RECURRENT SEQUENCES AND CRYPTOGRAPHY

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTERS IN SCIENCE

IN

MATHEMATICS

SUBMITTED TO

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

BY

SARITA KUMARI

ROLL NO.411MA2077

UNDER THE SUPERVISION OF

PROF.G.K.PANDA

DEPARTMENT OF MATHEMATICS

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA, INDIA

MAY, 2013

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

This is to certify that the project report "Recurrent Sequences and Cryptography" which is being submitted by Ms. Sarita Kumari, Roll No.411MA2077, for the award of the degree of Masters in Science from National Institute of Technology, Rourkela, is a record of bonafide review work, carried out by her under my supervision. The results embodied in this project work have not been submitted to any other university or institution for the award of any degree or diploma.

To the best of my knowledge, Ms.Sarita Kumari bears a good moral character and is mentally and physically fit to get the degree.

(Dr. G. K. Panda)
Professor of Mathematics
National Institute of Technology
Rourkela-769008
Odisha, India

ACKNOWLEDGEMENTS

I wish to express my deep sense of gratitude to my supervisor Dr.G.K.Panda, Professor, Department of Mathematics, National Institute of Technology, Rourkela for his inspiring, guidance and assistance in the preparation of this project work.

I am grateful to Prof.S.K.Sarangi, Director, National Institute of Technology, Rourkela for providing excellent facilities in the Institute for carrying out this project work.

I owe a lot to the Ph.D. students Mr. Sudhasu Sekhar Rout and Ms. Saoudamini Nayak for their help during the preparation of this project work.

I am extremely grateful to my parents who are a constant source of inspiration for me.

(Sarita Kumari)

TABLE OF CONTENTS

INTRODUCTION

In number theory, study of number sequences with interesting properties has been a source of attraction since ancient times. The most beautiful and simplest of all number sequences is the Fibonacci sequence. This sequence was first invented by Leonardo of Pisa (1180-1250), who was known as Fibonacci, to describe the growth of a rabbit population . It describes the number of pairs in a rabbit population in a rabbit population after n months if it is assumed that

- the first month there is just one newly born pair,
- newly born pairs becomes productive from their second month on,
- there is no genetic problems whatsoever generated by inbreeding,
- each month every productive pair bagets a new pair, and
- the rabbit never die

Thus, if in the n^{th} month, we have a rabbits and in the $(n+1)^{st}$ month, we have b rabbits, then in the $(n+2)^{nd}$ month we will necessarily have $a+b$ rabbits. That's because we know each rabbit basically gives birth to another each month (actually each pair gives birth to another pair, but it's the same thing) and that all a rabbits give birth to another number of a rabbits, become fertile after two months, which is exactly in the $(n+2)^{nd}$ month. That's why we have the population at moment is $n+1$ (which is b) plus exactly the population at time n (which is a).

Perhaps the greatest investigator of the properties of the Fibonacci and related number sequences was Francois Edouard Anatole Lucas (1842-1891). A sequence related to the Fibonacci sequence bears his name, called the Lucas sequence, in that of Fibonacci numbers. The number of ways of picking a set (including the empty set) from the Cyclic set $\{1,2,\ldots\ldots,n\}$ without picking two consecutive numbers is given by the n^{th} Lucas Number.

A Brief History of Cryptography and Data Security:

For over 4,500 years, cryptography has existed as a means of secretly communicating information. Egyptian hieroglyphics are the first example of the use of cryptography to hide information from those not "in the know". The use of cryptographic ciphers is central to events surrounding historical figures such as Julius Caesar, Queen Elizabeth I, Mata Hari, and Alfred Dreyfus, while playing a significant role in the Allies' victory over the Axis powers during World War II, directly affecting the outcome of the Battle of Midway and other engagements[4,1]. For those interested in cryptographic history, books such as Brute Force: Cracking the Data Encryption Standard, by Matt Curtin, and The Codebreakers. The Story of Secret Writing, by David Kahn, provide interesting reading on how cryptography has affected world events.

Cryptography in its more contemporary form was fathered by Claude Shannon in 1949.Widely known for his work in electronic communications and digital computing, Shannon established the basic mathematical theory for cryptography and its counterpart, cryptanalysis. Shannon's methods relied on a unique shared secret, referred to as the key, that allowed two parties to communicate securely as long as this key was not compromised. This class of algorithms, known as private-key, secret-key, or symmetric-key, was the sole method of secure communication until 1976, when Whitfield Diffie and Martin Hellman proposed a revolutionary key distribution methodology. This methodology led to the development of a new class of algorithms, termed public-key or asymmetric-key, where a pair of mathematically related keys are used and one of these keys is made public, obviating the need for a secret shared specifically between two parties. Today, information system typically use a hybrid approach, combining the benefits of symmetric-key and public-key algorithms to form a system that is both fast and secure.

Cryptography and Data Security in the Modern World:

Cryptography currently plays a major role in many information technology applications. With more than 188 million Americans connected to the Internet, the use of cryptography to provide information security has become a top priority. Many applications- electronic mail, electronic banking, medical databases, and electronic commerce- require the exchange of private information. For example, when engaging in electronic commerce, customers provide credit card numbers when purchasing products. If the connection is not secure, an attacker can easily obtain this sensitive data. In order to implement a comprehensive security plan for a given network to guarantee the security of a connection, the following services must be provided.

- *Confidentiality:* Information cannot be observed by an unauthorized party. This is accomplished via public-key and symmetric-key encryption.
- *Data Integrity:* Transmitted data within a given communication session cannot be altered in transit due to error or an unauthorized party. This is accomplished via the use of Hash Functions and Message Authentication Codes (MACs).
- *Message Authentication:* Parties within a given communication session must provide certifiable proof validating the authenticity of a message. This is accomplished via the use of *Digital Signatures*. The only communicating party that can generate a *Digital Signatures* that will successfully verify as belonging to the originator of the message

is the originator of the message. This process validates the authenticity of the message, i.e. that the claimed originator of the message is the actual originator of the message.

- *Non-repudiation:* Neither the sender nor the receiver of a message may deny transmission. This is accomplished via *Digital Signatures* and third-party notary services.
- *Entity Authentication:* Establishing the identity of an entity, such as a person or device.
- *Access Control:* Controlling access to data and resources. Access is determined based on the privilege assigned to the data and resources as well as the privilege of the entity attempting to access the data and resources.

Chapter 1 gives the basic definitions and some theorems in number theory like Division algorithm, Fermat's theorem, Euler's theorem and Quadratic reciprocity. Also we discuss how to find large primes. Chapter 2 gives the basic definition of Cryptography, Aspects and application of Cryptography, objectives and component of cryptography, categories of cryptography, RSA algorithm and Diffie-Hellman algorithm and digital signatures. In Chapter 3 we have implemented the Lucas sequences in Public key system.

CHAPTER 1

Preliminaries

Mathematics is the queen of all sciences and number theory is the queen of mathematics.

Carl Friedrich Gauss

In this chapter we present some definitions and known results from basic number theory. This chapter serves as base and background for the study of remaining chapter.

1.1 Definition. An integer b is divisible by an integer a, not zero, if there is an integerx such that $b = ax$, and we write $a|b$. In case b is not divisible by a, we write $a \nmid b$.

1.2 Theorem. *The division algorithm. Given any integer a and b, with $a > 0$, there exist integers $q$and r such that $b = qa + r, 0 \leq r < a$. If $a \nmid b$, then r satisfies the stronger $0 < r < a$.*

Proof: Consider the arithmetic progression

$$\dots, b - 3a, b - 2a, b - a, b, b + a, b + 2a, b + 3a, \dots$$

extending indefinite in both directions. In this sequence, select the smallest non-negative member and denote it by r. Thus by definition r satisfies the inequalities of the theorem.

But also r, being in the sequence, is of the form $b - qa$, and thus q is defined in the terms of r.

To prove the uniqueness of $r$and q, suppose there is another pair q_1 and r_1 satisfying the same conditions. First we prove that $r_1 = r$. For if not, we may presume that $r < r_1$so that $0 < r_1 - r < a$, and then we see that $r_1 - r = a(q - q_1)$ and so $a|(r_1 - r)$, a contradiction to $a|b, a > 0, b > 0, \Rightarrow a \leq b$. Hence $r = r_1$, and also $q = q_1$.

Theorem 1.1 is called the division algorithm. An algorithm is a mathematical procedure or method to obtain a result. We have stated theorem 1.1 in the form there exist integer $q$and r, and this wording suggests that we have a so-called existence theorem rather than an algorithm. However , it may be observed that the proof does not give a method for obtaining the integer q and r ,because the infinite arithmetic progression

$$\ldots, b - 3a, b - 2a, b - a, b, b + a, b + 2a, b + 3a, \ldots$$

need be examined only in part to yield the smallest positive member r.

1.3 Definition. *The integer a is a common divisor of b and c in case $a|b$ and $a|c$. Since there is only a finite number of divisors of any nonzero integer, there is only a finite number of common divisor of b and c, except in the case $b = c = 0$. If at least one of b and c is not zero, the greatest among their common divisor is called the greatest common divisor of b and c and is denoted by (b, c). Similarly , we denote the greatest common divisor g of the integers $b_1, b_2, \ldots, b_n$, not zero, by $(b_1, b_2, \ldots, b_n)$.*

1.4 Theorem. *The Euclidean algorithm.[2] Given integer b and $c > 0$, we make a repeated application of the division algorithm, theorem 1.1, to obtain a series of equations*

$$\begin{aligned}
b &= cq_1 + r_1, & 0 < r_1 < c,\\
c &= r_1 q_2 + r_2, & 0 < r_2 < r_1,\\
r_1 &= r_2 q_3 + r_3, & 0 < r_3 < r_2\\
&\vdots \qquad\quad \vdots & \vdots\\
r_{j-2} &= r_{j-1} q_j + r_j, & 0 < r_j < r_{j-1},
\end{aligned}$$

$$r_{j-1} = r_j q_{j+1}.$$

the greatest common divisor (b, c) of b and c is r_j, the last nonzero remainder in the division process. Values of x_0 and y_0 in $(b, c) = bx_0 + cy_0$ can be obtained by writing each r_i as a linear combination of band c.

Proof. The chain of equation is obtained by dividing c into b, r_1into c, r_2 into $r_1, \ldots, r_j$ into r_{j-1}. The process stops when the division is exact, that is , when the remainder is zero. Thus in application of theorem 1.1 we have written the inequalities for

the remainder without an equality sign. Thus, for example, $0 < r_1 < c$, in place of $0 \leq r_1 < c$, because if r_1 were equal to zero, the chain would stop at the first equation $b = cq_1$, in which case the greatest common divisor of b and c would be c.

We now prove that r_j is the greatest common divisor g of b and c.by the following result

For any integer $x, (a, b) = (b, a) = (a, -b) = (a, b + ax)$, we observed that

$$(b, c) = (b - cq_1, c) = (r_1, c) = (r_1, c - r_1 q_2)$$

$$(b, c) = (r_1, r_2) = (r_1 - r_2 q_3, r_2) = (r_3, r_2).$$

Continuing by mathematical induction, we get

$$(b, c) = (r_{j-1}, r_j) = (r_j, 0) = r_j.$$

To see that $r_j$is a linear combination of b and c , we argue by induction that each r_i is a linear combination of $b$and c. Clearly, $r_1$is a linear combination, and likewise r_2. In general, r_i is a linear combination of $r_{i-1}$and r_{i-2}. By the induction hypothesis we may suppose that these latter two numbers are a linear combination of $b$and c, and it follows thatr_i is also a linear combination of b and c.

1.5 Definition. *An integer $p > 1$ is called a prime number, or a prime, in case there is no divisor d of p satisfying $1 < d < p$. If an integer $a > 1$ is not a prime, it is called a composite number.*

Thus, for example, 2, 3, 5, and 7 are primes, whereas 4, 6, 8 and 9 are composite.

1.6 Theorem. (*Euclid***).** *The number of primes is infinite. That is, there is no end to the sequence of primes 2 , 3 , 5 , 7 , 11 , 13 ,…*

Proof suppose that $p_1, p_2, \ldots, p_r$are the first r primesand let

$$n = 1 + p_1 p_2 \ldots p_r.$$

Note that n is not divisible by any of p_1,p_2,⋯,p_r. Hence any prime divisor p of n is a prime distinct from $p_1, p_2, \ldots, p_r$. Since n is either a prime or has a prime factor , this implies that there is a prime distinct from $p_1, p_2, \ldots, p_r$. Thus we see that for any finite r, the number of primes is not exactly r. Hence the number of primes is infinite.

CONGURENCE

1.7 Definition. *If an integer m, not zero, divides the difference $a - b$, we say that a is congruent to b modulo m and write $a \equiv b (mod\ m)$. If $a - b$ is not divisible by m, we say that a is not congruent to b modulo m, and in this case we write $a \not\equiv b (mod\ m)$.*

1.8 Definition. *If $x \equiv y (mod\ m)$ then y is called a residue of x modulo m. A set $x_1. x_2, \ldots, x_m$ is called a complete residue system modulo m if for every integer y there is one and only x_j such that $y \equiv x_j (mod\ m)$.*

1.9 Definition. *A reduced residue system modulo m is a set of integers r_i such that $(r_i, m) = 1, r_i \not\equiv r_j (mod\ m)$ if $i \neq j$, and such that every x prime to m is congruent modulo m to some member r_i o the set.*

1.10 Definition. *The number $\phi(m)$ is the number of positive integer less than or equal to m that are relatively prime to m. This ϕ is called Euler's function.*

1.11 Theorem. (Fermat's theorem). *Let p denotes a prime. If $p \nmid a$ then $a^{p-1} \equiv 1(mod\ p)$. For every integer a, $a^p \equiv a(mod\ p)$.*

1.12 Theorem. (Euler's theorem). *If $(a, m) = 1$, then*

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

QUADRATIC RECIPROCITY

1.13 Result. *The Gaussian reciprocity law. If p and q are distinct odd primes, then*

$$(\frac{p}{q})\ (\frac{q}{p}) = (-1)\ (\frac{p-1}{2})\ (\frac{q-1}{2}).$$

THE JACOBI SYMBOL

1.14 Definition. *Let Q be positive and odd, so that $Q = q_1 q_2 \dots q_s$ where the q_i are odd primes, not necessary distinct. Then the Jacobi symbol $(\frac{P}{Q})$ is defined by*
$(\frac{P}{Q}) = \prod_{j=1}^{S} (p/q_j)$
Where (p/q_j) is the Legendre symbol.

PRIMALITY TESTING AND FACTORING

1.14 Pseudo primes and Carmichael Numbers

Applications of number theory to cryptography require a supply of large primes. The method of trial division is only practical for finding primes with at most 12 to 14 digits. We need better techniques to find much larger primes. Moreover, the methods have to be fast, as it is necessary to produce a large number of primes. The methods discussed in this section and the next are very efficient, but they are probabilistic. If one of these techniques returns that a number is composite, then its certain that the number is composite. But, if it returns that a number is prime then, there is a small probability that the number is actually composite. What this means that occasionally these algorithms report a composite number as prime. We will study the probability of failure of these algorithms, and it will be clear that the chance of failure of the strong pseudo prime test is negligible. The strong pseudo prime test is sufficient for all practical purposes.

To test if a number is prime, one would like to have a simple criterion that its computationally feasible. Unfortunately, the known criteria for primality fail to meet this requirement. For example, Wilson's Theorem states that $(n-1)! \equiv -1(mod\ n)$ if and only if n is prime. Unfortunately, there is no way to compute $(n-1)!$ In a reasonable amount of time. A lot of so-called "formulas for primes" are based on Wilson's Theorem and are utterly useless.

We appeal to Fermat's Theorem for salvation. The theorem states that for primes, $p.\ a^p \equiv a\ (mod\ p)$ for all a. We investigate the extent to which the simply converse of this theorem hold, i.e, does $a^n \equiv a\ (mod\ n)$ for all a imply that n is prime? If it doesn't for which n does it fail and how often?. Note that, this criterion would be computationally feasible, as $a^n \equiv a\ (mod\ n)$ can be computed in $O(log2n)$ steps by the technique of squaring and multiplication.

1.15 Definition. *A composite number p is pseudo prime to base 2 [or 2-pseudoprime or, psp(2)], if* $2^n \equiv 2\ (mod\ n)$.

1.16 Example. Lets verify that 341 is pseudo prime to base 2.

Firstly, 341=11.31 is composite..! to compute $2^{341} \equiv 2\ (mod\ 341)$., we compute $2^{341}\ mod\ 11$, and $2^{341}\ mod\ 31$. Notice that $2^{10} \equiv 2\ (mod\ 11)$.and $2^5 \equiv 2\ (mod\ 31)$, hence,

$2^{341} \equiv 2^{10.34}2^1 \equiv 2(mod\ 11)$ implies $2^{341} \equiv 2^{5.68}2^1 \equiv 2(mod\ 31)$. It follows that

$2^{341} \equiv 2\ (mod\ 341)$, so 341 is a 2-pseudoprime. In fact, 341 is the smallest 2-pseudoprime and gives a counter example to the converse of Fermat's theorem. It was discovered by Sarrus.

1.17 Example.

Show that 561=3.11.17 is a pseudo prime to base 2.

→ Note that $2^n \not\equiv 2\ (mod\ n)$. then n must be composite. This is an instance of compositeness test, where we know a number is composite without knowing any of the factors. If $2^n \equiv 2\ (mod\ n)$. we must check the compositeness by a different method, for example, trial division, to conclude that n is a *pseudo prime.*

We are interested in the number of 2-pseudoprimes and their distribution. If only a finite number of 2-pseudoprimes exist, then we could obtain a simple criterion for primality, at least for very large numbers. Unfortunately (but not unexpectedly), there are an infinite number of 2-pseudoprimes.

If n is a 2-pseudoprime, then $2^n - 1$ is a 2-pseudoprime. Therefore, there are infinitely many 2-pseudoprimes.

1.18 Definition.[7] *A composite number n is **pseudo prime to base** a [or a-pseudo prime or, psp(2)], if* $a^n \equiv a \ (mod\ n)$.

Note that if $(a, n) = 1$, then the condition is equivalent to $a^{n-1} \equiv 1 \ (mod\ n)$.

1.19 Example.

Let's verify that 91 is a 3- pseudo prime.

A direct computation shows that $3^{91} \equiv 3 \ (mod\ 91)$ and, 91 is composite as 91=7. 13. Calculations can also be done using Fermat's theorem and, computing $3^{91}\ mod\ 7$ and $3^{91}\ mod\ 13$.

For any a, there are infinitely many a- pseudo primes.

1.20 Definition. *We say that a number n passes the pseudo prime test to base a if*

$a^n \equiv a \ (mod\ n)$.

The pseudo prime test does not require that n be composite. Prime numbers pass the pseudo prime test to any base a, and if n is composite and passes the pseudo prime test to base a, then it is an a pseudo prime. If n fails the pseudo prime test, then it must be composite. To use this test as a test for primality, we need to know the number of a-pseudo primes to determine the probability that a composite number is misidentified as prime.

There are some 50,847,534 primes less than 10^9, but only 5597 psp(2)'s. If a number n is less than 10^9 satisfies $2^n \equiv 2 \ (mod\ n)$, then there is a very high probability [= 1-(5597)/(50847534)] that it is a prime. Our chances are increased if we also test n with respect to other bases. For example, there are 685 numbers less than or, equal to 10^9 that are pseudo primes to base 2, 3, and 5. We could construct a list of these and conduct a simple primality test.

Pseudo primePrimailty Test. This test applies to numbers less than 10^9. Make a note of the 685 pseudo primes to base 2, 3, and 5.

1. Does n pass the pseudo primes to bases 2, 3, and 5? If not, n is composite.

2. Is n one of the 685 precomputedpseudo primes to bases 2, 3, &, 5? If so, n is composite: otherwise, it is a prime.

While the range of numbers to which this test is applicable is not so large as we would like, the test is much faster than trial division. We have three pseudo prime tests, which can be performed in increasing time proportional to $log\ n$. a search of 685 numbers can be done in less than 10 operations.

We will not actually use this algorithm, as there are better methods discussed later. For large n, we might hope that if n is composite, its compositeness will be revealed in some pseudo

prime test. Unfortunately, there are many composite numbers that are pseudo primes to all bases. Such numbers were first studied by R. D. Carmichael.

1.21 Definition. *A composite number n that satisfies $a^{n-1} \equiv 1 \ (mod\ n)$ for all a such that $(a, n) = 1$ is called a **Carmichael number.***

1.22 Example. 561 is a Carmichael number.

First notice that 561 is composite as 561= 3. 11. 17. To show $a^{560} \equiv 1 \ (mod\ 561)$ for all $(a, 561) = 1$. So we will prove this using Chinese Remainder Theorem and Fermat's Theorem. Since

$$a^2 \equiv 1 \ (mod\ 3) \Rightarrow a^{560} \equiv (a^2)^{280} \equiv 1 \ (mod\ 3),$$

$$a^{10} \equiv 1 \ (mod\ 11) \Rightarrow a^{560} \equiv (a^{10})^{56} \equiv 1 \ (mod\ 11)$$

and

$$a^{16} \equiv 1 \ (mod\ 17) \Rightarrow a^{560} \equiv (a^{16})^{35} \equiv 1 \ (mod\ 17)$$

we have

$a^{560} \equiv 1 \ (mod\ 561)$ for all $(a, 561) = 1$.

Carmichael numbers are characterized by the following property.

1.23 Result. A composite number n is a Carmichael number if and only if for every $p|n$, we have $p - 1|n - 1$.

Strong Pseudo primes and Probabistic Primality Testing

An improvement in the pseudo prime test for identifying probable primes comes from the following observation. Suppose p is prime; then the equation $x^2 \equiv 1 \ (mod\ p)$ has two solutions, $x \equiv -1, 1 \ (mod\ p)$. The number of solutions to $x^2 \equiv 1 \ (mod\ n)$ is more than two for composite numbers. We want to exploit this fact to make the pseudo prime test reveal the compositeness of the number. Most of the results discussed here are due to Pomerance, Selfridge, and Wagstaff.

Earlier we saw that, $a^{n-1} \equiv 1 \ (mod\ n)$ does not imply that n is prime. Suppose, n is odd: then we can write $n - 1 = 2^r s$, with $r \geq 1$ and, s is odd. If n is prime,

$\left(a^{n-1/2}\right)^2 \equiv 1 \ (mod\ n)$ implies that $a^{(n-1)/2} \equiv \pm 1 \ (mod\ n)$. If $a^{(n-1)/2} \equiv (mod\ n)$ and, $4|\ n - 1$, then, $a^{(n-1)/4} \equiv \pm 1 \ (mod\ n)$. and, so until $(n - 1)/2^r$ becomes odd. In each step while solving $x^2 \equiv 1 \ (mod\ n)$, we should get 1 or, -1 when n is prime. For composite n, we may obtain a number other than ±1.

Since computing a square root is harder than computing a square, we start with a^3 and, compute $a^{2s}, a^{1s}, \ldots..$, instead of computing the sequence $a^{n-1}, a^{(n-1)/2}, a^{(n-1)/4}, \ldots..$ More precisely, computation of a^{n-1} is accompanied by the following sequence:

$$x_0 = a^s \bmod n,\ x_1 = a^{2s} \bmod n,\ x_2 = a^{4s} \bmod n.\ ..,$$

$$x_i = a^{2i}s \bmod n, \ldots . x^r = a^{2r}s \bmod n.$$

Note that $x_r \equiv 1\ (mod\ n)$. Each term of the sequence $x_1, x_2, \ldots, x_r$ is computed by squaring the previous term in the sequence and taking the remainder modulo n.

For n primes, $x_r \equiv 1(mod\ n)$ means that $(x_{r-1})^2 \equiv x_r \equiv 1\ (mod\ n)$ that is,

$x_{r-1} \equiv \pm 1\ (mod\ n)$. If $x_{r-1} \equiv (mod\ n)$, then $x_{r-2} \equiv \pm 1\ (mod\ n)$, Continuing in this manner, we have two possibilities: either $x_0 \equiv (mod\ n)$ or, there is an index i such that $x_i \equiv -1\ (mod\ n)$. This means that for p prime, $\{x_0, \ldots . x_r\}$ can be of the form

$$\{1,1,\ldots,1\}\ or, \{*,\ldots .*,-1,1,\ldots .1\}.$$

where $*$ represents some number different from -1 or, 1, not important for our purposes.

If n is composite, the analysis fails, as $x^2 \equiv 1\ (mod\ n)$ has more solutions than just ±1. The sequence $\{x_0, \ldots . x_r\}$ can be of the form

$$\{*,\ldots .*,1,\ldots .1\}\ or, \{*,\ldots .,*\}\ or, \{*,\ldots .*,-1,1,\ldots .1\}.$$

If we get a sequence of the type $\{*,\ldots .*,1,\ldots .1\}\ or, \{*,\ldots .,*\}$, then n must be composite.

1.24 Example. We compute the sequence$x_0, x_1, x_2, \ldots, x_r$ with $a = 2$ for $n = 341$. (Recall that 341 is the smallest 2-pseudoprime.)

First, 340=4.85=$2^2$85, so $r = 2$ and, $s = 85$. Next,

$$x_0 \equiv 2^{85} \equiv 32 \quad (mod\ 341) \quad (1)$$

$$x_1 \equiv {x_0}^2 \equiv 2^{170} \equiv 1 \quad (mod\ 341) \quad (2)$$

$$x_2 \equiv {x_1}^2 \equiv 1^2 \equiv 1 \quad (mod\ 341) \quad (3)$$

The compositeness of 341 is revealed in (2) where ${x_0}^2 \equiv 1\ (mod\ 341)$. but,

$$x_0 \equiv 32 \not\equiv 1\ (mod\ 341).$$

2. Consider 561, a Carmichael number, for which all pseudo prime tests fail to reveal its compositeness. First, factor 560= 35. 2^4, so $r = 4$ and, $s = 35$. We compute the desired sequence with $a = 2$.

$$x_0 \equiv 2^{35} \equiv 263 \quad (mod\ 561)$$

$$x_1 \equiv {x_0}^2 \equiv 2^{70} \equiv 166 \quad (mod\ 561)$$

$$x_2 \equiv {x_1}^2 \equiv 2^{140} \equiv 67 \quad (mod\ 561)$$

$$x_3 \equiv 2^{280} \equiv 1 \quad (mod\ 561)$$

1.25 Definition. *Suppose n is an integer and, $n-1 = 2^r s$. Then, n is said to pass the strong pseudo prime test to base a if*

1. $a^s \equiv 1 \ (mod\ n)$. *or,*

2. $a^{s\,2i} \equiv -1 \ (mod\ n)$ *for some* $0 \leq i < r$.

1.26 Definition. *An odd composite number that passes the strong pseudo prime test to base a is called a strong pseudo prime to base a [or, $spsp\ (a)$].*

1.27 Proposition. *If n is an odd pseudo prime to base 2 then, $2^n - 1$ is a strong pseudo primes to base 2.*

Simple Primality Test. Given $n \leq 25.\ 10^9$, this algorithm determines if n is prime.

1. If n fails the strong pseudo prime test to base 2, then n is composite.

2. If n fails the strong pseudo prime test to base 3, then n is composite.

3. If n fails the strong pseudo prime test to base 5, then n is composite.

4. If n is among the 13 numbers in the following table then, n is composite otherwise n is prime.

Table: Strong pseudo primes to bases 2, 3 and 5 and, the results of tests to bases 7,11 and, 13.

	Base 7	Base 11	Base 13
25,326,001	No	No	No
161,304,001	No	Spsp	No
960,946,321	No	No	No
1,157,839,381	No	No	No
3,215,031,751	Spsp	Psp	Psp
3,697,278,427	No	No	No
5,764,643,587	No	No	Spsp
6,770,862,367	No	No	No
14,386,156,093	Psp	Psp	Psp
15,579,919,981	Psp	Spsp	No
18,459,366,157	No	No	No
19,887,974,881	Psp	No	No
21,276,028,621	No	Psp	Spsp

1.28 Example. Let n=15790321; then $n-1$ has the factorization

$$2^4 \quad \overbrace{986895}^{s}$$

We compute the terms in the strong pseudo prime test with a=2.

$$2^s \equiv 128 \quad (mod\ n)$$

$$2^{2s} \equiv 16384 \ (mod\ n)$$

$$2^{4s} \equiv -1 \quad (mod\ n)$$

This shows that n passes the strong pseudo prime test to base 2. Similarly, we verify that n passes the strong pseudo prime test to bases 3 and, 5. Since, n does not appear in the table, n is prime. This test requires approximately $O(log_2\ s)$ multiplications as opposed to $\sqrt{n}$ divisions for trial division.

1.29 Theorem. *A composite number n is a strong pseudo prime to at most $n/2$ bases.*

Pollard's $(p-1)$ **method**

The idea behind the $(p-1)$ method is the following. Suppose n is the number to be factored, and say $p|n$, p prime. Now, $a^{p-1} \equiv 1(mod\ p)$ for any $(a,p) = 1$. Suppose, $p-1$ divides a number M; then $a^M \equiv 1\ (mod\ p)$, that is $p|\ a^M - 1$. Since, $p|n$ and, $p|\ a^M - 1$, p will divide $(a^M - 1, n)$. Instead of computing $a^M - 1$, we can compute $a^M - 1\ mod\ n$ and, $(a^M - 1\ mod\ n, n)$. If the GCD is not equal to n, then we would have a non-trivial factor of n. This factor need not be p.

1.30 Example. Consider n= 1073 = 29. 37. If $p = 29$, $p - 1 = 28$. Let $a = 2$, then, $2^{28} \equiv 900\ (mod\ n)$ and, $(900 - 1, n) = 29$. Similarly, $2^{36} \equiv 777\ (mod\ n)\ and, (777 - 1, n) = 37$, the second factor.

Algorithm (**Pollard** $p-1$**-method).** Given n is composite, this factorization algorithm computes $a^{K!} mod\ n$ successively for $k \leq B$, a pre-specified bound. The GCD

$(a^{K!} - 1\ mod\ n, n)$ is computed every 25 steps by accumulating products.

1. [Initialize] Let $a = 2, m = 1, Q = 1, k$

2. [Accumulate products] Let $a = a^k\ mod\ n.\ Q = Q(a-1) mod\ n,\ k = k+1,$

$m = m+1.\ If\ m = 25$, go to step 3; otherwise, repeat step 2.

3. [Compute GCD] Let $d = (Q, n)$; if $d \neq 1\ and, d \neq n$, return d as a factor of n; otherwise, go to step 4.

4. [Necessity of back-tracking] If $d = n$, then report that its necessary to backtrack and compute the GCD often. If $d = 1$ and $k \leq B, Q = 1, m = 1$, and go to step 2; otherwise if

m ≥ B, terminate the algorithm, as n does not have a factor p with $p - 1$ consisting of small primes.

1.31 Remarks.

(1) before factoring a number, one should apply the probabilistic compositeness of the earlier section.

(2) The algorithm should be tried only after removal of small factors of n by trial division.

(3) This is not a general purpose algorithm and, will frequently fail to find any factor. But, when it works, its impressive and can find some very large factors.

(4) It seems reasonable to take $B \leq 10^6$ in the algorithm. The algorithm can be made more efficient by computing $a^{LCM[1\ldots,K]}$ instead of $a^{K!}$.

(5) It may happen in some rare cases that all the factors are discovered at once, i.e, the GCD jumps from 1 to n in one step. Then, a different value of a might reveal the factor. Examples of such numbers include 2047 and, 536870911.

CHAPTER 2

CRYPTOGRAPHY

Why are numbers beautiful? It's like asking why is Beethoven's Ninth Symphony beautiful? If you don't see why, someone can't tell you. I know numbers are beautiful. If

they aren't beautiful, nothing is.

Paul Erdős

Cryptography means writing secret code.

2.1 Definition.

Cryptography is science of converting a stream of text into coded form in such a way that only the originator and/or receiver of the coded text can decode the text. In other words, Cryptography is the science of Information security. That means, it is used to protect confidentiality of the information.

Objectives of Cryptography

Confidentiality - The information cannot be understood by anyone for whom it was unintended.

Data Integrity - The information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected.

Authentication - The sender and receiver can confirm each other's identity and the origin/destination of the information.

Aspects and Applications of Cryptography

Modern Cryptography heavily depends on Mathematics and the usage of digital systems. People need privacy and security while communicating. Cryptography provides methods and techniques for a secure communication. Cryptography is widely used for military applications to keep sensitive information secret from the enemies (adversaries).

Nowadays with the technologic progress and our dependency on electronic system, we need more sophisticated techniques for a secure communication.

Components of Cryptography

The following are the terminology commonly used in Cryptography:

Plaintext - Human Language or Normal Text.

Encryption - It is the process of converting normal text or data information into gibberish text.

Cipher Text - The Encrypted text is called Cipher text.

Decryption - It is the process of converting gibberish text into normal text or data and hence obtaining plaintext back.

Overview - (Cryptography)[7]

In the basic communication scenario, there are two parties, Alice and Bob, who want to communicate with each other. A third party, Eve, is a potential eavesdropper. Alice wants to send a message to Bob, called "Plaintext". She encrypts it using a method pre-arranged with Bob. The encrypted message is known as "Cipher text". Usually, the encryption method is assumed to be known to Eve. Bob receives the Cipher text and changes it to the plaintext by using a decryption key. The message is kept secret to Eve because of the key.

History of Cryptography

Cryptography has roots that began around 2000 B.C. in Egypt when hieroglyphics were used to decorate tombs to tell the story of the life of the deceased. The practice was not as much to hide the messages themselves, but to make them seem more noble, ceremonial, and majestic. A Hebrew cryptographic method required the alphabet to be flipped so that each letter in the original alphabet is mapped to a different letter in the flipped alphabet. The encryption method was called atbash. Atbash encryption scheme is illustrated as follows:

$$A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ M\ N\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z$$

$$Z\ Y\ X\ W\ V\ U\ T\ S\ R\ Q\ P\ O\ N\ M\ L\ K\ J\ I\ H\ G\ F\ E\ D\ C\ B\ A$$

Example: The word '$security$' is encrypted into '$hvxfirgb$'.

This method is called substitution cipher, because a character is replaced with another character. It is referred to as a monoalphabetic substitution as it uses only one alphabet.

Permutation Cipher.

Another way of encryption was by rearranging letters instead of substituting them.

For Example:

Plaintext - "$HELLO\ WORLD$"

$$H\ W$$

$$E\ O$$

$$L\ R$$

$$L\ L$$

$$O\ D$$

Cipher text - "$HWEOLRLLOD$"

Around 400 B.C., the Spartans used a system of encrypting information by writing a message on a sheet of papyrus, which was wrapped around a staff. The message was only readable if it was around the correct staff, which allowed the letters to properly match up. This is referred to as the scytale cipher. When the papyrus was removed from the staff, the writing appeared as just a bunch of random characters.

Encryption using alphabetical transpositions

Earliest documented military use of Cryptography by J. Caesar (60 BC) Julius Caesar replaced each letter by another one in the same order (i.e., by shifting) Each letter was replaced by one, n positions away modulo alphabet size.

n = shift value = key

Similar Scheme used in India Early Indians also used substitution based on phonetics similar to Latin.

Ceaser Cipher

The encryption done by shifting of alphabets.

$$A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ M$$

$$N\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z$$

Shifting alphabet by n = 3 gives

$$A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ M$$

$$D\ E\ F\ G\ H\ I\ J\ K\ L\ M\ N\ O\ P$$

$$N\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z$$

$$Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z\ A\ B\ C$$

$$CRYPTOGRAPHY$$

becomes

$$FUBSWRJUDSKB$$

Today this technique seems too simplistic to be effective, but in that day not many people could read in the first place. More Sophisticated Examples: Use any permutation (that does not preserve any order) this is not much secure as only 26 possibilities are there. Cryptography was also used during World War 2. As computers came to be, the possibilities for encryption methods and devices advanced, and cryptography efforts expanded exponentially.

Categories of Cryptography

The Encryption/Decryption method falls into two categories:

- **Symmetric key (Private key)**
- **Asymmetric key (Public key)**

In Symmetric Key algorithms, the encryption and decryption key areknown to both Bob and Alice. Usually, the encryption key is shared and the decryption key can be easily calculated from it. In many cases, both encryption and decryption key are same. All of the (pre-1970) classical cryptosystems are symmetric, as are the more recent DES (Data Encryption Standard) and AES (Advanced Encryption Standard).

Disadvantages of Symmetric Key Cryptography

- **Need for secure channel for secret key exchange**: Sharing the secret key in the beginning is a problem in symmetric key encryption. It has to be exchanged in a way that ensures it remains secret.
- **Too many keys**: A new shared key has to be generated for communication with every different party. This creates a problem with managing and ensuring the security of all these keys.
- **Origin and authenticity of message cannot be guaranteed** since both sender and receiver use the same key, messages cannot be verified to have come from a particular user. This may be a problem if there is a dispute.

Symmetric Key vs. Public Key

Consider the following situation. Suppose Alice wants to send a message to Bob in a situation such that: They did not have any prior contact. They haven't agreed on a key. Alice doesn't want to send key through a courier. Otherwise all the information that Alice sends to Bob will potentially be obtained by evil observer Eve. This problem has a solution, a scheme called 'PKC' (Public Key Cryptosystem)

Public Key Cryptography

Public Key algorithms were introduced in 1970s which revolutionized cryptography. The encryption key is public, but it is computationally infeasible to compute the decryption key without the information which is known to Bob only. It was first publicly suggested by Diffie and Hellman (at Stanford University in 1976), without practical implementation. The most popular implementation of this scheme is 'RSA'. Other versions are due to ElGamal, etc.

Symmetric vs. Asymmetric key

In Symmetric key cryptography there is only one key which is shared between both parties who are communicating. In Public key cryptography there are two different keys among which one is public key. So any one can send message. There is no need for using different key while communicating with different parties.

One-Way Functions

One-way functions are widely used in cryptography, especially public-key cryptography. A one-way function is a function that is easy to compute and difficult to reverse. How might we express this notion of a one way function informally in complexity theoretic terms? Let's see some examples of one-way functions and how they are used in the cryptosystems.

One-Way Functions - Modular exponentiation

The process of exponentiation just means raising numbers to a power. Raising a to the power b, normally denoted a^bjust means multiplying a by itself b times. In other words:

$$a^b = a * a * a * \ldots * a \quad (b\ times)$$

Modular exponentiation means computing a^bmodulo some other number n. We write this as:

$$a^b mod\ n$$

Modular exponentiation is easy, but its inversion is difficult.

One-Way Functions - Discrete Log Problem

However, given $a, n,$ and $a^b mod\ n$ (when n is prime), calculating b is regarded by mathematicians as a hard problem. This difficult problem is often referred to as the discrete logarithm problem. In other words, given a number a and a prime number n, the function

$f(b) = a^b\ mod\ n$ is believed to be a one-way function.

One-way Functions – Examples

Modular Square Roots is also a one-way funcion. Finding square root of a number modulo some other number is difficult. For example: What is the square root of 56 module 101?

Multiplication of two prime numbers is believed to be a one-way function. A popular example of public key cryptosystem based on the above one-way function is RSA.

2.1 Definition.

Suppose $a \geq 1$ *and* $m \geq 2$ *are integers. If* $gcd(a, m) = 1$ *then we say*

that a and m are relatively prime.

2.1 Example. 3 and 8 are relatively prime as $\gcd(3, 8) = 1$.

RSA (Rivest Shamir Adleman)

The most successful implementation of PKC was proposed by Rivest, Shamir and Adleman in 1977, popularly known as RSA. It is based on the idea that factorization of large integers into their prime factors is difficult.

Before RSA

In 1997, documents released by CESG, a British Cryptographic agency showed that in 1970, James Ellis discovered 'PKC'. In 1973, Clifford Cocks had written an internal document describing a version of RSA algorithm.

RSA ALGORITHM

Key Generation	
Select p and q	p, q both large primes
Calculate n and $\varphi(n)$	$n = pq,$ $\varphi(n) = (p-1)(q-1)$
Select integer e	$\gcd(\varphi(n), e) = 1;$ $1 < e < \varphi(n)$
Calculate d	$d = e^{-1} \bmod \varphi(n)$
Public key	P UK = $\{n, e\}$
Private key	P RK = $\{p, q, d\}$

Encryption	
Plaintext:	M

Ciphertext:	$C = M^e \ (mod\ n)$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \ (mod\ n)$

Primitive Root[2]

2.2 Definition.

A primitive root modulo n is any number g with the property that any number co-prime to n is congruent to a power of g modulo n.
In other words, g is a generator of the multiplicative group of integers modulo n. That is, for every integer a co-prime to n, there is an integer k such that $g^k = a(mod\ n)$.
Such k is called the index or discrete logarithm of a to the base g modulo n.

2.2 Example. 3 is a primitive root modulo 7, because

$$3^1 = 3\ mod\ 7$$

$$3^2 = 2\ mod\ 7$$

$$3^3 = 6\ mod\ 7$$

$$3^4 = 4\ mod\ 7$$

$$3^5 = 5\ mod\ 7$$

$$3^6 = 1\ mod\ 7$$

Diffie-Hellman Key Exchange

Public Key Cryptography[7]

- Whit Diffie and Marti E. Hellman first publicly introduced Public Key Cryptography at Stanford University in the year 1976.
- Although they did not had a practical implementation of the same but it opened new directions in Public Key Cryptography.

Diffe-Hellman Key Exchange

Let p be a prime and a be a primitive root modulo p.

- User A key generation:
  Select private $X_A: X_A < P$
  Calculate public $Y_A: Y_A = a^{X_A} mod\ p$.

- User B key generation:
  Select private $X_B: X_B < P$
  Calculate public $Y_B: Y_B = a^{X_B} mod\ p$.

- Generation of Secret Key by A
  $k = (Y_B)^{X_A} mod\ p$

- Generation of secret Key by B
  $k = (Y_A)^{X_B} mod\ p$

A METHOD FOR OBTAINING DIGITAL SIGNATURE AND PUBLIC-KEY CRYPTOGRAPHY

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences[5]:

1. Couriers or other secure means are not needed to transmit keys. Since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.

2. A message can be signed using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious application in ``electronics mail '' and ``electronic funds transfer'' system.

A message is encrypted by representing it as a number M. Raising M to a publicly specified power e. and then taking the remainder when the result is divided by publicly

specified product, n, of two large secret prime numbers p and q . Decryption is similar; only the different, secret, power d is used, where $e.d \equiv 1\ (mod(p-1).(q-1))$. The security of the system rests in part on the difficulty of factoring the published divisor, n.

The era of electronic mail may soon be upon us; we must ensure that two important properties of current paper mail system are preserved:

(a) Messages are private, and
(b) Message can be signed.
We demonstrate in this paper how to build these capabilities into an electronic mail system.

At the heart of the proposal is a new encryption method. This method provides an implementation of a`` public-key cryptosystem'', an elegant concept invented by Diffie and Hellman. They presented the concept but not any practical implementation of such a system.

PUBLIC-KEY CRYPTOSYSTEM

In a ``public –key cryptosystem '' each user places in a public file an encryption procedure E. that is, the public file is a directory giving the encryption procedure of each user. The user keeps the secret the details of his corresponding procedure D. These procedure have the following four properties:

(a) Deciphering the enciphered form of a message M yields M. formally,

$$D(E(M) = M. \qquad (1)$$

(b) Both E and D are easy to compute.
(c) By publicly revealing E the user does not reveal an easy way to compute D. This means that in practice only he can decrypt messages encrypted with E, or compute D efficiently.
(d) If a message M is first deciphered and then enciphered, M is the result. Formally,

$$E(D(M) = M. \qquad (2)$$

An encryption (or decryption) procedure typically consist of a general method and an encryption key. The general method, under control or the key, enciphers a message M to obtain the enciphered form of the message, called the cipher text C. Everyone can use the same general method; the security of a given procedure will rest on the security of the key. Revealing an encryption algorithm then means revealing the key.

When the user reveals E he reveals a very inefficient method of computing $D(C)$:testing all possible message M until one such that $E(M) = C$ is found. If property (c) is satisfied the number of such message to test will be so large that this approach is impractical.

A function E satisfying (a)-(c) is a`` trap-door one way function'',. If it also satisfies (d) it is a`` trap-door one way permutation." Diffie and Hellman introduced the concept of trap-door one-way function but did not present any examples. These function are called ``one-way'' because they are easy to compute in one direction but very difficult to compute in the other direction. They are called ``trap-door'' functions since the inverse function are in fact easy to compute once certain private ``trap-door'' information is known. A trap-door one way function which also satisfies (d) must be a permutation: every message is the ciphertext for some other message and every ciphertext is itself a permissible message. ( the mapping is ``one-to-one'' and ``onto''). Property (d) is needed only to implement ``signatures.''

Suppose that A and B (also known as Alice and Bob) are two users of a public-key cryptosystem. We will distinguish their encryption and decryption procedure with subscript: E_A, D_A, E_B, D_B.

PRIVACY:

Encryption is the standard means of rendering a communication private. The sender enciphers each message before transmitting it to the receiver. The receiver (but no unauthorized person) knows the appropriate deciphering function to apply to the received message to obtain the original message. An eavesdropper who hears the transmitter message hears only ``garbage''(the ciphertext) which makes no sense to him since he does not know how to decrypt it.

The large volume of personal and sensitive information currently held in computerized data banks and transmitted over telephone lines makes encryption increasingly important. In recognition of the fact that efficiently, high –quality encryption techniques are very much needed but are in short supply, the NATIONAL BUREAU has recently adopted a ``Data Encryption Standard'', developed at IBM. The new standard does not have property (c). needed to implement a public-key cryptosystem.

All classical encryption methods (including the NBS standard) suffer from the ``key distribution problem''. The problem is that before a private communication can begin. Another private transaction is necessary to distribute corresponding encryption and decryption keys to the sender and receiver, respectively. Typically a private courier is used to carry a key from the sender to the receiver. Such a practice is not feasible if an electronic mail system is to be rapid and inexpensive. A public-key cryptosystem needs no private courier; the keys can be distributed over the insecure communication channel.

How can Bob send a private message M to Alice in a public –key cryptosystem? First, he retrieves E_A from the public file. Then he sends her the enciphered message $E_A(M)$. Alice decipher the message by computing $D_A(E_A(M) = M$. By property (c) of the public-key cryptosystem only she can decipher $E_A(M)$. She can encipher a private response with E_B, also available in the public file.

Observe that no private transaction between Alice and Bob are needed to establish private communication. The only ``step up'' required is that each user who wishes to receive private communication must place his enciphering algorithm in the public file.

Two users can also establish communication private over an insecure communications channel without consulting a public file. Each user sends his encryption key to the other. Afterwards all messages are enciphered with the encryption key of the recipient, as in public-key system. An intruder listening in on the channel cannot decipher any messages. Since, it is not possible to derive the decryption keys from the encryption keys. (we assume that the intruder cannot modify or insert messages into the channel.)

A public-key cryptosystem can be used to ``bootstrap'' into a standard encryption scheme such as the NBS method. Once secure communication have been established the first message transmitted can be a key to use in the NBS scheme to encode all messages.

SIGNATURES:

If electronic mail system are to be replace the exciting paper mail system for business transaction ``signing'' an electronic message must be possible. The recipient of a signed message has proof that the message originated from the sender. This quality is stronger than mere authentication (where the recipient can verify that the message came from the sender); the recipient can convince a ``judge'' that the signer sent the message. To do so, he must convince the judge that he did not forge the signed message himself! In an authentication problem the recipient does not worry about this possibility, since he only wants to satisfy himself that the message came from the sender.

An electronic signature must be message-dependent, as well as signer-dependent. Otherwise the recipient could modify the message before showing the message-signature pair to a judge. Or he could attach the signature to any message whatsoever, since it is impossible to detect electronic ``cutting and pasting''.

To implement signature the public-key cryptosystem must be implemented with trap-door one-way permutation (i.e. have property (d)), since the decryption algorithm will be applied to unenciphered messages.

How can user Bob send Alice a`` signed'' message $M$ in a public-key cryptosystem? He first compute his ``signature'' S for message M using D_B:

$$S = D_B(M).$$

(Deciphering an unenciphered message ``make sense'' by property (d) of a public-key cryptosystem: each message is the ciphertext for some other message.) He then encrypt S using E_A (FOR PRIVACY), and sends the result $E_A(s)$ to Alice. He need not send M as well: it can be computed from S.

Alice first decrypts the ciphertext with D_A to obtain S. she knows who is the presumed sender of the signature(in this case, Bob); this can be given if necessary in plain text attached to S. she then extracts the message with the encryption procedure of the sender, in this case E_B (available on the public file):

$$M = E_B(S).$$

She now possesses a message-signature pair (M, S) with properties similar to those of signed paper document.

Bob cannot later deny having sent Alice this message, since no one else could have created $S = D_B(M)$. Alice can convince a ``judge'' that $E_B(S) = M$, so she proof that Bob signed the document.

Clearly Alice cannot modify M to a different version M', since then she would have to create the corresponding signature $S' = D_B(M')$ as well.

Therefore Alice has received a message ``signed'' by Bob, which she can `` prove'' that he sent. But which she cannot modify.(nor can she forge his signature for any other message.)

An electronic checking system could be based on a signature system such as the above. It is easy to imagine an encryption device in your home terminal allowing you to sign checks that get sent by electronic mail to payee. It would only be necessary to include a unique check number so that even if the payee copies the check the bank will only honor the first version it sees.

Another possibility arises if encryption device can be made fast enough: it will be possible to have a telephone conversation in which every word spoken is signed by the encryption device before transmission.

When encryption is used for signatures as above, it is important that the encryption device not be ``wired in'' between the terminal ( or computer) and the communication channel, since a message may have to be successively enciphered with several keys. It is perhaps more natural to view the encryption device as a ``hardware subroutine" that can be executed as needed.

We have assume above that each user can always access the public file reliably. In a ``computer network" this might be difficult; an ``intruder" might forge messages purporting to be from the public file. The user would like to be sure that he actually obtains the encryption procedure of his desired correspondent and not, say. The encryption procedure of the intruder. This danger disappears if the public file ``signs" each message it sends to a user. The user can check the signature with the public file's encryption algorithm $E_{PF}$. The problem of ``looking up" E_{PF} itself in the public file is avoided by giving each user a description of E_{PF} when he first shows up (in person) to join the public-key cryptosystem and to deposit his public encryption procedure. He then stores this description rather than ever looking it up again. The need for a courier between every pair of user has thus been replaced by the requirement for a single secure meeting between each user and the public file manager when the user joins the system. Another solution is to give each user, when he signs up, a book (like a telephone directory) containing all the encryption keys of users in the system.

OUR ENCRYPTION AND DECRYPTION METHODS.

To encrypt a message M with our method, using a public encryption key (e, n), proceed as follows. (here e and n are pair of positive integers.)

First, represent the message as an integer between 0 and $n-1$. (Breaking a long message into a series of blocks, and represent each block as such an integer.) Use any standard representation. The purpose here is not to encrypt the message but only to get it into the numeric form necessary for encryption.

Then, encrypt the message by raising it to the eth power modulo n. that is, the result (the ciphertext C) is the remainder when M^e is divided by n.

To decrypt the ciphertext, raise it to another power d, again modulo n. The encryption and decryption algorithms E and D are thus:

$$C \equiv E(M) \equiv M^e (mod\ n), \text{ for a message } M.$$

$$D(C) \equiv C^d\ (mod\ n), \text{ for a ciphertext } C.$$

Note that encryption does not increase the size of a message; both the message and the cipher text are integer in the range 0 to $n-1$.

The encryption key is thus the pair of positive integers $(e, n.)$ similarly, the decryption key is the pair of positive integers (d, n). Each user makes his encryption key public, and keeps the corresponding decryption key private. (these integers should properly be subscribed as in n_A, e_A and d_A, since each user has his own set. However, we will only consider a typical set, and will omit the subscripts.)

How should you choose your encryption and decryption keys, if you want to use this method?

You first compute n as the product of two primes p and q:

$$n = p.q.$$

These primes are very large, ``random" primes. Although you will make n public, the factor p and q will be effectively hidden from everyone else due to the enormous difficulty of factoring n. This also hides the way d can be derived from e.

You then pick the integer d to be a large, random integer which is relatively prime to $(p-1).(q-1)$. That is, check that d satisfies:

$$gcd(d, (p-1).(q-1)) = 1$$

(``gcd" means "greatest common divisor").

The integer e is finally computed from p, q, and d to be the "multiplicative inverse" of d, modulo $(p-1).(q-1)$. Thus we have

$$e.d \equiv 1\ (mod(p-1).(q-1)).$$

THE UNDERLYING MATHEMATICS:

We demonstrate the correctness of the deciphering algorithm using an identity due to Euler and Fermat for any integer (message) M which is relatively prime to n,

$$M^{\phi(n)} \equiv 1\ (mod\ n). \qquad (3)$$

Here $\phi(n)$ is the Euler totient function giving number of positive integer less than n which are relatively prime to n. For prime numbers p,

$$\phi(p) = p - 1.$$

In our case, we have by elementary properties of the totient function

$$\phi(n) = \phi(p).\phi(q) \qquad (4)$$

$$= (p-1).(q-1)$$

$$= n - (p+q) + 1$$

Since d is relatively prime to $\phi(n)$, it had a multiplicative inverse e in the ring of integers modulo $\phi(n)$:

$$e.d \equiv 1\ (mod\ \phi(n)). \qquad (5)$$

We now prove that equation (1) and (2) holds (that is, that deciphering works correctly if e and d are chosen as above). Now

$$D(E(M)) \equiv (E(M))d \equiv (Me)d\ (\ mod\ n\ ) = M^{e.d}\ (mod\ n)$$

$$E(D(\ M)) \equiv (D(M))e \equiv (Md)e\ (mod\ n\ ) = M^{e.d}\ (mod\ n)$$

And

$M^{e.d} \equiv M^{k.\phi(n)+1}\ (mod\ n)$ (for some integer K).

From (3) we see that for all M such that p does not divide M

$$M^{p-1} \equiv 1\ (mod\ p)$$

And since $(p-1)$ divides $\phi(n)$

$$M^{k.\phi(n)+1} \equiv M\ (mod\ p)$$

This is trivially true when $M \equiv 0\ (mod\ p)$, so that this equality actually holds for all M. Arguing similarly for q yields

$M^{k.\phi(n)+1} \equiv M\ (mod\ q)$

Together these last two equation imply that for all M,

$M^{e.d} \equiv M^{k.\phi(n)+1} \equiv M\ (mod\ n)$.

This implies (1) and (2) for all $M, 0 \leq M < n$. Therefore E and D are inverse permutations.

A How to Encrypt and Decrypt Efficiently[3]

Computing $M^e (mod\ n)$ requires at most $2.log_2(e)$ multiplications and, $2.log_2(e)$ divisions using the following procedure (decryption can be performed similarly using d instead of e):

Step 1: Let $e_k e_{k-1} \ldots e_1 e_0$ be the binary representation of e.

Step 2: Set the variable C to 1.

Step 3: Repeat steps 3a and, 3b for $i = k, k-1, \ldots .0$:

Step 3a. Set C to the remainder of C^2 when divided by n.

Step 3b. If $e_i = 1$, the set C the the remainder of $C.M$ when divided by n.

Step 4: Halt. Now C is the encrypted form of M.

This procedure id called "exponentiation by repeated squaring and multiplication". This procedure is half as good as the best; more efficient procedures are known. Knuth studies this problem in detail.

The fact that the enciphering and deciphering are identical leads to a simple implementation. (The whole operation can be implemented on a few special-purpose integrated circuit chips.)

A high-speed computer can encrypt a 200-digit message *M* in a few seconds: special purpose hardware would be much faster. The encryption time per block increases no faster than the cube of the number of digits in *n.*

B How to Find Large Prime Numbers

Each user (privately) choose two large random numbers p and, q to create his own encryption and decryption keys. These numbers must be large so that it is not computationally feasible for anyone to factor

$n = p.q$ (Remember that n, but not p or, q, will be in the public file).

We recommend using 100-digit (decimal) prime numbers p and, q, so that n has 200 digits.

To find a 100-digit "random" prime number generate (odd) 100-digit random numbers until a prime number is found. By the prime number theorem, about $(ln\ 10^{100})/2 = 115$ numbers will be tested before a prime is found.

To test a large number b for primality we recommend the elegant "probabilistic" algorithm due to Solovay and, Strassen. It picks a random number a from a uniform distribution on $\{1, \ldots . b-1\}$, and tests whether

$$gcd\ (a,b) = 1\ and, J(a,b) = a^{\frac{b-1}{2}} (mod\ b),$$

where $J(a,b)$ is the Jacobi symbol. If b is prime, the above eq^{n}. is always true. If b is composite, the above eq^{n}. will be false with probability at least ½. If the above eq^{n}. holds for 100 randomly choosen values of a then, b is almost certainly prime: there is a (negligible) chance of one in 2^{100} that b is composite. Even if a composite were accidentally used in our system, the receiver would probably detect this by noticing that decryption didn't work correctly. When b is odd, $a \leq b$, and, $gcd(a,b) = 1$, the Jacobi symbol $J(a,b)$ has a value in $\{-1, 1\}$ and can be efficiently computed by the program:

$$\mathrm{J}(a,b) = if\ a = 1 \text{ then 1 else}$$

$$\text{If } a \text{ is even, then } J(a/2,b).(-1)^{(b^2-1)/8}$$

$$\text{Else, } J(b\ (mod\ a), a).(-1)^{(a-1).(b-1)/4}$$

(The computations of $J(a,b)$ and $gcd(a,b)$ can be nicely combined too.) Note that this algorithm doesn't test a number for primality by trying to factor it. Other efficient procedures for testing a large number for primality are given.

To gain additional protection against sophisticated factoring algorithms, p and, q should differ in length by a few digits, both $(p-1)$ and, $(q-1)$ should contain a large number of prime factors, and $gcd(p-1, q-1)$ should be small. The latter condition is easily checked.

To find a prime factor p such that $(p-1)$ has a large prime factor, generate a large random prime factor u, then let p be the first prime in the sequence $i.(u+1)$ for $i = 2, 4, 6 \ldots$ (This shouldn't take too long.) Additional security is provided by ensuring that $(u-1)$ also has a large prime factor.

A high-speed computer can determine in several seconds whether a 100-digit number is prime, and can find the first prime after a given point in a minute or, two.

Another approach to finding large prime factors is to take a number of known factorization, add one to it, and test the result for primality. If a prime p is found its possible to prove that it really is prime by using the factorization of $p-1$. We omit a discussion of this since the probabilistic method is adequate.

C How to Choose d

Its very easy to choose a number d which is relatively prime to $\Phi(n)$. for example, any prime number greater than $max(p, q)$ will do. Its important that d should be chosen from a large enough set so that a cryptanalyst can't find it by direct search.

D How to Compute e from d and $\Phi(n)$

To compute e, use the following variation of Euclid's algorithm for computing the greatest common divisor of $\Phi(n)$ and d. Calculate $gcd(\Phi(n), d)$ by computing a series $x_0, x_1, x_2 \ldots, where\ x_0 \equiv \Phi(n),\ x_1 = d$

and, $x_{i+1} \equiv x_{i-1}(mod\ xi)$, until an x_k equal to zero is found. Then $gcd(x0, x1) = x_{k-1}$. Compute for each x_i numbers a_i and b_i such that $x_i = a_i . x_0 + b_i . x_1$.

if $x_{k-1} = 1$, then b_{k-1} is the multiplicative inverse of $x_1(mod\ x_0)$, Since k will be less than $2log_2(n)$, this computation is very rapid.

If e turns out to be less than $log_2(n)$ start over by choosing another value of d. This guarantees that every encrypted message (except $M = 0$ or, $M = 1$) undergoes some "wrap-around" (reduction modulo n).

CHAPTER 3

SOME REMARKS ON LUCAS-BASED CRYPTOSYSTEMS

FIBONACCI NUMBER

In mathematics the Fibonacci numbers or Fibonacci series or Fibonacci sequence are the numbers in the following integer sequence:

0,1,1,2,3,5,8,13,21,34,55,89,144…

Or alternatively

1,1,2,3,5,8,13,21,34,55,89,144…

By definition, the first two numbers in the Fibonacci sequence are 0 and 1(alternatively 1 and 1) and each subsequent number is the sum of the previous two.

In mathematical terms, the sequence F_n of Fibonacci number is defined by the recurrence relation

$$F_n = F_{n-1} + F_{n-2}$$

With seed values

$$F_0 = 0$$

$$F_1 = 1$$

In the first term form or

$$F_1 = 1$$

$$F_2 = 1$$

In the second form.

LUCAS NUMBER

The Lucas number or Lucas series are an integer sequence named after the mathematician Francois Eduardo Anatole Lucas (1842-1891), who studied both that sequence and the

closely related Fibonacci number. Lucas number and Fibonacci number form complementary instances of Lucas sequence.

3.1 Definition.

Similarly to the Fibonacci numbers, each Lucas number is defined to be the sum of its two immediate previous terms that is it is a Fibonacci integer sequence. However, the first two Lucas numbers are L_0=2 and L_1=1 instead of 0 and 1, and the properties of Lucas number are therefore somewhat different from those of Fibonacci numbers.

The Lucas numbers may thus be defined as follows:

L_n= 2 if n=0

1 if n=1

$L_{n-1} + L_{n-2}$ if n>0.

The sequence of Lucas number begins:

2,1,3,4,7,11,18,29,47,76,123…

EXTENSION TO NEGATIVE INTEGERS

Using $L_{n-2} = L_n - L_{n-1}$, one can extend the Lucas numbers to negative integers to obtain a doubly infinite sequence:

…-11,7,-4,3,-1,2,1,3,4,7,11…(terms L_n for -5≤n≤5 are shown)

The formula for terms with negative indices in this sequence is $L_{-n} = (-1)^n L_n$

LIST OF FIBONACCI NUMBERS

The first 21 Fibonacci numbers F_n for n=0,1,2…20 are

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = 1$$

$$F_3 = 2$$

$$F_4 = 3$$

$$F_5 = 5$$

$$F_6 = 8$$

$$F_7 = 13$$

$$F_8 = 21$$

$$F_9 = 34$$

$$F_{10} = 55$$

$$F_{11} = 89$$

$$F_{12} = 144$$

$$F_{13} = 233$$

$$F_{14} = 377$$

$$F_{15} = 610$$

$$F_{16} = 987$$

$$F_{17} = 1597$$

$$F_{18} = 2584$$

$$F_{19} = 4181$$

$$F_{20} = 676$$

The sequence can also be extended to negative index `n' using the rearrangement relation

$$F_{n-2} = F_n - F_{n-1}$$

Which yields the sequence of negafibonacci number satisfying $F_{-n} = (-1)^{n+1} F_n$

Thus the bidirected sequence is

$$F_{-8} = -21$$

$$F_{-7} = 13$$

$$F_{-6} = -8$$

$$F_{-5} = 5$$

$$F_{-4} = -3$$

$$F_{-3} = 2$$

$$F_{-2} = -1$$

$$F_{-1} = 1$$

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = 1$$

$$F_3 = 2$$

$$F_4 = 3$$

$$F_5 = 5$$

$$F_6 = 8$$

$$F_7 = 13$$

$$F_8 = 21$$

LUCAS PRIME

A Lucas prime is a Lucas number that is prime. The first few Lucas prime are:

2,3,7,11,29,47,199,521,2207,3571,9349,…

If $L_n$is prime then` n' is either 0, prime or a power of 2. L_2m is prime for m=1,2,3 and 4 and no other known value of m.

INTEGER SEQUENCE

In mathematics, an integer sequence is a sequence of integers. An integer sequence may be specified explicitly by giving a formula for its nth term or implicitly by giving a relationship between its terms. For example 0,1,1,2,3,5,8,13… (the Fibonacci sequence) is formed by starting with `0' and `1 ' and then adding any two consecutive term to obtain the next one; an implicit description. The sequence 0,3,8,15,… is formed according to the formula $n^2 - 1$ for the nthterm;an explicit definition.

LUCAS SEQUENCE[6]

Let P and Q be integers and let α be a root of $x^2 - Px + Q = 0$ in the field $Q\sqrt{\Delta}$. [$Q\sqrt{\Delta}$= $\{a + b\sqrt{\Delta} | a, b \in Q\}$.

Where $\Delta = P^2 - 4Q \in \mathbb{Z}$ is assumed to be a non-square (but not necessary square free). Then α is an element of the ring of δ_Δ of the quadratic field $Q\sqrt{\Delta}$, and there exists $v = v(\alpha)$ and $u = u(\alpha)$ such that

$$\alpha = \frac{v + u\sqrt{\Delta}}{2}$$

$\because x^2 - Px + Q = 0$, then $\alpha = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

$$\alpha = \frac{p \pm \sqrt{\Delta}}{2}$$

So, $= \frac{p+\sqrt{\Delta}}{2}$, $\beta = \frac{p-\sqrt{\Delta}}{2}$

Infact, for every $k \geq 1$ it holds that $2\alpha^k \in \mathbb{Z}[\sqrt{\Delta}]$, and we can write $\alpha^k = \frac{v_k + u_k\sqrt{\Delta}}{2}$, for certain integers

$v_k = v(\alpha^k) = v_k(\alpha)$ and $u_k = u(\alpha^k) = u_k(\alpha)$

Choosing $\alpha = \frac{p+\sqrt{\Delta}}{2}$ and its conjugate $\beta = \bar{\alpha} = \frac{p-\sqrt{\Delta}}{2}$

We find that $v_1(\alpha) = v(\alpha) = P$

$$u_1(\alpha) = u(\alpha) = 1$$

And $\alpha^2 = (\frac{p+\sqrt{\Delta}}{2})^2 = \frac{p^2+\Delta+2P\sqrt{\Delta}}{4}$

$v_2(\alpha) = v(\alpha^2) = \frac{p^2+\Delta}{2}$, $u_2(\alpha) = u(\alpha^2) = P$

and by induction, we see that v_k and u_k are given by the recurrence relation

$$u_{k+2} = u_{k+2}(P, Q) = P_{uk+1} - Q_{uk}, u_1 = 1 , u_0 = 0$$

$$v_{k+2} = v_{k+2}(P, Q) = P_{vk+1} - Q_{vk}, v_1 = P , V_{0,} = 2$$

3.1 REMARKS.

Thus the v_k, u_k may be seen as the `co-efficient of the powers of α that may b computed by the above recurrence relation. Knowing v_k and u_k implies knowledge of α^k , which immediately ties the problems of determining k from v_k and u_k to the discrete logarithm of α^k with respect to the base α.

Depending on which view we like to stress we will write $u_k(\alpha)$ or $v_k(P, Q)$ and these are related via $\alpha = P + \sqrt{P^2} - 4Q$

Of the many relation between the u_k , v_k we derive a few that are relevant for what is to follow. The first lemma deals with the `$u$' and `v' of the conjugate, traces and norms of powers.

1 LEMMA. For every α and every $k \geq 0$.

(i) $v_k(\beta) = v_k(\alpha)\ and\ u_k(\beta) = -u_k(\alpha).$
(ii) $\alpha^k + \beta^k = v_k(\alpha) = v_k(\beta)$
(iii) $\alpha^k \beta^k = Q^k = \frac{{v_k}^2(\alpha) - \Delta {u_k}^2(\alpha)}{4}$

Proof.(i) $\alpha = \frac{p+\sqrt{\Delta}}{2}$

From these we have

$v_1(\alpha) = P \ \ and\ u_1(\alpha) = 1$

So ,$v_1 = v_1(\alpha) = P\ and\ u_1 = u_1(\alpha) = 1$ (1)

$$\beta = \frac{p - \sqrt{\Delta}}{2}$$

$$v_1(\beta) = P \ \ and\ u_1(\beta) = 1$$

So , $v_1 = v_1(\beta) = P\ and\ u_1 = u_1(\beta) = -1$ (2)

From (1) and (2) we have

$$v_1(\alpha) = v_1(\beta) \text{ and}$$

$$u_1(\beta) = -u_1(\alpha)$$

Thus

$$v_k(\alpha) = v_k(\beta)$$

$$u_k(\beta) = -u_k(\alpha)$$

.

(ii)

$\alpha^k = \frac{v_k + u_k\sqrt{\Delta}}{2}$

$\beta^k = \frac{v_k - u_k\sqrt{\Delta}}{2}$

$\Rightarrow \frac{v_k + u_k\sqrt{\Delta}}{2} + \frac{v_k - u_k\sqrt{\Delta}}{2}$

$= \frac{v_k + u_k\sqrt{\Delta} + v_k - u_k\sqrt{\Delta}}{2}$

$= \frac{2v_k}{2}$

$= v_k$

Therefore $v_k(\alpha) = v_k = v_k(\beta)$

(iii) $\alpha^k.\beta^k = \alpha^k.\frac{v_k - u_k\sqrt{\Delta}}{2}$

$= (\frac{v_k + u_k\sqrt{\Delta}}{2})(\frac{v_k - u_k\sqrt{\Delta}}{2})$

$= \frac{{v_k}^2 - {u_k}^2\sqrt{\Delta}}{4}$

$\alpha^k.\beta^k = \frac{{v_k}^2(\alpha) - {u_k}^2(\alpha)\sqrt{\Delta}}{4}$

2 LEMMA. :- for all $K \geq m \geq 0$

$$v_{k+m=}v_k v_m - Q^m v_{k-l}$$

Proof. $V_k(\alpha) = V_k(\beta) = \alpha^k + \beta^k$ (from lemma 1)

$$\alpha^k \beta^k = Q^k = \frac{{v_k}^2(\alpha) - \Delta{u_k}^2(\alpha)}{4}$$

$$v_k = \alpha^k + \beta^k$$

$$V_m = \alpha^m + \beta^m$$

$$Q^m = \alpha^m\beta^m$$

$$V_{k-m} = \alpha^{k-m} + \beta^{k-m}$$

$v_k v_m - Q^m v_{k-l} = (\alpha^k + \beta^k)(\alpha^m + \beta^m) - \alpha^m \beta^m (\alpha^{k-m} + \beta^{k-m})$

$= \alpha^{k+m} + \alpha^k \beta^m + \beta^k \alpha^m + \beta^{k+m} - (\alpha^m \beta^m . \alpha^{k-m} + \alpha^m \beta^m . \beta^{k-m})$

$= \alpha^{k+m} + \beta^{k+m}$

$= v_{k+m}(\alpha)$

3 LEMMA. for $K \geq 1$

(i) $u_{2k} = u_k v_k$ and $v_{2k} = {v_k}^2 - 2Q^k$.

(ii) $u_{2k+1} = (P_{u2k}(\alpha) + v_{2k}(\alpha)/2$ and $v_{2k+1} = (\boldsymbol{\Delta}\, u_{2k}(\alpha) + P_{v2k}(\alpha)/2.$

Proof:- (i) $v_{2k} = {v_k}^2 - 2Q^k$

${v_k}^2 - 2Q^k = (\alpha^k + \beta^k)^2 - 2(\alpha^k \beta^k)$

$= v_{2k}$

$u_{2k} = u_k v_k$

$\Rightarrow (\alpha^k \alpha^k)^2 = (\alpha^k)^2$

$= \frac{(v_k(\alpha) + u_k(\alpha)\sqrt{\Delta})^2}{2}$

$= \frac{\frac{({v_k}^2(\alpha) + {u_k}^2(\alpha)\Delta)}{2} + v_k(\alpha) u_k(\alpha)\sqrt{\Delta}}{2}$

Comparing the co-efficient we have

$(\alpha^k)^2 = (\alpha^{2k}) = u_{2k} = v_k(\alpha) u_k(\alpha)$

$u_{2k} = v_k u_k.$

(ii) $\alpha^{2k+1} = \alpha^{2k} . \alpha$

$= \alpha . \alpha^{2k}$

$= \left(\frac{P + \sqrt{\Delta}}{2}\right)\left(\frac{V_{2K(\alpha)} + U_{2K}(\alpha)\sqrt{\Delta}}{2}\right)$

$$= \frac{Pv_{2k}(\alpha) + u_{2k(\alpha)}\Delta + P_{u2k(\alpha)} + v_{2k(\alpha)\sqrt{\Delta}}}{2}$$

Comparing the co-efficient of v_{2k+1} and u_{2k+1} we have

$$u_{2k+1} = (P_{u2k}(\alpha) + v_{2k}(\alpha)/2 \text{ and } v_{2k+1} = (\boldsymbol{\Delta}\, u_{2k}(\alpha) + P_{v2k}(\alpha)/2.$$

4 LEMMA. For every P and Q

$$v_{km}(P,Q) = v_k(u_m(P,Q), Q^m) \; and$$

$$u_{km}(P,Q) = u_k(v_m(P,Q), Q^m)u_m(P,Q)$$

Proof:- Let $\alpha^m = \frac{v_m + u_m\sqrt{(\Delta)}}{2}$

$$= \frac{v_m + \sqrt{u_m}^2(\Delta)}{2}$$

$$= \frac{v_m + \sqrt{v_m}^2 - 4Q^m}{2}$$

By lemma (1) we have

$$P' = v_m(P,Q)$$

$$Q' = Q^m$$

$$\alpha^m = \frac{[P' + \sqrt{(P'^2 - 4Q')}}{2}$$

$$(\alpha^m)^k = \frac{v_k(P',Q') + u_k(P',Q')\sqrt{(P'^2 - 4Q')}}{2}$$

$$\alpha^{km} = \frac{v_k(v_m(P,Q), Q^m) + u_k(v_m(P,Q), Q^m)\, u_{m+}(P,Q)\sqrt{(\Delta)}}{2}$$

Comparing the co-efficient we have

$$v_{km}(P,Q) = v_k(u_m(P,Q), Q^m)$$

$$u_{km}(P,Q) = u_k(v_m(P,Q), Q^m)u_m(P,Q)$$

LUC

PUBLIC KEY SYSTEM (LUC). Each user publishes the product n of two large primes p and q, and an index e with $\gcd\big(e,(p^2-1)(q^2-1)\big)=1$.The corresponding d such that $de \equiv mod(p^2-1)(q^2-1)$ is kept secret.

A message m is an integer satisfying $1 \le m \le n-1$ with $\gcd(m,n)=1$. To encrypt a message $m$meant for some user, one looks up the user's n and e, and computes the encrypted message $y = v_e(m,1)mod\ n$ that is ,P is equal to the message, and $Q=1$. This computation can be carried out using the recurrence given in lemma 2 in $O(\log e)$ elementary operations on integers modulo n. To decrypt the message, the user calculates

$$v_d(y,1) \equiv v_d(v_e(m,1),1) \equiv v_{de}(m,1) \equiv m\ mod\ n$$

(By lemma 4). The final identity holds because $\alpha^{de} \equiv \alpha$ modulo both $p$and q.

Alternatively, to use LUC as a signature scheme, the user's signature on a message m equals $v_d(m,1)mod\ n$, which can be verified by checking that $v_e(v_d(m,1),1) \equiv m\ mod\ n$.

REFERENCES

1. Diffie. W. and Hellman, M. New directions in cryptography. IEEE Trans. Inform. Theory IT-22, (1976),644-654.
2. Niven, I and Zuckerman, H.S. An introduction to the theory of numbers. Wiley, New York 1972.
3. Knuth, D.E The art of computer programming. V0l-2:Seminumerical Algorithms. Addison-Wesley, Reading Mass., 1969.
4. Elbirt, A.J., Understanding and Applying Cryptography and Data Security, CRC press.
5. Rivest,R.L., Shamir.A. and Adleman.L., A method for obtaining Digital Signatures and Public-key Cryptosystem, Comm. ACM 21(1978), 120-126.
6. Bleichenbacher Daniel., Bosma W. and Lenstra , A.K., Some Remarks on Lucas-Based Cryptosystems. Advances in cryptology-CRYPTO 95 LNCS,(1995), 386-396.
7. Kumanduri,R. and Romero.C., Number Theory with computer applications.