CRYPTOGRAPHY AND ENCRYPTION

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR

Bachelor's degree in Mathematics

Jointly by,

ANUJITH RAJU	PRN: 210021031526
NEENA JAYADEVAN	PRN: 210021031535
NIDHIYA ANTONY	PRN: 210021031536

Under the supervision of

NAVIN TOMY

Department of Mathematics

Bharata Mata College Thrikkakara

2021 – 2024

Affiliated to Mahatma Gandhi University, Kottayam

ACKNOWLEDGEMENT

We like to express our sincere thanks to Mr. NAVIN TOMY, Professor of Department of Mathematics, Bharata Mata College Thrikkakara, whose guidance helped us to complete this project work. We are also thankful to Dr. LAKSHMI C, Head of the Department of Mathematics for her helping attitude and valuable advices.

We are obligated to all teachers of Department of Mathematics for their encouragement and cooperation for completing this work.

Place: Thrikkakara

Date: 23.04.2024

Anujith Raju

Neena Jayadevan

Nidhiya Antony

CERTIFICATE

This is to certify that the Dissertation entitled "**CRYPTOGRAPHY AND ENCRYPTION**" submitted by Mr. Anujith Raju in partial fulfilment of the requirements for the B.Sc. Degree in Mathematics is a bonafide record of the studies undertaken by them under my supervision of the Department of Mathematics, Bharata Mata College, Thrikkakara during 2021 – 2024 . This dissertation has not been submitted for any other degree elsewhere.

Place: Thrikkakara

Date: 23.04.2024

Mr. NAVIN TOMY

DECLARATION

We hereby declare that this project entitled "**CRYPTOGRAPHY AND ENCRYPTION**" is a bonafide record of work done by us under the supervision of Mr. NAVIN TOMY and the work has not previously formed the basis for any other qualification, fellowship, or similar title of other university or board.

Place: Thrikkakara

Date: 23.04.2024

Anujith Raju

Neena Jayadevan

Nidhiya Antony

CONTENTS

INTRODUCTION

CHAPTER 1: PRELIMINARIES

CHAPTER 2: DISCRETE LOGARITHM PROBLEM

CHAPTER 3: INTEGER FACTORISATION PROBLEM

CHAPTER 4: RSA CRYPTOGRAPHY

CHAPTER 5: APPLICATIONS OF ENCRYPTION

CONCLUSION

REFERENCE

ABSTRACT

Chapter 1 includes the preliminaries of Cryptography and Encryption

Chapter 2 includes discrete logarithm problem, primitive root of a prime number and solutions of discrete logarithm problem

Chapter 3 includes integer factorization problem and solutions of integer factorization problem

Chapter 4 includes more information about RSA cryptography

Chapter 5 includes application of encryption, encryption in daily life, study of end-to-end encryption in Whatsapp

INTRODUCTION

The Internet has had a profound impact on people's lives in the 21st century, and it's evident that ongoing advancements in information processing and telecommunications will continue to shape our daily routines. As we increasingly rely on electronic media for personal and professional activities, the secure conduct of online business transactions has become paramount. With the rise of apps like online banking and stock trading, the transmission of sensitive data over the Internet has surged. However, the technology facilitating these transactions often relies on transmitting data as binary code over unprotected channels, posing risks to data confidentiality, integrity, and validity. Users are unable to ascertain whether unauthorized individuals have intercepted or tampered with messages during transmission, and virtual communication lacks means of verifying the identity of chat partners. The National Institute of Standards and Technology (NIST) highlights the prevalence of sensitive information, such as social security numbers and credit card details, being transmitted over the Internet during transactions, underscoring the urgency of securing electronic transactions. Encryption emerges as a vital tool in ensuring the confidentiality and integrity of transactions, providing an effective means of safeguarding sensitive information.

Cryptography relies on mathematical principles to encrypt and decrypt data, enabling users to securely transmit or store private information over unsecured networks. However, the field dedicated to breaking secure communication is known as cryptanalysis. In today's digital landscape, information security is of paramount importance, with electronic documents gaining popularity as replacements for traditional paper and microfilm. Governments, businesses, and individuals increasingly seek secure information storage through electronically stored documents, which offer rapid transmission, reduced storage requirements, and simplified access via databases.

The swift rise in information's worth is due to its increased utility. Nonetheless, electronically stored data is more susceptible to security threats compared to its printed form. Electronic information is easier to pilfer remotely than printed documents, and electronic communication is more susceptible to tampering and interception. Information security involves a range of measures aimed at preventing unauthorized access to electronic data, including protecting against data loss, manipulation, destruction, and disclosure.

Mathematics plays a crucial role in cryptography, enabling the encoding and decoding of data. This allows users to transmit or store private information over insecure networks securely. The study of breaking secure communication, however, is termed cryptanalysis. In the context of cryptography, two individuals, typically represented as Alice and Bob (abbreviated as A and B), aim to communicate securely over an insecure channel, ensuring that their messages remain unreadable and unmodifiable by an eavesdropper, often referred to as Eve. The objective of cryptography is to enable two parties to communicate using codes that remain indecipherable to other parties. A participant involved in transmitting, receiving, or manipulating data is known as a party or entity, with the organization lawfully transmitting data referred to as the sender and the entity receiving information termed as the receiver. An entity attempting to compromise the security of information exchanged between the sender and receiver may assume the role of the receiver or sender. Such an adversary can also be described using various terms, including attacker, enemy, eavesdropper, opponent, or intruder.

Chapter 1

PRELIMINARIES

The notation a|b is used to denote that, given two positive integers, a and b, a divides b, meaning that b is a multiple of a. We may determine that there exists an integer k such that b = ak if a|b. The subsequent characteristics of divisibility directly after the definition.

Let the numbers a, b, and c be arbitrary.

When a|b and b|c, a|c

When a|b and a|c, for all integers i and j, then a|(ib+jc).

A = b or A = -b is true if a|b and b|a.

1.1 THE GREATEST COMMON DIVISOR (GCD)

The largest integer that divides both a and b is the greatest common divisor of positive integers a and b, or gcd (a, b). Alternatively, we may state that the number c is represented by gcd (a, b), meaning that if d|a, d|b, and finally d|c. We refer to a and b as relatively prime if gcd (a, b)=1. By using the following guidelines, we may apply the concept of GCD to a pair of arbitrary numbers. a = gcd(0,a) = gcd (a,0). The formula gcd (a,b) = gcd (|a|,|b|) accepts negative values.

Thus GCD (12, 0) = 12 and GCD (10403, 303) = 101.

1.2 MULTIPLICATION BY A FIXED NON-ZERO INTEGER, A, IN Zp IS A PERMUTATION

Select a prime number, p. The numbers (1·a) mod p, (2·a) mod p,..., ((p−1)·a) mod p, for every fixed nonzero number an in Zp, are a permutation of the set {1, 2, · · ·,p − 1}.

1.3 CHINESE REMAINDER THEOREM

The equations x mod m = a and x mod n = b have a single, unique solution if m and n are relatively prime integers and Zm and Zn are a collection of integers x in the range of 0 to mn − 1.

1.4 MODULAR MULTIPLICATIVE INVERSE

A non-negative element x of Z_n admits an inverse if and

only if gcd(x , n) = 1, ix + jn = gcd (x , n) = 1.

If we can find such integers i and j, we immediately obtain $i \equiv x^{-1} \bmod n$

The computation no of the integers I and j can be done with Extended

Euclid’s algorithm.

1.5 EXTENDED EUCLID’S ALGORITHM

Let a and b be positive integers, and denote with d their greatest common divisor , d = gcd (a , b).

Let q = a mod b and r be the integer such that a = rb + q , that is , q = a – rb. Euclid’s algorithm is based on the repeated application of the formula

D = gcd (a , b) = gcd (b , q)

Chapter 2
DISCRETE LOGARITHM PROBLEM

2.1.1. MODULAR ARITHMETIC

$a \equiv b \ (mod\ n)$
a−b = multiple of n or a = kn+b

2.1.2.DISCRETE LOGARITHM

Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups.
If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, we know every element h in G can be written as g^x for some x.
The discrete logarithm to the base g of h in the group G is defined to be x .
For example, if the group is Z_5*, and the generator is 2, then the discrete logarithm of 1 is 4 because 24 ≡ 1 mod 5.

2.1.3. PRIMITIVE ROOT OF A PRIME NUMBER

A number 'r' is a primitive root modulo n if every number coprime to n is congruent to a power of 'r' modulo n.

OR

'r' is said to be a primitive root a prime number 'p',

If r^1 mod p , r^2 mod p , r^3 mod p , … , r^{p-1} mod p are distinct

EXAMPLE 1: Is 2 a primitive root of prime number 5?

SOLUTION:

2^1 mod 5	2 mod 5	2
2^2 mod 5	4 mod 5	4
2^3 mod 5	8 mod 5	3
2^4 mod 5	16 mod 5	1

2 is a primitive root of 5.

EXAMPLE 2: Is 3 a primitive root of prime number 7?

SOLUTION:

3^1 mod 7	3 mod 7	3
3^2 mod 7	9 mod 7	2
3^3 mod 7	6 mod 7	6
3^4 mod 7	18 mod 7	4
3^5 mod 7	12 mod 7	5
3^6 mod 7	15 mod 7	1

3 is a primitive root of 7.

2.1.4. DISCRETE LOGARITHM PROBLEM

The discrete logarithm problem is defined as: given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G.

Discrete logarithm problem is not always hard. The hardness of finding discrete logarithms depends on the groups.

For Example:

Consider g^x mod p

g →generator or primitive root of p

p →prime number

To find g^x mod p in one direction is easy

But to find x if g^x mod p is given is easy if p is a smaller value and hard if p is a larger value.

So discrete logarithm problem is a one way function.

NOTE: The strength of one way function is depending on how much time it takes to break it

2.2. SOLUTIONS OF DISCRETE LOGARITHM PROBLEM

The discrete logarithm problem (DLP) is fundamental in cryptography, particularly in the context of public-key cryptography and elliptic curve cryptography. There are various algorithms for solving the discrete logarithm problem, each with different efficiency depending on the size of the group or curve involved.

2.2.1. BRUTE FORCE SEARCH

Brute force search is the most straightforward method for solving the discrete logarithm problem (DLP).

1. BASIC IDEA: The goal is to find x given g^x mod p, where g is a generator of a group modulo a prime p. Brute force search involves simply trying all possible values of x until the correct one is found.
2. SEARCH SPACE: Since 0 to p-1 are the possible values for the exponent x, the search space for brute force search is typically the integers from 0 to p-1.
3. COMPLEXITY: The time complexity of brute force search is O(p), since it involves trying p different values of x.
4. EFFICIENCY: Brute force search is the least efficient method for solving the DLP, especially for large prime numbers. Because of its inefficiency, it cannot be used in cryptographic protocols, which usually involve large prime numbers.
5. LIMITATIONS: Larger the prime modulus or the group size, more impracticable is the brute force search. For example, since there are 2^{256} possible outcomes for x with a 256-bit prime modulus, brute force search is not computationally practical with current technology.
6. USE CASES: Brute force search might be feasible in educational contexts or for small-scale experiments with very small groups or moduli.

In summary, while brute force search is conceptually simple, it is not a viable option for solving the discrete logarithm problem in practice due to its inefficiency, especially for large prime moduli.

EXAMPLE 1: Solve $\log_2 9$ mod 11.

SOLUTION:

Here p=11, g=2, X=9

$\text{Log}_g X \equiv n \pmod{p}$

$X \equiv g^n \pmod{p}$

$9 \equiv 2^n \pmod{11}$

Try 'n' = 1, 2, 3, ...

$9 \equiv 2^6 \pmod{11}$

Answer is 6

2.2.2. INDEX CALCULUS METHOD

Compared to brute force, this approach is more effective, particularly for some kinds of groups, such as prime-order subgroups of multiplicative groups of finite fields.

1. BASIC IDEA: Index calculus method is based on the idea of representing the discrete logarithm problem in terms of logarithms over a suitable auxiliary field. The discrete logarithm in the original group can be found by solving a system of linear equations in this auxiliary field.
2. PRECOMPUTATION PHASE: Before solving individual discrete logarithm instances, the index calculus method typically involves a precomputation phase where a set of "smooth" elements in the group are identified. An element g is considered smooth if its order has only small prime factors.
3. LINEAR ALGEBRA: Once a sufficient number of smooth elements are identified, the index calculus method uses linear algebra techniques to build a matrix representing relations between these smooth elements. These relations are derived from the factorization of the element's orders.
4. SOLVING LINEAR EQUATIONS: The next step involves solving a system of linear equations over a finite field using Gaussian elimination or other linear algebra techniques. The solutions to these equations provide information about the discrete logarithm of the target element.
5. COMPLEXITY: For some kinds of groups, the index calculus approach is substantially faster than brute force search due to its sub-exponential complexity. However, its efficiency depends on the size of the field and the smoothness of elements in the group.

6. APPLICABILITY: The index calculus method is particularly effective in groups where efficient algorithms for factorization exist and where the number of smooth elements can Γbe efficiently determined. It is commonly used in cryptography for solving DLP instances in prime-order subgroups of finite fields, such as the multiplicative group modulo a prime.
7. SECURITY IMPLICATIONS: The existence of index calculus method motivates the use of larger prime moduli or different types of groups with harder DLP instances to ensure cryptographic security against attacks.

EXAMPLE 1: Find

$$x \equiv \log_{22} 4 \pmod{3361}$$

SOLUTION:

$$4 \equiv 22^X \pmod{3361}$$

<u>Precomputation:</u>

1. (Choose Factor Base) Select a factor base Γ, consisting of the first 4 prime numbers,

   $\Gamma = \{2,3,5,7\}$, with $p_4 \leq 7$, the bound of the factor base.
2. (Compute 22^e mod 3361) Randomly choose a set of exponent $e \leq 3359$, compute 22^e mod 3361, and factor it as a product of prime powers:

   $22^{48} \equiv 2^5 \cdot 3^2 \pmod{3361}$,
   $22^{100} \equiv 2^6 \cdot 7 \pmod{3361}$,
   $22^{186} \equiv 2^9 \cdot 5 \pmod{3361}$,
   $22^{2986} \equiv 2^3 \cdot 3 \cdot 5^2 \pmod{3361}$.

3. (Smoothness) The above four relations are smooth with respect to B = 7. Thus,

   $48 \equiv 5 \log_{22} 2 + 2 \log_{22} 3 \pmod{3360}$,
   $100 \equiv 6 \log_{22} 2 + \log_{22} 7 \pmod{3360}$,
   $186 \equiv 9 \log_{22} 2 + \log_{22} 5 \pmod{3360}$,
   $2986 \equiv 3 \log_{22} 2 + \log_{22} 3 + 2 \log_{22} 5 \pmod{3360}$.

Compute $k \equiv \log_\beta \alpha \pmod p$

1. Compute

   $\text{Log}_{22} 2 \equiv 1100 \pmod{3360}$,

   $\text{Log}_{22} 3 \equiv 2314 \pmod{3360}$,

   $\text{Log}_{22} 5 \equiv 366 \pmod{3360}$,

   $\text{Log}_{22} 7 \equiv 220 \pmod{3360}$.

2. (Compute $4 \cdot 22^r \bmod p$) Randomly choose exponent $r = 754 \leq 3659$ and compute $4 \cdot 22^{754} \bmod 3361$.
3. (Factor $4 \cdot 22^{754} \bmod 3361$ over Γ)

   $$4 \cdot 22^{754} \equiv 2 \cdot 3^2 \cdot 5 \cdot 7 \pmod{3361}.$$

   Thus,

   $\text{Log}_{22} 4 \equiv -754 + \log_{22} 2 + 2\log_{22} 3 + \log_{22} 5 + \log_{22} 7 \equiv 2200$.

   That is,

   $$22^{2200} \equiv 4 \pmod{3361}.$$

2.2.3. POLLARD’S RHO ALGORITHM

Pollard's Rho Algorithm is a probabilistic algorithm used to solve the discrete logarithm problem (DLP) and integer factorization.

1. BASIC IDEA: Pollard's Rho Algorithm relies on the concept of a "random walk" in the group to find a collision. By repeatedly applying a function and tracking the resulting sequence of elements, eventually, two elements will collide, indicating a cycle.
2. FUNCTION CHOICE: Algorithm uses a function that maps elements of the group to other elements. A common choice for this function is a polynomial function modulo p, where p is the prime modulus of the group. The polynomial function with good random behavior and efficient collision is chosen.
3. FLOYD’S CYCLE DETECTION: Once the random walk generates a sequence of elements, the algorithm uses Floyd's cycle detection algorithm to identify a cycle within

the sequence. This involves using two pointers - one moving at twice the speed of the other - to detect a cycle.

4. COLLISION DETECTION: When a cycle is detected, it means that two elements in the sequence have collided. These collided elements provide information that can be used to compute the discrete logarithm.
5. COMLEXITY: The time complexity of Pollard's Rho Algorithm depends on the size of the group and the behavior of the chosen function. Since it has a sub-exponential complexity, it is generally more efficient than brute force search but potentially slower than more complex algorithms such as the index calculus.
6. APPLICABILITY: Pollard's Rho Algorithm is particularly effective in groups where efficient collision-finding functions can be defined. It is commonly used in cryptographic protocols, especially in contexts where the DLP instances are relatively small or where other algorithms are not applicable.
7. SECURITY IMPLICATIONS: The efficiency of Pollard's Rho Algorithm highlights the importance of choosing sufficiently large prime moduli or other types of groups to ensure cryptographic security against attacks. Also, the efficiency of the algorithm highlights the necessity of updating cryptographic systems on a regular basis as processing power grows and new algorithms are created.

EXAMPLE 1: Consider a prime modulus p = 19 and a generator g = 3. We want to find the discrete logarithm of y = 14, i.e., we want to find x such that $g^x \equiv y \pmod{p}$.

SOLUTION:

1. <u>Initialization:</u>
   - We start with two “pointers” or variables, x and y, both initialized to 1.
   - We also define a function $f(x) = g^x \bmod p$, where g = 3 and p = 19.
2. <u>Random walk:</u>
   - At each step, we update x and y using the function f(x).
   - We update x by computing f(x) once.
   - We update y by computing f(f(y)) twice.
   - We repeat these steps until we find a collision, i.e., when $x \equiv y \bmod p$
3. <u>Cycle Detection:</u>

- We use Floyd's cycle detection algorithm to detect when x and y collide.
- We maintain two pointers, one moving at twice the speed of the other.
- When the pointers meet, we know there's a cycle.

4. Collision Resolution:
   - Once a collision is detected, we backtrack to find the values of x and y that lead to the collision.
   - We calculate the difference between the values of x and y when they collide. This difference represents the discrete logarithm.

STEPS:

$x = y = 1$

$x_1 = f(x_0) = f(1) = 3$

$y_1 = f(f(y_0)) = f(f(1)) = f(3) = 9$

$x_2 = f(3) = 3^3 \bmod 19 = 8$

$y_2 = f(f(9)) = f(3) = 8$

$x_2 = y_2$ (collision detected)

$x = 3$ & $y = 8$ (values of x and y when they collided)

$x - y = 3 - 8 = -5$

Since $-5 \equiv 14 \bmod 19$, we have found the discrete logarithm of $y = 14$ with respect to base $g = 3$ and modulo $p = 19$.

2.2.4. GENERAL NUMBER FIELD SIEVE(GNFS)

The General Number Field Sieve (GNFS) is an extension of the Number Field Sieve (NFS).
It is one of the most efficient algorithms for factoring large integers and has important implications for the security of cryptographic systems.

1. BASIC IDEA: The General Number Field Sieve (GNFS) operates in accordance with the fundamental principles of the Number Field Sieve (NFS) method, yet integrates further enhancements and refinements in order to enhance overall efficiency. The process involves the utilization of sieving techniques to identify smooth relations, subsequently applying

linear algebra methodologies to address a set of linear equations, and ultimately combining the results to determine the prime factors of the specified integer.

2. SPECIAL NUMBER FIELDS: To construct the linear algebra step more efficiently, special number fields called algebraic number fields are used. These number fields are chosen based on properties that make them suitable for the factorization task.
3. SIEVING: To efficiently find smooth relations, GNFS uses a highly optimized sieving algorithm. To improve the efficiency of the sieving process, advanced techniques such as lattice sieving and large prime variation are often used.
4. LINEAR ALGEBRA: The linear algebra step in GNFS is optimized to handle large matrices efficiently. To solve the system of linear equations more quickly, advanced techniques such as block Lanczos algorithms and lattice reduction methods are employed.
5. COMPLEXITY: GNFS has a time complexity that is sub-exponential in the size of the target integer, making it significantly faster than brute force methods for factoring large integers. The size of the target integer and the efficiency of implementation are the factors on which the complexity of GNFS depends.
6. APPLICABILITY: The General Number Field Sieve (GNFS) is extensively employed for the factorization of large integers within cryptographic schemes that rely on the RSA algorithm. It is capable of factoring integers with hundreds of digits efficiently, making it a critical tool for ensuring the security of cryptographic systems.
7. SECURITY IMPLICATIONS: The efficiency of GNFS highlights the importance of using sufficiently large key sizes in cryptographic systems to resist attacks based on integer factorization. Cryptographic systems must be periodically updated to increase key sizes and maintain security against advances in factorization techniques.

Chapter 3

INTEGER FACTORISATION PROBLEM

INTEGER FACTORISATION

Integer factorization is the process of decomposing a complete number into a smaller integers which , when multiplied together , give the original number. This is a fundamental problem in number theory with applications in cryptography, particularly in public –key cryptography.

- General Number Field Sieve (GNFS);
- Pomerance's Quadratic Sieve (QS);
- Lenstra's Elliptic Curve Method (ECM);
- Pollard's p and p-1 Methods;

Integer factorization becomes significantly harder as the numbers to be factored grow larger , which is why it forms the basis of many cryptographic algorithms like RSA .The security of RSA encryption , for example , relies on the difficulty of factoring the product of two large prime numbers.

BASIC CONCEPTS

The integer factorization Problem (IFP) may be described as follow;

IFP := {Input : $n \in$ composites

{Output : f such that $f \mid n$ & $1 < f < n$

The most common equation for integer factorization is the product of prime numbers that composes the integer . For example , if you have the integer n , its factorization equation would be:

$$n = P_1\wedge a_1 \times P_2\wedge a_2 \times P_3\wedge a_3 \ldots \times P_k\wedge a_k$$

where $\mathcal{P}_i$ are prime numbers and a_i are their corresponding exponents.

Generally speaking , the most useful factoring algorithms fall into one of the following two main classes:

(A) General-purpose factoring algorithms have a runtime primarily determined by the size of the number N being factored ,rather than being heavily influenced by the size of the discovered factor p.

(B) Special purpose factoring algorithms : The running time depends mainly on the size of p (the factor found) of n.

Note that there is a quantum factoring algorithms, first proposed by Shor, which can run in polynomial-time

3.1 p and p-1 Methods

The ρ (rho) method was developed by John Pollard in 1975, while the p − 1 method was devised by Peter Montgomery in 1987. The basic concept behind the ρ (rho) and p − 1 methods lies in exploiting properties of modular arithmetic and the multiplicative structure of integers to efficiently factorize large composite numbers.

The method uses an iteration of the form:

$$x_0 = \text{random}(0, n - 1),$$

$$x_i \equiv f(x_i - 1) \pmod{n},\ i = 1, 2, 3, \dots$$

where x_0 is a random starting value, n is the number to be factored, and $f \in Z[x]$ is a polynomial with integer coefficients; usually, we just simply choose $f(x) = x_2 \pm a$ with $a = -2, 0$. Then starting with some initial value x_0, a "random" sequence $x_1, x_2, x_3, \dots$ is computed modulo n in the following way:

$x_1 = f(x_0)$

$x_2 = f(f(x_0)) = f(x_1)$

$x_3 = f(f(f(x_0))) = f(f(x_1)) = f(x_2)$

.

.

$x_i = f(x_i - 1)$.

Let d be a nontrivial divisor of n, where d is small compared with n. Since there are relatively few congruence classes modulo d (namely, d of them), there will probably exist integers x_i and x_j which lie in the same congruence class modulo d, but belong to different classes modulo N; in short, we will have

$x_i \equiv x_j \pmod{d}$,

$x_i \equiv x_j \pmod{n}$.

EXAMPLE 1.1

Let $n = 1387 = 19 \cdot 73$, $f(x) = x_2 - 1$ and $x_1 = 2$. Then the "random" sequence x_1, x_2, x_3….. is as follow:

2, 3, 8, 63, 1194, 1186, 177, 814, 996, 310, 396, 84, 120, 529, 1053, 595, 339

where the repeated values are overlined. Now we find that

$x_3 \equiv 6 \pmod{19}$

$x_3 \equiv 63 \pmod{1387}$

$x_4 \equiv 16 \pmod{19}$

$x_4 \equiv 1194 \pmod{1387}$

$x_5 \equiv 8 \pmod{19}$

$x_5 \equiv 1186 \pmod{1387}$

.

.

So we have

$\gcd(63 - 6, 1387) = \gcd(1194 - 16, 1387) = \gcd(1186 - 8, 1387) = \cdots = 19$.

Of course, as mentioned earlier, d is not known in advance, but we can keep track of the integers x_i which we do know, and simply compare x_i with all the previous x_j with $j < i$, calculating gcd $(x_i - x_j, n)$ until a nontrivial gcd occurs:

gcd ($x_1 - x_0$, n) = gcd (3 − 2, 1387) = 1,

gcd ($x_2 - x_1$, n) = gcd (8 − 3, 1387) = 1,

gcd ($x_2 - x_0$, n) = gcd (8 − 2, 1387) = 1,

gcd ($x_3 - x_2$, n) = gcd (63 − 8, 1387) = 1,

gcd ($x_3 - x_1$, n) = gcd (63 − 3, 1387) = 1,

gcd ($x_3 - x_0$, n) = gcd (63 − 2, 1387) = 1,

gcd ($x_4 - x_3$, n) = gcd (1194 − 63, 1387) = 1,

gcd ($x_4 - x_2$, n) = gcd (1194 − 8, 1387) = 1,

gcd ($x_4 - x_1$, n) = gcd (1194 − 3, 1387) = 1,

gcd ($x_4 - x_0$, n) = gcd (1194 − 2, 1387) = 1,

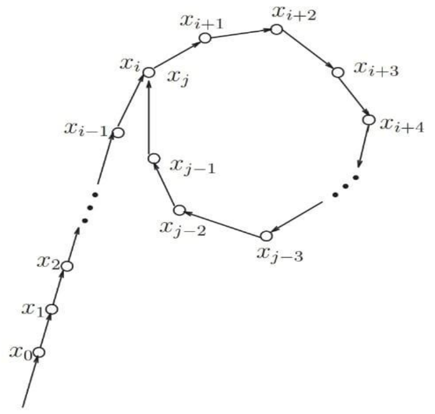
gcd ($x_5 - x_4$, n) = gcd (1186 − 1194, 1387) = 1,

gcd ($x_5 - x_3$, n) = gcd (1186 − 63, 1387) = 1,

gcd ($x_5 - x_2$, n) = gcd (1186 − 8, 1387) = 19

So after 13 comparisons and calculations, we eventually find the divisor 19.

Illustration of *ρ* Method



EXAMPLE 1.2

Let once again n = 1387 = 19 · 73, f (x) = x_2 − 1, and $x_0 = y_0 = 2$. By comparing pairs x_i and x_{2i} , for i = 1, 2,..., we get:

$f(y_0) = 22 - 1 = 3$,

$f(f(y_0)) = 32 - 1 = 8 = y_1$

$\Longrightarrow \gcd(y_1 - x_1, N) = \gcd(3 - 8, 1387) = 1$

$f(y_1) = 82 - 1 = 63$,

$f(f(y_1)) = 632 - 1 = 1194 = y_2$

$\Longrightarrow \gcd(y_2 - x_2, N) = \gcd(8 - 1194, 1387) = 1$

$f(y_2) = 11942 - 1 \bmod 1387 = 1186$,

$f(f(y_2)) = 11862 - 1 \bmod 1387 = 177 = y_3$

$\Longrightarrow \gcd(y_3 - x_3, N) = \gcd(63 - 177, 1387) = 19$.

The divisor 19 of 1387 is then found.

EXAMPLE 1.3

Use p − 1 method to find three prime factors of 271 − 1.

SOLUTION

The *p-1* method is a factorization method based on Fermat's Little Theorem. It's particularly useful when *p*−1 has only small prime factors. Here's how it works:

Given a number *n* and a base a^n-1 ≡ 1 mod n, then either *n* is prime , or *n* shares a common factor with a^n-1 – 1.

Let’s apply this method to find three prime factor of 271 -1 = 270:

1. Choose a suitable base a . common choices include small prime numbers. Let’s try *a* = 2.
2. Compute 2^270 mod 271. This can be done efficiently using repeated squaring modulo 271.
   2^270 mod 271 ≡ 1
   Since 2^270 ≡ 1 mod 271 , we know that either 271 is prime , or it shares a factor with 2^270-1.
3. Let’s find the greater common divisor (gcd) of 2^270 -1 and 271. This can be done using an algorithms like the Euclidean algorithm.

gcd(2^270-1,271)=gcs(1,271) =1

since the gcd is 1 , it indicates that 271 is a prime number.

So, using the p -1 methods ,we've found one prime factor of 271 -1 , which is 271.

3.2 lenstra's elliptic curve Method

Lenstra's elliptic curve factorization method, also known as the Lenstra elliptic curve factorization algorithm (ECM), is a powerful technique for factoring large composite integers into their prime factors. The method was proposed by Hendrik Lenstra in 1987 and builds upon the properties of elliptic curves over finite fields.

The basic idea behind Lenstra's ECM is to exploit the group structure of points on elliptic curves to find factors of a given integer N. It involves randomly choosing points on an elliptic curve defined over a finite field and performing operations on these points to search for non-trivial factors of N.

The algorithm proceeds as follows:

1. **Choose an elliptic curve :** select an elliptic curve E defined over a finite |Fp , where Ƥ is a prime number . the curve equation is typically in the form *y^2 =x^3 +ax+b* where a and b are coefficient chosen such that the curve has a certain properties necessary for the algorithm.

2. **Choose a point :** Randomly select a point P on the elliptic curve E defined over |FP

3. **Factor finding :** Use scalar multiplication on P to generate a sequence of points P,2P,3P,. . . ,KP. At each step, perform operations modulo N to avoid excessively large integers. If, at any point, the algorithm encounters a point with a small order or a point that fails to satisfy certain conditions, it can be used to find a non-trivial factor of N.

4. **Repeat or switch curves :** If the algorithm does not find a factor, repeat the process with a different curve or different starting point. The efficiency of the algorithm can be improved by using various optimization techniques, such as using curves with special properties or employing multiple curves simultaneously.

Lenstra's ECM is particularly efficient at finding small prime factors of composite numbers, making it a valuable tool for integer factorization, especially in the context of cryptographic protocols where the security relies on the difficulty of factoring large numbers. However, it is less effective for factoring semiprime numbers with large prime factors.

EXAMPLE 2.1

Let's say we want to factor the composite number *N =91*

SOLUTION

1. **Choose an elliptic curve :** we can choose an elliptic curve defined over a finite field , such as E :y^ =x^3 – 3x + 4 over the prime field |F97
2. **Choose a point :** we randomly select a point P on the elliptic curve. Let's say P =(17,10).
3. **Factor finding:** we perform scalar multiplication on P to generate a sequence of points .modulo N .For instance , 2P = (29,17), 3P =(77,5) , 4P =(26,16) and so on. At some point, we may encounter a point that helps us find a non-trivial factor of N
4. **Repeate or switch curves :** If needed, we can repeat the process with a different curve or starting point until we find the factors.

EXAMPLE 2.2

Let's say we want to factor the composite number *N =143*

SOLUTION

1. **Choose an elliptic curve :** we can choose an elliptic curve. Such as :y^2=x^3-3x+5 over the prime field |F149.
2. **Choose a point :** we randomly select a point P on the elliptic curve. Let's say P = (37,42).

3. **Factor finding :** We perform scalar multiplication on P modulo N. For example, $2P$=(141,134), $3P$=(94,88), $4P$=(26,27), and so on. At some point, we may encounter a point that helps us find a non-trivial factor of N
4. **Repeat or switch curves :** If needed, we can repeat the process with a different curve or starting point until we find the factors.

These are simplified examples, and in practice, the algorithm involves more complex calculations and optimizations. Additionally, the choice of curve and starting point can significantly affect the efficiency of the factorization process.

Chapter 4

RSA CRYPTOGRAPHY

RSA cryptology, named after its creators Rivest, Shamir, and adleman, is a widely used encryption algorithm. It relies on the mathematical difficulty of factoring large prime numbers to secure data transmission. In RSA , a public key is used for encryption, while a private key is used for decryption. This asymmetry allows secure communication over insecure channels. RSA is a cornerstone of cybersecurity, used in HTTPS, SSH, and other secure protocols.

RSA Encryption relies on the mathematical properties of large prime numbers and modular arithmetic.let’s see how it works;

1. Key Generation:

Choose two large prime numbers, p and q.

Compute their product, $n = p * q$. This is used as the modulus for both the public and private keys.

Compute Euler's totient function, $\varphi(n) = (p-1)(q-1)$.

Choose an integer e such that $1 < e < \varphi(n)$ and e is coprime with $\varphi(n)$. This is the public exponent.

Compute the modular multiplicative inverse d of e modulo $\varphi(n)$. This is the private exponent.

2. Encryption:

To encrypt a message M,the sender uses the recipients public key(e,n).

$$C \leftarrow M^e \bmod n \qquad \text{(RSA Encryption)}$$

3. Decryption

To decrypt the ciphertext C, the recipient uses their private key (d, n).

$$M \leftarrow C^d \bmod n \qquad \text{(RSA Decryption)}$$

The security of RSA relies on the difficulty of factoring the modulus n into its prime factors p and q. As long as n is sufficiently large and remains secret, it is computationally infeasible to determine d from e and n, making RSA a secure encryption scheme.

Using RSA for Digital Signatures

It is implied that the RSA cryptosystem directly allows digital signatures by the symmetry of the encryption and decryption procedures. Indeed, by using the decryption function on message M, a digital signature S is produced.

$$S \leftarrow M^d \bmod n \qquad \text{(RSA signature)}$$

The encryption function is now used to verify the digital signature S.i.e by checking ,

$$M \equiv S^e \pmod n \qquad \text{(RSA verification)}$$

Analysis and Setup for RSA Encryption

It is easy to analyse the running times for RSA encryption, decryption, signatures, and verification. In fact, the FastExponentiation () function can be used to do the constant number of modular exponentiations needed for each such operation.

THEOREM 2

Let n represent the RSA cryptosystem's modulus. The arithmetic operations required for RSA encryption, decryption, signing, and verification are O (log n).

The public and private key pairs must be created in order to configure the RSA cryptosystems. Specifically, we must calculate the associated public key (e, n) and private key (d, p, q). It involves ;

- choosing two randomly chosen primes, p and q, with a specified bit count. Primality testing of the random integers can be used to achieve this.
- Choosing an integer that is relatively prime to Φ (n). To do this, choose primes less than Φ (n) at random until we find one that does not divide Φ (n). In actuality, checking small primes from a list of known primes is sufficient
- In $Z_{\Phi(n)}$, find the multiplicative inverse d of e. The extended Euclid's algorithm can be used for this.

HASH FUNCTIONS

Hash functions are often used in information security applications and are very helpful. A mathematical operation known as a hash function transforms an input numerical value into another numerical value that has been compressed. The hash function accepts arbitrary-length inputs, but its result is always of a fixed length. Message digest, or just hash values, are values that a hash function returns.

FEATURES OF HASH VALUES

1. Fixed length output

- Data of any length can be transformed into a fixed length using a hash function. Hashing the data is a common term used to describe this procedure.
- Because hash functions are typically significantly smaller than input data, they are also referred to as compression functions.
- Since hash is smaller representation of a larger data, it is
- also referred to as a digest.
- An n-bit hash function is a hash function that produces n bits of output. Well-known hash algorithms provide values in the 160–512 bit range.

2. Efficiency of operation

- Computation of h(x) for any hash function h given input x is often a quick process.
- Compared to symmetric encryption, hash functions are much faster computationally.

PROPERTIES OF HASH FUNCTIONS

The following characteristics are needed for the hash function to be a useful cryptographic tool:

1.Pre - Image Resistance

- A hash function should be computationally difficult to reverse according to this criteria.
- Stated otherwise, if a hash function (h) yields a hash value (z), then it should be a challenging task to identify any input value (x) that hashes to z.
- This property guards against an attacker attempting to locate the input with just a hash value.

2. Second Pre - Image Resistance

- This property states that it should be difficult to find another input with the same hash given an input and its hash.
- Put differently, if a hash function h yields the hash value h(x) given an input value x, then it should be challenging to discover another input value y such that h(y) = h(x).
- This feature of the hash function guards against an attacker who obtains the input value and its hash and tries to replace the original input value with a false value that is acceptable.

3. Collision Resistance

- Because of this characteristic, it ought to be challenging to discover two distinct inputs of any length that produce the same hash. Another name for this characteristic is collision-free hash functions.

APPLICATIONS OF HASH FUNCTIONS

Hash functions are used in two main applications.

1. Password storage
   Password storage is protected using hash functions
   - The hash values of passwords are stored in files by most login processes, rather than the password being stored in clear text.
   - The password file is made up of a table with pairings that have the format

2.Data integrity check

One of the most used applications of hash functions is data integrity check. It is employed in the generation of data files and checksums.

One-way hash functions : sometimes known as ***message digests*** or ***fingerprints***, are frequently employed with public-key cryptosystems. The following is a non-formal explanation of this function. An one-way hashing algorithm , A string (message) M of any length is mapped by H to an integer d = H (M) with a set number of bits, known as the digest of M, which meets the following requirements:

1.The digest of a string M can be computed fast given the string M.

2. It is not computationally possible to obtain M given the digest d of M, but not M.
When two strings M1 and M2 cannot be found using the same digest, then a one-way hash function is considered ***strongly collision-resistant.*** Similarly, if it is computationally impossible to find another string M0 with the same digest given a string M, then the function is considered ***collision-resistant.***

Digital signature creation can be speed up by the use of one-way hash algorithms. It is possible to sign a message's digest rather than the message itself if we have a collision-resistant, one-way hash function; the signature S is therefore provided by:

$S = D (H (M))$

Chapter 5

APPLICATIONS OF ENCRYPTION

5.1. APPLICATIONS OF ENCRYPTION

1. SECURE COMMUNICATION:

   Encryption serves as a vital tool for safeguarding confidential data transmitted across networks, including emails, instant messages, and online banking transactions. Its role is to encode information in a manner that only authorized individuals can decipher, ensuring the security and integrity of the data during transmission and preventing unauthorized access or tampering.

2. DATA PROTECTION:

   Encryption aids in protecting data stored on various devices such as computers, smartphones, and servers, thereby preventing unauthorized access in the event of theft or loss.

3. AUTHENTICATION:

   Encryption plays a crucial role in digital signatures and certificates by confirming the legitimacy of users and websites, thus guaranteeing secure connections.

4. SECURE TRANSACTIONS:

   Encryption enhances the security of online transactions by encoding credit card details, passwords, and other confidential data while facilitating payment procedures.

5. FILE PROTECTION:

   Encryption can be utilized on single files or entire disk drives to prevent unauthorized entry, particularly for sensitive documents or intellectual property.

6. PRIVACY PRESERVATION:

Encryption contributes to safeguarding privacy by ensuring that only authorized individuals can access personal data, thereby reducing the risk of surveillance or identity theft.

7. REGULATORY COMPLIANCE:

   Industries like healthcare and finance are mandated to employ encryption for sensitive data to meet regulatory requirements such as HIPAA and GDPR.

8. SECURE MESSAGING:

   End-to-end encryption within messaging applications guarantees that only the sender and designated recipient can access messages, thereby enhancing privacy and confidentiality.

9. DATA INTEGRITY:

   Encryption methods can additionally serve to verify data integrity by identifying any unauthorized alterations or tampering.

5.2. END-TO-END ENCRYPTION IN WHATSAPP

End-to-end encryption on WhatsApp guarantees that only the sender and the designated recipient(s) possess the ability to view the messages. End-to-end encryption within WhatsApp offers users high levels of privacy and security, ensuring that only the sender and designated recipient(s) can access the content of their messages.

1. KEY GENERATION:

   Upon initiating usage of WhatsApp, the application creates a set of cryptographic keys specifically for you, consisting of a public key and a private key.

2. PUBLIC KEY EXCHANGE:

Upon initiating usage of WhatsApp, the application creates a set of cryptographic keys specifically for you, consisting of a public key and a private key.

3. MESSAGE ENCRYPTION:

When you transmit a message, your WhatsApp application encrypts it utilizing the public key of the recipient. Consequently, solely the recipient's device, equipped with the corresponding private key, possesses the capability to decrypt and access the message.

4. END-TO-END ENCRYPTION:

Encryption takes place on your device and persists until the recipient's device decrypts the message. This guarantees that no entity, including WhatsApp, can intercept and view your messages during transmission.

5. SECURITY VERIFICATION:

WhatsApp offers a security verification function, enabling users to confirm the legitimacy of their conversations through the comparison of security codes. This measure assists in preventing man-in-the-middle attacks and upholding the integrity of communication.

5.3. ENCRYPTION IN EVERYDAY LIFE

Encryption is essential in multiple aspects of daily life, ensuring privacy, security, and integrity in various digital interactions.

1. Messaging Apps:

Messaging platforms such as WhatsApp, Signal, and iMessage employ encryption to safeguard the confidentiality of conversations, guaranteeing that only the sender and recipient possess the ability to access the content.

2. Email:

Secure email providers utilize encryption to safeguard the privacy of email communications, inhibiting unauthorized access to the contents of emails.

3. Online Banking:

   Financial institutions and banks employ encryption to ensure the security of online banking transactions, safeguarding sensitive details like account numbers, passwords, and financial activities from interception and unauthorized access.

4. Online Shopping:

   E-commerce platforms utilize encryption to safeguard transactions, encoding credit card details and personal information during online purchases to prevent unauthorized access and fraudulent activities.

5. Social Media:

   Social media networks utilize encryption to safeguard user data and communications, thereby ensuring the privacy and security of personal information exchanged on these platforms.

6. File Storage and Sharing:

   Encryption is employed by cloud storage platforms like Google Drive and Dropbox to safeguard files stored on their servers, thereby ensuring that only individuals with authorization can access and make changes to the data.

7. Smartphones and Devices:

   Encryption safeguards data stored on smartphones, laptops, and other devices, ensuring the protection of personal information, photos, and documents in the event of loss or theft.

8. Web Browsing:

   Encryption safeguards data stored on smartphones, laptops, and other devices, ensuring the protection of personal information, photos, and documents in the event of loss or theft.

CONCLUSION

The security of the Internet is improving, especially regarding the infrastructure designed to protect financial transactions, which is advancing as the Internet becomes increasingly integral to business operations. With the rising popularity of multicast communication, it is reasonable to anticipate and prepare for the extension of the fundamental technology that safeguards multicast communication.

Cryptography is a potential solution to numerous challenges associated with using the Internet for communication. However, there are persistent concerns regarding the potential pitfalls of incorrect implementations, given that cryptography relies on the precise application of complex mathematical formulas and protocols. Additionally, the crucial role of users in providing the correct keys adds another layer of concern regarding communication security.

When employing cryptography technology securely, it's important to carefully manage how public keys are associated with user identities, how stolen keys are detected and revoked, and the duration for which a stolen key remains valuable to a criminal. Despite the intriguing nature of cryptography, its efficacy hinges on user behavior since security ultimately concerns humans. Users often choose easily memorable keys, distribute them widely, and may neglect to update them for extended periods. The complexity of cryptography renders many individuals functionally unable to comprehend it, resulting in a lack of motivation to adhere to cryptographic security protocols.

The importance of cryptography in security will continue to increase as society increasingly relies on automated information resources. Enhanced methods for access control and data security will be essential for electronic networks used in government operations, financial transactions, e-commerce, inventory management, service delivery, data storage, and distributed computing. Cryptography provides a straightforward means of securing information. However, some modern applications, such as banking systems, consider DES (Data Encryption Standard) to be insecure, supported by analytical discoveries revealing theoretical weaknesses in the cipher.

REFERENCES

1. Computational Number Theory and Modern Cryptography by Song Y.Yan
2. Introduction to Mathematical Cryptography by Peter Maga
3. MENEZES: Handbook of applied cryptography
4. STALLINGS, WILLIAM: Cryptography and network security 7th edition
5. MAO, WENBO: Modern cryptography
6. Applications of Modern Cryptology
7. COZZENS, MARGARET&MILLER, STEVEN: The Mathematics of Encryption An Elementary Introduction
8. MARTIN, KLEITH: Everyday Cryptology
9. https://youtu.be/za9azzh4v9A?si=XjQLtL6eLyWtivoH
10. https://youtu.be/yKNTfeqSEhc?si=Az0fpuX9-JuHmX1I