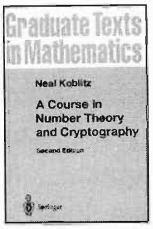
A Course in Number Theory and Cryptology

Rajat Tandon



A Course in Number Theory and Cryptology (2nd ed.), by Neil Koblitz Graduate Texts in Mathematics 114, Springer Verlag, Berlin, 1994, pp. 236, Price. ~Rs.475/- (Special edition to be sold in India, akistan, Bangladesh, Srilanka &Nepal)

Cryptography is the study of methods of sending messages in disguised form so that only the intended recipients can remove the disguise and read the message. The process of encoding messages is known as encryption and the process of decoding them decryption. Formerly encrypted messages were exchanged between parties that were known to each other; for example, divisions of the same army or embassies of the same or allied countries. But today I may want to communicate confidentially with a doctor who I have not met but who has a website and consults on the net. I would want only the doctor and not other 'nosy parkers' to know about my medical problems. I would then have to communicate with the doctor by an encrypted message which only he can decrypt. However, I don't know the doctor so I cannot use a prearranged code. The method of encrypting to the doctor must be *Public* though only he should be able to decrypt. Thus the need for a *Public key cryptogram* as opposed to a classical cryptogram. (For a fascinating history of the subject see the book by Simon Singh [1]).

Cryptography refers to a *systematic* way of disguising messages. For instance, two parties could decide to communicate within a vocabulary of a basic (and predecided) 500 words and agree upon a substitution of these 500 words by some 500 symbols which they learn by heart. Then all messages would be exchanged with these 500 symbols. This is not cryptography. On the other hand if we agree that the letters A-Z will correspond, in order, to the 26 numbers 0-25 so that YES becomes 24, 4, 18; we add 5(modulo 26) to each of these numbers and then reconvert to letters so

$$\text{YES} \rightarrow 24, 4, 18 \rightarrow 3, 9, 23 \rightarrow \text{DJX}$$

The code for YES is thus DJX. This is a cryptogram. To decode we must reconvert the letters to numbers, substract 5(modulo 26) from these numbers and then reconvert to letters. This example contains the essence of a cryptogram – we first make letters (or pairs of letters, or triples of letters, etc.) correspond to elements of a finite set S of mathematical objects (numbers, vectors, elements of a finite field, points on an elliptic curve!); then we apply a 'jumbling' bijective function $f: S \rightarrow S$ and finally we reconvert to letters (or pairs of letters, or triples of letters, etc.). In the example given above $S=\{0, 1, 2, \ldots, 25\}$ and $f: S \rightarrow S$ is the function $f(x)=x+_{26}5$, where $+_{26}$ is addition modulo 26. The decoding function is $f^{-1}(x)=x+_{26}(-5)$. Notice, once we know the encoding function f it is easy to compute the decoding function f^{-1}.

In modern cryptosystems, we require the

coding function f_A, to a particular person A in the communication network, to be public so that anyone can send a coded message to A by using f_A but only A should be able to decode the message, i.e., only A should be able to compute f_A^{-1}. A function f_A of this type, i.e., even a knowledge of f_A does not guarantee that one can compute f_A^{-1} in a reasonable amount of computer time, is called a trapdoor function and it is the existence of these functions that allow us to have public key cryptograms. The definition of a trapdoor function is not a mathematical one, it is empirical. The 'reasonable amount of computer time' depends on the state of technology (both hardware and software) at a given time. What is trapdoor today may not be trapdoor tomorrow. It would be interesting to find functions that are provably trapdoor (according to some reasonable definition) but this problem is a challenging one. In recent times there have been claims to the conceptualisation of such functions.

The most commonly used cryptosystem is the one discovered by Rivest, Shamir and Adleman [2] known as the RSA system. It exploits the idea that a composite number n with large (of the order of 200 digits, say) prime factors is hard to factorise, i.e., we do not have a polynomial time algorithm to do so. This means that we do not have an algorithm to factor n for which the time taken on a computer would be a polynomial function of the number of digits of n (not n itself). In this system each person A in the communication network chooses two large primes p_A and q_A and computes $n_A = p_A\, q_A$ and this product is revealed to the public. He/she then choses an integer e_A, coprime to $\phi(n_A) = (p_A - 1)(q_A - 1)$, and then computes $d_A = e_A^{-1} \pmod{\phi(n_A)}$. e_A is also declared to the public but d_A is kept secret. Notice that a person cannot compute d_A unless he/she knows $\phi(n_A)$ and for this p_A must be known. The set S for sending encrypted messages to A is the set of integers modulo n_A and $f_A(x) = x^{e_A} \pmod{n_A}$. A simple calculation shows (using Euler's theorem – see *Box* 1) that $f_A^{-1}(x) = x^{d_A} \pmod{n_A}$. e_A and n_A are public so anyone can send a message to A but only A can decrypt the message.

This shows that one of the number-theoretic problems of interest to a cryptologist is: given a large odd number n, how do you determine whether n is a prime or not? For instance, it is known (Fermat's little theorem – see *Box* 1) that if p is a prime number and a is any number not divisible by p then p must divide $a^{p-1} - 1$. Hence given an odd number n, if we can find some a, coprime to n, such that

Box 1.

Let Z_n^* be the set of natural numbers in the range $0 < x < n$ which are coprime to n. This is a group under the binary operation of multiplication modulo n and its order is denoted by $\phi(n)$. If, for instance, $n = pq$ where p and q are primes then it is easy to see that $\phi(n) = (p-1)(q-1)$. Now in any finite group G of order m, if $a \in G$, then $a^m = e$. Hence if a is coprime to n we have $a^{\phi(n)} \equiv 1 \pmod{n}$. This is Euler's theorem. If n is a prime, we get Fermat's little theorem.

$n \not| (a^{n-1} - 1)$ then clearly n is not a prime. This gives a test for compositeness. It is then natural to ask whether the converse of Fermat's little theorem is true, i.e., given that $n \mid (a^{n-1}-1)$ for all a coprime to n does it follow that n is a prime. The answer is NO and composite numbers that have this property are known as Carmichael numbers. Check that 561 is one such number. It was not known whether there are infinitely many such Carmichael numbers till 1994 when the question was answered in the affirmative [3]. Primality testing and factorisation is a fascinating area of research and Koblitz's book gives several primality tests and known algorithms for factorisation. Of course, none of these are polynomial time algorithm.

Another idea that is exploited in creating a trapdoor function is the following : In a cyclic group G (of large order) with generator a, given a random $y \in G$, it is, in general, computationally difficult to find an n such that $y=a^n$. This is known as the discrete log problem. Koblitz gives several examples of cryptosystems based on this idea. The group G should be one in which one can make explicit computations. It could be the muliplicative group of non-zero elements of a finite field (which is always cyclic) or a cyclic subgroup of the group of points on an elliptic curve. An elliptic curve over a field F of characteristic not 2 or 3 is basically a curve of the form $Y^2=X^3+aX+b$ with $a, b \in F$ and such that the cubic X^3+aX+b has no repeated roots in an algebraic closure of F.

Let $E=\{(x, y) \in F \times F \mid y^2=x^3+ax+b\}$. It is one of the beautiful facts of 'nature' that if we adjoin to E a symbol O then we can define on the extended set $\widetilde{E}$ a binary operation with respect to which $\widetilde{E}$ becomes an abelian group with the symbol O acting as the identity. If F is a finite field then $\widetilde{E}$ is a finite group and cryptologists have exploited the discrete log idea in cyclic subgroups of $\widetilde{E}$ of large order.

I have been introducing the subject of cryptography to students of mathematics and computer science using Koblitz's book for the last four years. As a prerequisite, I have been assuming that the students know only some elementary group theory and elementary real analysis. For a student who has already covered a book on algebra at the level of Herstein's *Algebra* or Artin's *Algebra*, the book reads like a novel and is easy to understand. For others, there is enough revision of the algebra required in the book to make the subject comprehensible if not comfortably understandable. The latter state is attained with a little effort. A course based on the book can thus be attended not only by students of mathematics but also students of applied mathematics, statistics, computer science and physics who have had some exposure to group theory at the undergraduate level, but no such exposure at the post graduate level.

The book requires some knowledge of finite fields and the ring of integers modulo n. The latter is today covered in class 12 in the ISC, CBSE and most state board examinations. I found that the best approach is to define an algebraically closed field, assume that every

field is contained in such a field and then, using this fact, develop the necessary field theory.

Students must be encouraged to do the exercises in the book. An algorithm is not really understood till it is performed several times. Ideally, if the students know a programming language like C they should be encouraged to write programs for the various algorithms given in the book. The material in the book can also be taught without a computer but with just a calculator. The examples are then less realistic but contain the essence of the ideas behind the various algorithms.

Koblitz's book has been pathbreaking in exposing the new and exciting subject of cryptography to mathematicians (number theorists, in particular). They suddenly find that they can do some interesting and challenging mathematics and still be fashionably useful in doing so!

Suggested Reading

[1] Simon Singh, *The Code Book*, The Fourth Estate, London, 1999.

[2] L M Adleman, R L Rivest and A Shamir, A method for obtaining digital signature and public-key cryptosystems, *Comm. of the ACM*, Vol. 21, pp.120-126, 1978.

[3] W R Alford, A Granville and C Pomerance, There are infinitely many Carmichael Numbers, *Annals of Math.*, Vol.140, pp.703-722, 1944.

Rajat Tandon, Department of Mathematics and Statistics, University of Hyderabad, Hyderabad 500 046, India. Email: rtsm@yohyd.erent.in

Mathematics and Music

Polyrhythm is the practice of playing two lines of music in two different rhythms. For example, Brahms was especially fond of the combination where one instrument plays three notes in the same time that another plays just two. Modern composers ventured further. Conlon Nancarrow despaired of finding players who could handle the complexities of his rhythms and turned to composing for a pair of player pianos. Once freed from human limitations, he could make really daring leaps, culminating in his study #33: "Canon – $\sqrt{2}$ /2". It was his first one involving irrational tempo relations; each piano is moving along at a fixed speed, where the ratio of the two speeds is $\sqrt{2}$ /2!