced results like Euler's theorem.

2.2 Basic Algorithms

2.2.1 Euclid's GCD Algorithm

The **Greatest Common Divisor** (GCD) of two integers a and b is the largest integer d such that d divides both a and b.

Key Idea: If we could factorize a and b, we could easily determine their GCD. For example, consider:

$$a = 2^4 \cdot 157 \cdot 4513^3, \quad b = 2^2 \cdot 157 \cdot 2269^3 \cdot 4513.$$

Here, the GCD is given by:

$$\gcd(a, b) = 2^2 \cdot 157 \cdot 4513 = 2{,}834{,}164.$$

However, computing prime factorizations is often impractical for large numbers. Instead, we use Euclid's Algorithm.

The Euclidean Algorithm is based on the principle:

$$\gcd(a, b) = \gcd(a \mod b, b).$$

The algorithm starts with two numbers a, b, where $a > b$. The remainder $r = a \mod b$ is computed. Then, a is repeatedly replaced with b (the smaller number), and b with r (the remainder). This process continues until the remainder is 0. The last non-zero remainder (b) is the GCD of a and b.
Steps:

1. Let $r_0 = a$ and $r_1 = b$.
2. Compute remainders $r_2, r_3, \ldots$ using:

$$r_{i+2} = r_i \mod r_{i+1}, \quad \text{where } r_{i+2} < r_{i+1}.$$

3. Stop when $r_{m+1} = 0$. The GCD is r_m.

Example: Compute $\gcd(21, 12)$:

$$\begin{aligned} \gcd(21, 12) &= \gcd(21 \mod 12, 12) = \gcd(9, 12), \\ \gcd(9, 12) &= \gcd(12 \mod 9, 9) = \gcd(3, 9), \\ \gcd(3, 9) &= \gcd(9 \mod 3, 3) = \gcd(0, 3). \end{aligned}$$

Thus, $\gcd(21, 12) = 3$.

2.2.2 The Extended Euclidean Algorithm

In addition to computing the GCD, the Extended Euclidean Algorithm finds integers x and y such that:

$$\gcd(a, b) = ax + by = r.$$

This is useful in many applications, such as finding modular inverses. For $\gcd(a, b) = d$ where $d = 1$, we can compute $ax + yN = 1$. Here $\mathbf{x}$ is the multiplicative inverse of a in modulo N. So, if $\gcd(a, N) = 1$, then $a^{-1} \mod N = x \mod N$.

Algorithm:

1. Start with $r_0 = a$, $r_1 = b$, $s_0 = 1$, $s_1 = 0$, $t_0 = 0$, $t_1 = 1$.
2. For each step, compute:

$$q_i = \left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor, \quad r_{i+1} = r_{i-1} - q_i r_i,$$
$$s_{i+1} = s_{i-1} - q_i s_i, \quad t_{i+1} = t_{i-1} - q_i t_i.$$

3. Stop when $r_{i+1} = 0$. Then, $\gcd(a, b) = r_i$, and $x = s_i$, $y = t_i$.

Example: Compute $\gcd(36, 24)$ and coefficients x, y:

$$\begin{aligned}
&\text{Step 1: } q = \left\lfloor \frac{36}{24} \right\rfloor = 1, \quad r = 36 - 1 \cdot 24 = 12, \\
&\text{Update: } x = 0 - 1 \cdot 1 = -1, \quad y = 1 - 1 \cdot 0 = 1, \\
&\text{Step 2: } q = \left\lfloor \frac{24}{12} \right\rfloor = 2, \quad r = 24 - 2 \cdot 12 = 0, \\
&\text{Update: } x = 1 - 2 \cdot (-1) = 3, \quad y = 0 - 2 \cdot 1 = -2.
\end{aligned}$$

Thus, $\gcd(36, 24) = 12$, with $x = -1$, $y = 1$.

2.2.3 The Chinese Remainder Theorem

Let $m_1, ..., m_r$ be pairwise relatively prime (i.e., $\gcd(m_i, m_j) = 1$ for all $i \neq j$). Let $x = a_i \mod m_i$ for all i. The CRT guarantees a unique solution given by:

$$x = \sum_{i=1}^{r} a_i M_i y_i \mod M$$

where

$$M_i = M/\ m_i$$

and

$$y_i = M_i^{-1} \mod m_i$$

y_i is the modular inverse of M_i modulo m_i (this can be computed using the Extended Euclidean Algorithm). The theorem is a way to solve a system of simultaneous modular congruences, finding a unique solution for a number that satisfies multiple modular equations, provided that the moduli are coprime/relatively prime.

$$\begin{gathered}
x \equiv a_1 \mod m_1 \\
x \equiv a_2 \mod m_2 \\
... \\
x \equiv a_k \mod m_k
\end{gathered}$$

In that case, there exists a **unique solution** x modulo M, where M is the product of all the moduli:

$$M = m_1 \cdot m_2 \cdot \ldots \cdot m_k.$$

Example:

$$\begin{gathered}
x \equiv 5 \mod 7 \\
x \equiv 3 \mod 11 \\
x \equiv 10 \mod 13
\end{gathered}$$

Then $M = 7 \cdot 11 \cdot 13 = 1001$. $M_1 = 1001/7 = 143$, $M_2 = 1001/11 = 91$, $M_3 = 1001/13 = 77$. $y_1 = 5, y_2 = 4, y_3 = 12$.

$$\begin{aligned}
x = \sum_{i=1}^{r} a_i M_i y_i \mod M &= 5 \cdot 143 \cdot 5 + 3 \cdot 91 \cdot 4 + 10 \cdot 77 \cdot 12 \mod 1001 \\
&= 3575 + 1092 + 9240 \mod 1001 = 13907 \mod 1001 = 894
\end{aligned}$$

add here why we need the CRT

2.2.4 Computing Legendre and Jacobi Symbols

Definition 16. *Let n be a positive integer. An integer a is called a* ***quadratic residue modulo*** *n if there exists an integer x such that:*

$$x^2 \equiv a \pmod{n}.$$

In other words, a is a quadratic residue modulo n if a is congruent to the square of some integer x modulo n.
If no such x exists, then a is called a ***quadratic non-residue modulo*** *n.*

Example: For $n = 7$, the integers modulo 7 are $\{0, 1, 2, 3, 4, 5, 6\}$. Computing the squares of each integer modulo 7:

$$\begin{aligned} 0^2 &\equiv 0 \pmod{7}, \\ 1^2 &\equiv 1 \pmod{7}, \\ 2^2 &\equiv 4 \pmod{7}, \\ 3^2 &\equiv 2 \pmod{7}, \\ 4^2 &\equiv 2 \pmod{7}, \\ 5^2 &\equiv 4 \pmod{7}, \\ 6^2 &\equiv 1 \pmod{7}. \end{aligned}$$

The quadratic residues modulo 7 are:

$$\{0, 1, 2, 4\}.$$

The quadratic non-residues modulo 7 are:

$$\{3, 5, 6\}.$$

Symmetry of squares: squaring numbers modulo n often produces repeated results due to symmetry in the group of residues. This means that different numbers can have the same square when considered modulo n. For each x, its symmetric counterpart $n - x$ produces the same square modulo n. The total number of unique quadratic residues is approximately half of n (or $\lfloor n/2 \rfloor + 1$ if 0 is included).

Definition 17. *Legendre Symbol: Let p be a prime number, and let a be an integer. The* ***Legendre symbol*** *$\left(\frac{a}{p}\right)$ is defined as follows:*

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \textit{if } a \textit{ is a quadratic residue modulo } p \textit{ and } a \not\equiv 0 \pmod{p}, \\ -1 & \textit{if } a \textit{ is a quadratic non-residue modulo } p, \\ 0 & \textit{if } p \mid a \textit{ (i.e., if } a \equiv 0 \pmod{p}\textit{)}. \end{cases}$$

- $\left(\frac{a}{p}\right) = 1$ means there exists an integer x such that $x^2 \equiv a \pmod{p}$ (i.e., a is a quadratic residue modulo p).
- $\left(\frac{a}{p}\right) = -1$ means that no such integer x exists (i.e., a is a quadratic non-residue modulo p).
- $\left(\frac{a}{p}\right) = 0$ means that a is divisible by p (i.e., $a \equiv 0 \pmod{p}$).

In simpler terms, the Legendre symbol answers the question: **"Can a be written as the square of some number, when working modulo p?"**
To detect squares modulo a prime p, we define:

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}. \tag{1}$$

Additional formulae:

$$\left(\frac{a}{p}\right) = \left(\frac{a \pmod{p}}{p}\right), \tag{2}$$

$$\left(\frac{a \cdot b}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right), \tag{3}$$

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}, \tag{4}$$

$$\left(\frac{a}{p}\right) = \begin{cases} -\left(\frac{p}{a}\right) & \text{if } p \equiv 3 \pmod 4, \\ \left(\frac{p}{a}\right) & \text{otherwise.} \end{cases} \tag{5}$$

Example: Compute the Legendre symbol $\left(\frac{15}{17}\right)$ to check if 15 is a quadratic residue modulo 17.

$$\begin{aligned}
\left(\frac{15}{17}\right) &= \left(\frac{3}{17}\right)\cdot\left(\frac{5}{17}\right) && \text{by equation (3)} \\
&= \left(\frac{17}{3}\right)\cdot\left(\frac{17}{5}\right) && \text{by equation (5)} \\
&= \left(\frac{2}{3}\right)\cdot\left(\frac{2}{5}\right) && \text{by equation (2)} \\
&= (-1)\cdot(-1)^3 && \text{by equation (4)} \\
&= 1.
\end{aligned}$$

So $\left(\frac{15}{17}\right) = 1$, and thus 15 is a quadratic residue modulo 17.

Instead of manually testing all possible values of x to see if $x^2 \equiv a \pmod p$, the Legendre symbol gives a quick, direct answer. This is especially helpful when working with large prime numbers.

- **Public-key cryptography** (like RSA) often relies on modular arithmetic and quadratic residues. The Legendre symbol helps in many cryptographic algorithms, such as those involving **Elliptic Curve Cryptography (ECC)**, **zero-knowledge proofs**, and **primality testing**.
- For example, in **the Diffie-Hellman key exchange**, one might need to check if certain numbers are quadratic residues in a modular group.

The Legendre symbol above is only defined when its denominator is a prime, but there is a generalization to composite denominators called the Jacobi symbol.

Definition 18. *For any integer a and odd integer n, the **Jacobi symbol** is defined as the product of the Legendre symbols corresponding to the* prime factors *of $n > 2$:*

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1}\left(\frac{a}{p_2}\right)^{e_2}\cdots\left(\frac{a}{p_k}\right)^{e_k},$$

where

$$n = p_1^{e_1}\cdot p_2^{e_2}\cdots p_k^{e_k}$$

is the prime factorization of n.

It is defined as follows:

$$\left(\frac{a}{n}\right) = \begin{cases} 0 & \text{if } n \mid a, \\ 1 & \text{if } a \text{ is a quadratic residue modulo } n \text{ and } a \not\equiv 0 \pmod n, \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } n. \end{cases}$$

Remark 2. *If a is square, then the Jacobi symbol will be 1. However, if the Jacobi symbol is 1, a* might *not be a square.*

2.3 Primality Tests

Prime numbers are needed almost always in every public key algorithm. How can you find prime numbers?

Theorem 2. ***The Prime Number Theorem:*** *The number of primes less than X can be given estimated with:*

$$\pi(X) \approx \frac{X}{\log(X)}$$

There are many prime numbers! The probability of a random value to be a prime is $\frac{1}{\log(p)}$. If we need a prime number with 100% certainty, we need a proof of primality.

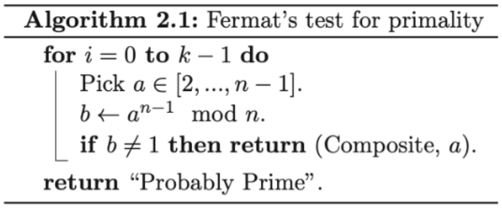
2.3.1 Fermat's Primality Test

Recall that

$$a^{\phi(N)} \equiv 1 \mod N$$

If N is a prime, this equality holds. However, if this equality holds, N is *not necessarily* prime. Probably prime: N is composite with a probability of $\frac{1}{2^k}$. k refers to the number of independent tests or iterations performed to check the primality of a number N. Each test involves choosing a random integer a and checking whether $a^{N-1} \equiv 1 \pmod{N}$.

Remark 3. *Carmichael numbers are composite numbers that pass the Fermat primality test for all possible values of a. They are rare but can be problematic in cryptographic applications. They always return* probably prime.



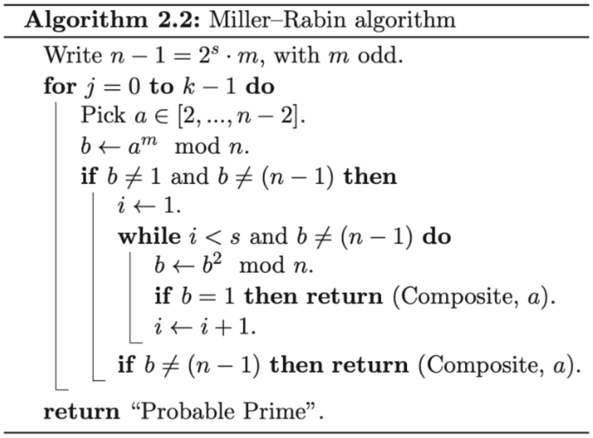
Algorithm 2.1: Fermat's test for primality

```
for $i = 0$ to $k - 1$ do
    Pick $a \in [2, ..., n - 1]$.
    $b \leftarrow a^{n-1} \mod n$.
    if $b \neq 1$ then return (Composite, $a$).
return "Probably Prime".
```

Figure 2: Algorithm for Fermat Primality Test

2.3.2 Miller-Rabin Primality Test

The Miller-Rabin test is an improvement over the Fermat test. Unlike deterministic primality tests (which can definitively prove whether a number is prime), the Miller-Rabin test provides a result with high probability. If the test declares a number to be composite, then it is definitely not prime. However, if the test declares the number to be prime, there is still a small chance that it is actually composite (this is the "probabilistic" part). The Miller-Rabin test checks whether a number n passes certain conditions that hold for all prime numbers. It does this by examining the modular arithmetic properties of numbers related to n. If n passes these tests, it is likely prime. If it fails, n is definitely composite.



Algorithm 2.2: Miller–Rabin algorithm

```
Write $n - 1 = 2^s \cdot m$, with $m$ odd.
for $j = 0$ to $k - 1$ do
    Pick $a \in [2, ..., n - 2]$.
    $b \leftarrow a^m \mod n$.
    if $b \neq 1$ and $b \neq (n - 1)$ then
        $i \leftarrow 1$.
        while $i < s$ and $b \neq (n - 1)$ do
            $b \leftarrow b^2 \mod n$.
            if $b = 1$ then return (Composite, $a$).
            $i \leftarrow i + 1$.
        if $b \neq (n - 1)$ then return (Composite, $a$).
return "Probable Prime".
```

Figure 3: Algorithm for Miller-Rabin Test

2.4 Elliptic Curves

Definition 19. *An elliptic curve is an equation of the form $F : y^2 = x^3 + ax + b \mod p$, with constants a, b.*

- *$p > 3$, otherwise $x^3 = x$*
- *If P is on F, then also $P + P, P + P + P$, ... are on F.*

Not all equations make good elliptic curves. They must satisfy a condition that ensures there are no sharp points or self-intersections. The curve must be *smooth* and *non-singular.*

$$4a^3 + 27b^2 \neq 0$$

Point Addition:

- Addition of two points P and Q on the curve gives you another point R, which is also on the curve. To add $P = (x_1, y_1)$ and $Q = (x_2, y_2)$:

  1. Draw a straight line through P and Q. This line will generally intersect the curve at exactly one more point, say R'.
  2. Reflect R' across the x-axis to get $R = (x_3, y_3)$, the result of $P + Q$.

  The formulas to compute $R = (x_3, y_3)$ are:

  $$x_3 = \lambda^2 - x_1 - x_2$$

  $$y_3 = \lambda(x_1 - x_3) - y_1$$

  where λ (the slope of the line) is:

  $$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

- If you're adding P to itself (doubling), the line you draw is the tangent to the curve at P. The formulas for $R = 2P = (x_3, y_3)$ are:

  $$x_3 = \lambda^2 - 2x_1$$

  $$y_3 = \lambda(x_1 - x_3) - y_1$$

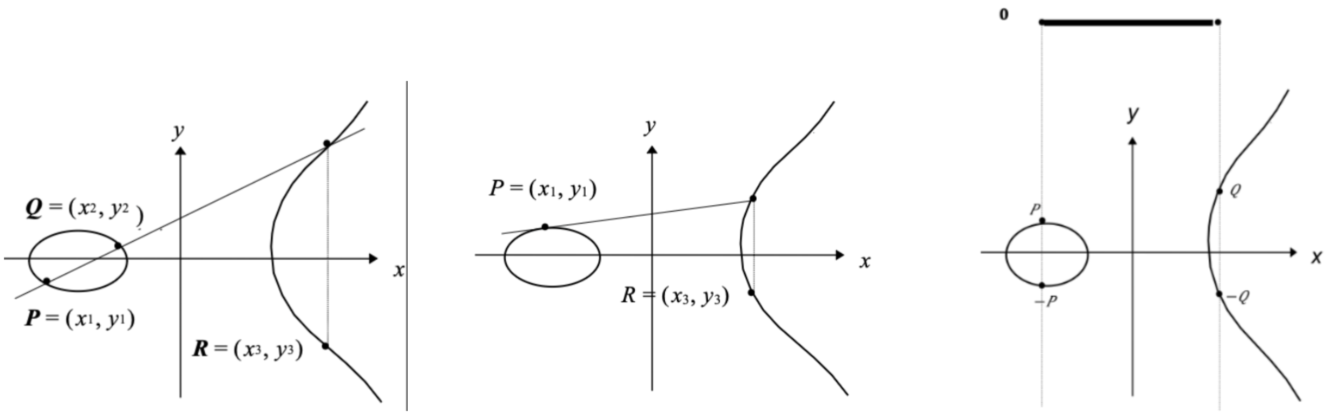
  Here, λ (the slope of the tangent) is:

  $$\lambda = \frac{3x_1^2 + a}{2y_1}$$

- If P and Q are vertical opposites (e.g., $P = (x, y)$ and $Q = (x, -y)$), the line through them is vertical, and their sum is the **point at infinity**, $\mathcal{O}$. Think of $\mathcal{O}$ as the "zero" for point addition.

- Special properties:

  - **Commutative:** $P + Q = Q + P$
  - **Associative:** $(P + Q) + R = P + (Q + R)$
  - **Identity Element:** Adding the point at infinity $\mathcal{O}$ to any point P gives P (like adding zero).



(a) Point Addition (b) Point Doubling (c) Zero Point

Using elliptic curves lets us create very secure systems with shorter keys (really large prime numbers are not required), which means faster and more lightweight encryption.

Definition 20. ***Elliptic Curve Discrete Log Problem***: *For a given integer m and a point P, it is easy to compute $Q = mP$. However, given P and Q, it is hard to compute m.*

Imagine a simple operation: start at one point on the curve, "add" it to itself repeatedly, and you get another point. If I tell you the starting point and the number of additions, it's easy to figure out the end point. But if I give you the end point and ask you to figure out the number of additions, that's really hard. This is what makes elliptic curve cryptography (ECC) secure.

3 Lecture 3

3.1 The Syntax of Encryption

3.2 Classical Ciphers

3.2.1 Caesar Cipher

3.2.2 Shift Cipher

3.2.3 Substitution Cipher

3.2.4 Vigenère Cipher

3.2.5 Permutation Cipher

4 Lecture 4

4.1 Security Definitions

Definition 21. *Computationally Secure: it takes N operations using the* ***best known algorithm*** *to break a crytographic system and N is too large to be feasible.*

Definition 22. *Provably Secure: breaking the system is reduced to solving some well-studied hard problem.*

Definition 23. *Unconditional Secure/Perfectly Secure: the system is secure against an adversary with unlimited computational power.*

Key size is important. Advances in computer hardware and algorithms are important. In the future, it will be broken due to hardware or better algorithms.

4.2 Probability and Ciphers

Definition 24. *Let P denote the set of plaintexts, K the set of keys, and C denote the set of cipher texts. $p(P = m)$ is the probability that the plaintext is m. Then,*

$$p(C = c) = \sum_{k:c\in\mathbb{C}(k)} p(K = k) \cdot p(P = d_k(c))$$

add examples in here

4.3 Perfect Secrecy

Previously, the ciphertext revelas a lot of information about the plaintext. We want a system in which ciphertext does not reveal anything about the plaintext.

Definition 25. *Perfect secrecy: a cryptosystem has perfect secrecy if*

$$p(P = m|C = c) = p(P = m)$$

for all plain texts m and ciphertexts c.

Lemma 1. *Assume the cryptosystem is perfectly secure, then*

$$\#K \geq \#C \geq \#P$$

where # denotes the number of items in the corresponding set.

4.4 One-Time Pad

Theorem 3. *Shannon's Theorem: Let $(P, C, K, e_k(), d_k())$ denote a cryptosystem with $\#K = \#C = \#P$. Then the cryptosystem provides perfect secrecy if and only if:*

- *Every key is used with equal probability $1/\#K$*
- *For each $m \in P$ and $c \in C$, there is a unique key k such that $c = e_k(m)$*

Add modified shift cipher here

4.5 Entropy

Due to the key distribution problem (the key must be as long as the message), perfect secrecy is not practical. Instead, we need a cryptosystem in which **one key can be used many times**, and **a small key can encrypt a long message**. Such a system is not perfectly secure, but it should be computationally secure. We need to measure the amount of informatiomn first: Shannon's entropy.

Example: For a specific question X: "Will you go out with me?", the answer is Yes or No. If you always say No, the amount of information, $H(X) = 0$. If you always say Yes, $H(X) = 0$. You know the result. If you say Yes and No with equal probability, $H(X) = 1$. When you get the answer, no matter what it is, you learn a lot. Here, $H()$ is the entropy, and is independent of the length of X.

Definition 26. *Shannon's Entropy: Let X be a random variable which takes a finite set of values x_i, with $1 \leq 1 \leq n$, and has a probability distribution $p(x)$. We use the convention that if $p_i = 0$ then $p_i \log_2(p_i) = 0$. The entropy of X is defined as:*

$$H(X) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$$

Properties:

- $H(X) \geq 0$
- $H(X) = 0$ *if* $p_i = 1$ *and* $p_j = 0$ *for* $i \neq j$
- *if* $p_i = 1/n$ *for all* i, *then* $H(X) = \log_2(n)$

4.6 Spurious Keys and Unicity Distance