.

Primality, factorisation, cryptography and elliptic curves

B.Sury

Stat-Math Unit
Indian Statistical Institute
Bangalore

Expanded version of lectures given in
RRI in Jan 2009 and CUSAT in May 2005

Introduction

Gauss considered the problem of efficiently determining whether a given natural number is prime or not a very important one. He seemed to view it almost as an embarrassment that despite all sorts of advances in number theory, the basic question remained unanswered satisfactorily. With the advent of number-theoretic tools in cryptography, the problem has gained even more importance. Several years of concentrated attempts have led us now to powerful algorithms to check primality. Incidentally, Gauss also included factorizing a number efficiently as a basic problem and this is still unsolved which remains a boon as far as cryptography is concerned! It is a matter of pride to Indians that the first polynomial-time primality algorithm has been discovered by three Indians recently. Let us travel along the path towards this algorithm gently now so that we can sight various little gems along the way.

1 Simple primality tests

What is that sets apart primes from non-primes? Of course, we are not looking for tests which just re-express the definition; for instance, we do not want to test for factors from 2 to the number. We could test for factors until the square root of the number. But, this takes too many steps; we shall see that we want the number of steps taken to be only something like the logarithm of the number, if possible. Ok, coming back to the question, what distinguishes primes from composites? Here is one :
For a natural number n, the various binomial coefficients $\binom{n}{1}, \binom{n}{2}, \cdots, \binom{n}{n-1}$ are all multiples of n if and only if n is prime.
The proof goes as follows. Now $\binom{n}{r} = \frac{n(n-1)\cdots(n-r+1)}{r(r-1)\cdots 1}$. Of course n divides $n(n-1)\cdots(n-r+1) = \binom{n}{r} r!$ for every n. If n is a prime, then it cannot divide $r!$ because it does not divide any of the terms in the product $r(r-1)\cdots 1$ (this is where we use n is a prime !). Thus, in this case n would have to divide $\binom{n}{r}$. Conversely, if n is not a prime, look at any prime p dividing it. If p^k is the highest power of p dividing n, then we see that p^k cannot divide $\binom{n}{p}$. Why ? $\binom{n}{p} = \frac{n(n-1)\cdots(n-p+1)}{p(p-1)\cdots 1}$ has a single power of p in the denominator

while the numerator has exactly one term divisible by p which is the top term n. Thus, the power of p dividing this quotient is $k-1$.

This simple property characterizing primes is very useful. This is the key fact behind the first important result known as Fermat's little theorem which was first proved by Euler. To discuss it succinctly, let us first introduce the notion (and the notation of) "congruence" (both due to Gauss) :
$a \equiv b$ modulo n to read 'a is congruent to b modulo n' if $a-b$ is a multiple of n. That is, $a-b=nm$ for some (positive or negative) integer m. The advantage of this is that one can work with 'congruence modulo n' as though it were an equality. For instance, $a \equiv b$ modulo n and $a' \equiv b'$ modulo n (same n !) implies $a \pm a' \equiv b \pm b'$ and $aa' \equiv bb'$ modulo n. This is the basis of setting up many divisibility tests like the familiar tests for $3, 9, 11$ etc.
Using the above characterization of primes in the congruence notation, we have by binomial expansion :
For integers a, b and primes p we have $(a+b)^p \equiv a^p + b^p$ mod p.
So, the p-th power map respects addition (and obviously multiplication) mod p. In particular, we get :

Fermat's little theorem.
For any prime p and any integer n, we have $n^p \equiv n$ modulo p. Equivalently, for any integer a relatively prime to p, we have $a^{p-1} \equiv 1$ modulo p.
In other words, on the set $\mathbf{Z}_p$ of integers $0, 1, \cdots, p-1$ mod p, the p-th power map is the identity.

Wilson's congruence.
$(n-1)! \equiv -1$ modulo n if and only if n is prime.
The assertion that the congruence cannot hold for composite n is clear because a proper divisor of n is a term in $(n-1)!$. For prime n, this just follows from Fermat's little theorem. In fact, for any prime p, the congruence $X^{p-1} - 1 \equiv 0$ modulo p has solutions $1, 2, \cdots, p-1$. Thus,
$X^{p-1} - 1 - (X-1)(X-2)\cdots(X-(p-1))$ is a polynomial whose coefficients are all equal to 0 modulo p. A comparison of the constant terms of the two polynomials gives us Wilson's congruence.
Here is a different way to see Wilson's congruence for a prime p. Firstly, if $1 \leq a < p$, then a has a 'multiplicative inverse modulo p'; that is, there is some $b < p$ with $ab \equiv 1$ mod p. This is because the smallest natural number of the form $au + pv$ as u, v vary over all integers must be their GCD (by the division algorithm !) Therefore, one has u (which can be taken to be between

1 and $p-1$) such that $au+pv=1$. This is what is asserted above. In other words, we have shown that the integers $1, 2, \cdots, p-1$ form a 'group' under multiplication modulo p. Now, $(p-1)!$ is a product of all a's between 1 and $p-1$ and, mod p, each term cancels with its multiplicative inverse *excepting those special elements which are their own inverses !* Now, such a special a satisfies $a^2 \equiv 1 \bmod p$; that is, p divides $a^2-1=(a+1)(a-1)$. Now, use p is a prime to conclude that the only two such special elements are 1 and $p-1$. Therefore, $(p-1)! \equiv p-1 \bmod p$ - Wilson's congruence !

Here is yet another way of proving the congruence.
For any natural number d we have by inclusion-exclusion that

$$d! = \sum_{r=0}^{d-1} (-1)^r \binom{d}{r} (d-r)^d.$$

Another way of seeing this identity is as follows. For a function f, one has its 'forward difference' Δf defined as $(\Delta f)(x) = f(x+1) - f(x)$. If $\Delta^r f$ denotes Δ iterated r times, it follows by induction on n that

$$(\Delta^n f)(x) = \sum_{r=0}^{n} (-1)^r \binom{n}{r} f(x+n-r).$$

Let f be a polynomial of degree $d \geq 1$. Then Δf is a polynomial of degree $d-1$. Of course, if f is a constant, Δf is the zero function. Therefore, if $n > d$, we have $(\Delta^n f)(x) = 0 \ \forall \ x$.
Further, $\Delta^d f$ is the constant $d! a_d$ - this is seen once again by induction - this time on d. Here, a_d is the top coefficient of f.
Writing this for the polynomial $f(x) = x^d$ gives us

$$d! = \sum_{r=0}^{d} (-1)^r \binom{d}{r} (x+d-r)^d \quad \forall \quad x.$$

In particular,

$$d! = \sum_{r=0}^{d-1} (-1)^r \binom{d}{r} (d-r)^d.$$

Thus, we have another way of obtaining the identity. In this identity, let us take $d = p-1$ for some prime $p > 2$. We get

$$(p-1)! = \sum_{r=0}^{p-2} (-1)^r \binom{p-1}{r} (p-1-r)^{p-1}.$$

Reading it modulo p, by Fermat's little theorem, we obtain

$$(p-1)! \equiv \sum_{r=0}^{p-2} (-1)^r \binom{p-1}{r} \mod p.$$

However, $\sum_{r=0}^{p-2} (-1)^r \binom{p-1}{r} = -1$ since

$$(-1)^{p-1} + \sum_{r=0}^{p-2} (-1)^r \binom{p-1}{r} = \sum_{r=0}^{p-1} (-1)^r \binom{p-1}{r} = (1-1)^{p-1} = 0.$$

Thus, we obtain Wilson's congruence $(p-1)! \equiv -1 \bmod p$ for any prime p. Wilson's characterizing congruence is also impractical as it involves computing $(n-1)!$

Gauss proved the beautiful fact that for any prime p, there is always some $a < p$ of 'order' $p-1$ (that is, such that $p-1$ is the *smallest* n for which $a^n \equiv 1$ modulo p). Any such integer a would be called a 'primitive root mod p'. A consequence of the existence of a primitive root a mod p would be that the various powers $a, a^2, \cdots, a^{p-1}$ are all distinct modulo p (and hence, they are $1, 2, \cdots, p-1$ in some order !) Here is a nice exercise. Find the sum mod p of the primitive roots mod p.
Let us prove Gauss's theorem now.
We claim that the number of solutions of $a^n \equiv 1$ modulo p for any $n < p-1$ is at the most n in the set of integers modulo p. Indeed, let $a_0, a_1, \cdots, a_n$ be distinct elements each satisfying $a_i^n = 1$. Thus, $a_0, a_1, \cdots, a_n$ can be viewed as positive integers (all $< p$) such that p divides each $a_i^n - 1$. Let b_i be integers with $pb_i = a_i^n - 1$ for $i = 0, 1, 2, \cdots, n$. Thus, we have the matrix equation

$$\begin{pmatrix} 1 & a_0 & a_0^2 & \cdots & a_0^n \\ 1 & a_1 & a_1^2 & \cdots & a_1^n \\ 1 & a_2 & a_2^2 & \cdots & a_2^n \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 1 & a_n & a_n^2 & \cdots & a_n^n \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ \ddots \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} pb_0 \\ pb_1 \\ pb_2 \\ \ddots \\ pb_n \end{pmatrix}.$$

Calling the left-most matrix A and multiplying on the left by adj(A), we have

$$\begin{pmatrix} -det(A) \\ 0 \\ \ddots \\ 0 \\ det(A) \end{pmatrix} = adj(A) \begin{pmatrix} pb_0 \\ pb_1 \\ pb_2 \\ \ddots \\ pb_n \end{pmatrix}.$$

As adj(A) has entries from the integers, the right hand side above is of the form $\begin{pmatrix} pc_0 \\ pc_1 \\ pc_2 \\ \ddots \\ pc_n \end{pmatrix}$. Thus, $pc_1 = \det(A) = \prod_{i>j}(a_i - a_j)$, which is impossible as all the a_i's are distinct and less than p.
So, we have proved for each $n < p$ that there are at the most n among $1, 2, \cdots, p-1$ which satisfy $a^n \equiv 1$ modulo p, Consider some $n < p$ and let us count how many $a's$ have 'order' n (that is, satisfy $a^n \equiv 1$ with n smallest possible). Clearly, if there is such an 'a', then n must divide $p-1$ (because $p-1 = qn+r$ with $r < n$ would give $a^r \equiv 1$ modulo p). For any divisor n of $p-1$, look at such an a if it exists. Then, $a, a^2, \cdots, a^n$ are distinct modulo p and all of them give solutions of the congruence $X^n - 1 \equiv 0$ modulo p. By the earlier observation, these must be *all* the solutions. Among them, the ones which have 'order' n are a^d with d relatively prime to n. Hence we have for each divisor n of $p-1$ either 0 or $\phi(n)$ a's of order n. Therefore, the totality of $1, 2, \cdots, p-1$ is made up of these and we get
$p-1 = \sum_{n|(p-1)} f(n)$ where $f(n) = \phi(n)$ or 0 according as to whether there is some a of order n or not.
Now, one has $m = \sum_{d|m} \phi(d)$ for each natural number m. This implies that $f(n) = \phi(n)$ for all $n|(p-1)$. In particular, $f(p-1) \neq 0$ (!)

As a whole, $\mathbf{Z}_p = \{0, 1, \cdots, p-1\}$ has the structure of a field under the operations of addition mod p and multiplication mod p. One may study things like polynomials with coefficients in it akin to the usual polynomials over real or complex numbers. One may construct other finite fields with the help of such polynomials. In fact, if $f(X)$ is any such polynomial of degree n which is irreducible, then one formally considers finite sums of the form $a_0 + a_1\theta + \cdots + a_{n-1}\theta^{n-1}$ for a 'symbol' θ and a_i's in $\mathbf{Z}_p$. The relevance of $f(X)$ comes as follows. To multiply two such expressions $p(\theta), q(\theta)$, just multiply out $p(X)q(X)$ as polynomials over $\mathbf{Z}_p$ but consider the remainder polynomial $r(X)$ on division by $f(X)$; then $r(\theta)$ is taken to be the product of $p(\theta)$ and $q(\theta)$. Similar definition is given for the sum. Note that we are just reading all polynomials "modulo $f(X)$" and we use the division algorithm which is valid for polynomials over any field. This finite field is usually denoted by $\mathbf{Z}[X]/(f(X))$. Finite fields are central to coding theory and cryptography.

2 Public key cryptography

"*Cryptography is communication in the presence of adversaries*" - this cryptic definition is due to Rivest, one of the pioneers of public key cryptography. Ignoring the practical uses for the moment, we may think of cryptography as a game where the cryptographer (the encoder) is in constant struggle to elude the cryptanalyst (the codebreaker). In the cryptosystems discussed so far, the sender and the receiver must have exchanged some secret key before they communicate. This needs a secure channel. Moreover, this must remain secret at *both ends*. An additional difficulty is that each transaction requires a different key as two or more transactions with the same key can usually lead to breaking of the message. Further, if n people need to exchange secret messages, they need $\frac{n(n-1)}{2}$ secret key exchanges. In all these (so-called private key) cryptosystems discussed, anyone who could decipher messages could, with little or no effort, determine the enciphering key. For a long time, it was thought that deciphering was synonymous with finding the enciphering key (an interesting letter from Casanova refers to this as if it were a foregone conclusion). In public key cryptosystems which is our subject matter here, the major advantage is that to send any number of secure messages to a person A, just two keys are needed throughout - an enciphering key known to the public and a deciphering key known only to A. In public key cryptosystems, we shall see that someone who knows how to encipher cannot use the enciphering key (in practical terms) to determine the deciphering key. In other words, the enciphering function f which changes plaintext to a ciphertext is easy to compute once the enciphering key is given, but it is very hard in practice to compute the inverse function f^{-1} without the deciphering key. Such functions are called trapdoor functions and number theory seems to provide a wealth of such functions. It was a serendipitous discovery of Diffie & Hellman which gave birth to public key cryptosystems. The reason for the name 'public' key is that the enciphering key can be made public. One can imagine a public directory containing the enciphering key E_A of each person A. Anyone wanting to send a secret message to A can use the key E_A to encipher the message and send. Now, the only person who knows the deciphering key is A herself. The sender does not have to be known to the receiver nor is a prior relation of trust necessary to be established between them. Unlike private key cryptosystems where the time taken to encipher or to decipher are of the same magnitude, in public key cryptosystems, deciphering takes

much more time than (something like the exponential of) enciphering. Public key cryptosystems, though secure, are much slower. Normally, one uses them to exchange secure keys for a private cryptosystems which is used to send the actual messages.
Here are a few comments on the state of the art. The vulnerability or security of a popular cryptosystem like RSA, is based on the belief that factoring in general is a hard problem. One should note that it is not known that breaking RSA is equivalent to factoring. Although it was believed to be so until some time ago, some recent work of Boneh & Venkatesan points evidence to the contrary. On the other hand, cryptosystems based on class groups of orders (of discriminant n) in number fields or, on the group of rational points of elliptic curves over Z/nZ, can be proved to be as hard to break as factoring n is. Also, it ought to be stressed that what is considered secure today could be insecure tomorrow. There is no single system that has been proved to be secure, at present.

Idea of signing messages

Each person A has a *public* enciphering transformation f_A using which anyone can send messages to A. Of course, A herself knows the inverse transformation f_A^{-1} which is very hard to invert in practice. Now, suppose Alka wants to send her 'signature' S to Beena. It is not enough if Alka sends $f_{Beena}(S)$ to Beena since anyone can do it and Beena will not know for certain that it is from Alka. So, Alka sends Beena the message $f_{Beena}f_{Alka}^{-1}(S)$. Since Beena expects that Alka has sent the message, she will apply $f_{Alka}f_{Benna}^{-1}$ and recover the message knowing at the same time that for practical purposes only Alka could have known f_{Alka}^{-1}.
In the above, we have tacitly assumed that the set of plaintexts equals the set of ciphertexts. One has to modify it otherwise.

RSA cryptosystem

This is the most popular of public key cryptosystems in use today. It was a system described by Rivest, Shamir and Adleman in 1977. It is based on the following elementary fact from number theory. If $p \neq q$ are primes and $n = pq$, then the number $\phi(n) = (p-1)(q-1)$ satisfies Euler's congruence

$a^{\phi(n)} \equiv 1 \bmod n$ for any $(a, n) = 1$. Let us describe the RSA system now.

I. Each user A selects two large primes $p_A \neq q_A$. Write $n_A = p_A q_A$.
II. Each user A selects a large random d_A such that $(d_A, \phi(n_A)) = 1$.
III. Each user A determines the unique $e_A \leq \phi(n_A)$ such that $e_A d_A \equiv 1 \bmod \phi(n_A)$. Note also that $(e_A, \phi(n_A)) = 1$.
IV. Each user A keeps p_A, q_A, d_A private.
V. The numbers n_A, e_A are made public.
VI. Plaintexts are represented by a sequence of integers between 0 and $n_A - 1$.
VII. Public can use the enciphering transformation

$$f_A : \mathbf{Z}/n_A\mathbf{Z} \to \mathbf{Z}/n_A\mathbf{Z} \ ; \ P \mapsto P^{e_A} \ \ mod \ \ n_A$$

to send messages to A. The inverse of f_A is $C \mapsto C^{d_A} \bmod n_A$ is known only to A.

Why it works :
First, mathematically, A can read the message because of the following reason. If $(P, n_A) = 1$, that is clear from Euler's congruence. If $p_A | P$, then $q_A \not| P$ as $P < p_A q_A$; so

$$P^{e_A d_A} = P^{1+t(p_A-1)(q_A-1)} \equiv P \ \ mod \ \ q_A.$$

Evidently

$$P^{e_A d_A} \equiv 0 \equiv P \ \ mod \ \ p_A.$$

Now, knowing $p_A.q_A$ (which A does), it is easy to compute their product n_A as well as $\phi(n_A) = (p_A - 1)(q_A - 1)$. Also, raising to a power is not considered time-consuming as it can be done by a method of repeated squaring. However, only knowing n_A, it is very difficult, in practical terms, to factorise and obtain p_A and q_A. Knowing $\phi(n_A)$ is also equivalent to knowing p_A and q_A because $\phi(n_A) = n_A - p_A - q_A + 1$ would give us $p_A + q_A$.
It is unknown as yet as to how to break RSA without factoring n_A.

Signature through RSA
To send her signature S to Beena, Alka proceeds as follows. Note that the numbers n_A, n_B for Alka and Beena (although public) are usually different. To deal with this, Alka sends $f_B f_A^{-1}(S)$ if $n_A < n_B$ and sends $f_A^{-1} f_B(S)$ if $n_A \geq n_B$. These are, respectively, $(S^{d_A} \bmod n_A)^{e_B} \bmod n_B$ and $(S^{e_B} \bmod n_B)^{d_A} \bmod n_A$.

Example.
As mentioned earlier, small primes p, q should be avoided in order that factorisation is computaionally infeasible. However, for the sake of demonstration, let us take small primes. Let A have the public key $(n, e) = (6012707, 3674911)$. Actually, she has chosen the primes $p = 2357$ and $q = 2551$ and has computed $n = pq = 6012707$ and $\phi(n) = 6007800$. Her enciphering key, she takes to be $e = 3674911$ and, therefore, her deciphering key is $d = 422191$. To encipher the message $m = 5234673$ to be sent to A, a sender B (computes and) sends $c = m^e \bmod n$; this equals 3650502. On receiving this, A deciphers m by computing $c^d \bmod n$.

Which choices are to be avoided ? :
(i) *Small p, q should be avoided.*
Typically, one works with prime numbers of the order of 1000 bits in order that the factorisation problem is computaionally infeasible (that is expected to take millions of years even with the aid of existing recent powerful attacking factorisation methods like the quadratic sieve).
(ii) *$|p - q|$ should not be too small.*
Otherwise, $p \sim \sqrt{n} \sim q$ and an eavesdropper can simply check for factors close to $\sqrt{n}$.
(iii) *Small enciphering key e should be avoided.*
This is for two reasons. The first basic one is that if $m^e < n$, then the public knowledge $m^e \bmod n$ is already the whole of m^e and one can recover m simply by taking integer e-th roots ! The second reason (which applies especially when the same message is being sent to several people) is that Chinese remainder theorem can be easily solved knowing a few of the congruences. For example, suppose $e = 3$ and the enciphered texts $m_i = m^3 \bmod n_i$ are sent to three persons A_i for $i = 1, 2, 3$. If $m^3 < n_1 n_2 n_3$, then a common solution for the 3 congruences $x \equiv m_i \bmod n_i$ is $x = m^3$ itself. An eavesdropper simply comutes the cuberoot of x to know the message m.
(iv) *Small d should be avoided.*
If d has bit size at the most quarter of that of n, there is an efficient algorithm to compute it from (n, e). One can avoid it by having d approximately of the same bit size as n.
(v) *Common n should be avoided.*
Suppose Alka and Beena have a common n. Let $(e_A, e_B) = 1$. Imagine the scenario when Chandra sends a message P to both Alka and Beena. Now Chandra sends $P_A \equiv P^{e_A} \bmod n$ to Alka and sends $P_B \equiv P^{e_B} \bmod n$ to

Beena. If an eavesdropper Damini intercepts these messages, she can decipher the messages and recover P even without knowing d_A, d_B as follows. She computes $f_A = e_A^{-1} \bmod n$ and the number $f = (e_A f_A - 1)e_B^{-1}$. Then,

$$P_A^{f_A} P_B^{-f} \equiv P^{e_A f_A - e_B f} = P \mod n.$$

Thus, the RSA protocol fails for a common modulus even if the deciphering keys are kept secret.
Also, there is another way of seeing protocol failure for a commond modulus. It is clear that in a community with a common n, if someone's deciphering key d is somehow divulged, then n is factored and then all the deciphering keys d's are known ! To see how finding d leads to factoring n, we proceed as follows.
Now $a^{ed-1} \equiv 1 \bmod n$ for all $(a, n) = 1$. Write $ed - 1 = 2^s t$ with t odd. It can be shown that for at least half of the elements a in $(Z/nZ)^*$, there is a common i with

$$a^{2^i t} \equiv 1 \mod n,$$
$$a^{2^{i-1} t} \not\equiv \pm 1 \mod n.$$

The GCD of $a^{2^{i-1}t} - 1$ and n is then a factor of n. The eavesdropper simply looks at various random a and $i \leq s$.

Further remarks on RSA
Compared to the conventional private key cryptosystems, RSA takes much longer. Ironically, though public key cryptosystems like RSA were invented to obviate the necessity of exchanging many keys, it is serving exactly that purpose viz., it is used to exchange keys necessary for some private key cryptosystem ! I have heard from a cryptologist who worked in a government security agency that there is one more reason for not using RSA where government security is involved. Common sense tells us that one day if a successful attack is launched which breaks RSA, then there will be nothing else to fall back on! Therefore, it is mainly in financial transactions or signature authentication or private key exchanges where RSA is principally used nowadays.
The RSA involves the permutation polynomial x^e in F_q, and there have been later cryptosystems based on other permutation polynomials like the Dickson polynomials etc.
There are two main points which should be kept in mind :

(i) *It is* **not proved** *that finding fairly large primes p, q is easier than factoring pq.*
(ii) *It is* **not known** *that breaking RSA is equivalent to factoring n. Although it was believed to be so until some time ago, some recent work of Boneh & Venkatesan points evidence to the contrary.*
Later, we will discuss aspects of the problems of primality testing and factoring so that we can compare the orders of difficulty in performing RSA and in an eavesdropper breaking it.

Rabin cryptosystem

Recall the point (i) made above in relation to RSA. In contrast, here is a public key cryptosystem invented in 1979, deciphering which can be *proved* to be equivalent to the problem of factoring. This is as follows. To send a message P to Beena, Alka considers Beena's public key $n_B = p_B q_B$. She represents the message P as a positive integer $< n_B$ and coprime to n_B, and sends Beena the message $S = P^2 \bmod n_B$. To decipher P, Beena essentially uses the Euclidean algorithm. Knowing her p_B, q_B, it is possible to determine four possible candidates for the square-root of S mod n_B. Let us write this when $p_B \equiv q_B \equiv 3 \bmod 4$ which is a slightly simpler case. Beena computes numbers u, v such that $up + vq = 1$ (we have written p, q for p_B, q_B here). Further, she computes the numbers $r = S^{(p+1)/4} \bmod q$ and $s = S^{(q+1)/4} \bmod p$. Then, P is one of the four numbers $\pm(ups \pm vqr) \bmod n_B$. One problem is that Beena has to decide which of the four is the correct message. Note that the problem that an eavesdropper has to solve is that of finding the square-root of S mod n_B. This is computaionally equivalent to factoring n_B.

Diffie-Helman cryptosystem

Although we noted right in the beginning that public key cryptosystem was a discovery emerging from the work of Diffie & Hellman, we discussed RSA first as that is the most popularly used one. Now, we discuss the so-called discrete log problem which is related to the Diffie-Hellman system.

Discrete log problem

In the finite cyclic group Z/nZ of integers mod n under addition mod n,

suppose we are given a generator a; that is, an element a coprime to n. Given any element b, it can be expressed as ax, and it is quite easy to find x. Indeed, $x = a^{-1}b$ can be found by finding a^{-1} mod n using the Euclidean algorithm. Thus, the time to find a solution to $ax = b$ in Z/nZ is of a small order. In contrast, consider the multiplicative group F_q^* of a finite field; it is a cyclic group. If g is a generator, then any nonzero element of the field is of the form g^x. If the field and a generator g of its multiplicative group are made public, and an element $h = g^x$ is also made public, it turns out to be extremely hard to compute x. This x is known as the discrete logarithm of h in F_q. In comparison, in real numbers, it very easy to find the logarithm of a number. The discrete log problem for F_q^* is the problem of computing x given g and g^x (and q, of course). For the group-theorist who thinks any two finite cyclic groups of the same order are absolutely alike, here is a shocker! The discrete log problem is trivially solvable in Z/nZ but, so far, it has proved computationally infeasible to solve the discrete log problem for F_q^*. The U.S.Government's digital signature algorithm is based on the discrete log problem for subgroups of F_p^*. In some cases (for instance, when p is a prime such that the prime factors of $p-1$ are all small), the discrete log problem (DLP) can be successfully attacked in F_p.

Diffie-Helman key exchange problem

This was discovered in 1976. As before, the finite field F_q is public knowledge. So is a generator g of F_q^*. Persons A and B who wish to exchange a secret message (a key perhaps) choose two secret numbers a, b respectively, each less than $q-1$. They send the messages (this is public knowledge) g^a and g^b respectively. On seeing g^a, B computes $(g^a)^b = g^{ab}$. Similarly, A also knows $g^{ab} = (g^b)^a$. *It is considered extremely hard to compute g^{ab} knowing only g, g^a, g^b.* This way, A and B can share g^{ab} as their common enciphering private key to communicate via a conventional cryptosystem. Finding g^{ab}, given g^a and g^b is also known as the Diffie-Hellman problem (DHP).

DLP versus DHP :
(i) *If the discrete log problem is solved, then the Diffie-Hellman cryptosystem will be broken.*
(ii) *As of today, it is unknown whether there is a way of finding g^{ab} directly from g^a and g^b without knowing a and b i.e., without solving the discrete log problem.*
This is an open problem. However, ideas of Boneh, Lipton, Maurer & Wolf

show that for certain special classes of groups, the DLP and the DHP are polynomially equivalent. For this, one needs the choice of an elliptic curve E over F_p such that $E(F_p)$ is cyclic, and its order l is *smooth* in the sense that all prime factors of $l-1$ are at most as big as a polynomial function of $\log p$. The problem is that it is not yet clear whether such curves exist.

Massey-Omura cryptosystem

Imagine that I want to send you a diamond by post. The only condition is that whatever we send each other by post is to be sent only in a locked box and that anybody who intercepts it without the correct key cannot open it. Assuming that we do not have any prior communication, how are we to accomplish this? Before answering this, we inroduce a cryptosystem - this looks like a digression from the task but it is really not.
To send a secret message P, I select a cyclic group G which you (and everyone in the world also) know. Then, I select a random $e_I < O(G)$ which is coprime to $O(G)$. I compute $d_I = e_I^{-1}$ in G. Then, I send you the message P^{e_I}. This is meaningless to you (or any eavesdropper for that matter) when you receive it. But, you don't bother but simply choose a random e_Y of your own which is coprime and less than $O(G)$. You send me back the message $(P^{e_I})^{e_Y} = P^{e_I e_Y}$. Since I know d_I, I recover P^{e_Y} which I send you back as it is. No eavesdropper can make sense of this still. When you receive it, you can recover P as you know e_Y!
In other words, to send you the diamond, I put it in a locked box and send it to you. You receive it and put your own lock and send it back to me. Then, I unlock my lock and send it back to you so that you can unlock the box !

El Gamal cryptosystem

This was invented in 1985. This is a system where the receiver not only receives the secret message but also an authentication of the sender's signature. It works as follows. A finite cyclic group G and a generator g are made public. Typically, a finite field F_q and a generator g of $G = F_q^*$ are made

public. Alka and Beena select random integers $a, b < q$ and make public g^a and g^b respectively. In order to send a message $P \in G$ to Beena, Alka sends the message (g^a, Pg^{ab}). She knows g^{ab} since she knows g^b and a. On receiving this, Beena computes $(g^a)^b$ and then P. Such a message P can serve as a key exchange. In order to authenticate her signature in addition, Alka proceeds as follows. She selects a random $a < q$ and computes g^a as before. In addition, she chooses a secret $s < q$ which is coprime to $q - 1$ and computes g^s. She solves for t to satisfy $g^P = (g^a)^{g^s}(g^s)^t$. Indeed, $t := s^{-1}(P - g^s a) \bmod q - 1$. Note that only Alka could have solved for t since she alone knows a. She sends Beena the tuple (g^a, Pg^{ab}, g^s, t). Beena first computes P as before which enables her to compute g^P and verify that it equals $(g^a)^{g^s}(g^s)^t$. Once she verifies that, she is sure that P has been sent by Alka.

Knapsack problem

This is important for historical reasons - it was the first concrete realization of a public key scheme. The original Merkle-Hellman cryptosystem is now known to be insecure and, in fact, a polynomial time algorithm is known to break it. One version of the knapsack problem known as the Chor-Rivest knapsack scheme has at present resisted breaking. What is the knapsack problem? Given positive integers $a_1, a_2, \cdots, a_n$, and given a positive integer N, can N be written as a sum of some of the a_i's? If yes, how does one choose an n-bit integer $(t_1, \cdots, t_n)_2$ such that $\sum_{i=1}^n t_i a_i = N$? Indeed, interestingly, both these problems are computaionally equivalent. Imagining the a_i's to be the volumes of items to be packed in a knapsack of volume N, the problem is to achieve perfect efficiency. As a general problem, it is extremely hard (NP-complete). However, the Merkle-Hellman cryptosystem involves the knapsack problem for *superincreasing* sequences a_i (that is, $\sum_{i=1}^j a_i < a_{j+1}$ for all $j < n$). For such sequences, the problem is easily solved in polynomial time by the "greedy algorithm".

The Merkle-Hellman cryptosystem

(i) There is a common n chosen for the whole public.
(ii) Each person privately chooses a super-increasing sequence $(b_1, \cdots, b_n)$ and a modulus $M > \sum_{i=1}^n b_i$.
(iii) Each person privately chooses an integer w coprime to and less than M.
(iv) Each person privately chooses a disguising permutation σ and computes

$a_i = wb_{\sigma(i)}$.
(v) Each person's public key is the sequence $a_1, \cdots, a_n$.

With this background, the Merkle-Hellman cryptosystem can be described as follows. Let us say Beena wants to send Alka a message P which is represented as an n-bit $m_1 m_2 \cdots m_n$.
Beena sends Alka the message $m = m_1 a_1 + \cdots + m_n a_n$, where a_i's are Alka's public sequence described above.
Alka recovers the message bits m_i's as follows. She solves the knapsack problem for her super-increasing sequence $b_1, \cdots, b_n$ to compute

$$w^{-1}m = u_1 b_1 + \cdots + u_n b_n$$

where the left hand side is to be understood as the residue mod M. Since

$$w^{-1}m = w^{-1} \sum m_i a_i = \sum m_i b_{\sigma(i)},$$

we have $m_i = u_{\sigma(i)}$.

Why Merkle-Hellman cryptosystem is insecure

One knows now a polynomial-time algorithm to break it (this is not the most powerful method though). This finds integers w_0, M_0 such that w_0/M_0 is close to w^{-1}/M and such that $c_i = v_0 a_i \bmod M$ form a super-increasing sequence. If an eavesdropper uses this sequence in place of the sequence b_i, then she can decipher the message.

Zero knowledge protocol

Before proceeding, we just recall the connotation of the word 'protocol' here. Here, it means a specific distributed algorithm defined by a sequence of steps specifying the actions of two or more persons to achieve a specific security objective. There are some people who claim to 'know' and predict your future etc. They give a few feelers to let you know that they are capable. Suppose I want to communicate to you that I have found a solution to some problem and I want to do it in such a way that I do not tell you the solution but still convince you that I have solved the problem. This is known as a zero-knowledge proof.

Example : Fiat - Shamir protocol
Suppose you have chosen a secret message P less than and coprime to your public key $n = pq$. Suppose you make public knowledge the message $S = P^2 \mod n$. I claim that I have found your P. To verify this, you can proceed as follows. You ask me to choose a random r and send you the value of $r^2 \mod n$. Once I do that, you toss a coin and you ask me to send you the value of r or of rP according as whether you get haids or tails, say. Note that I can answer both questions correctly only if I really know P. Thus, after d trials, you will be able to verify with probability $1 - \frac{1}{2^d}$ that I know what P is.

Example : Discrete log protocol
As another example, suppose G is a group of order N and $g \in G$ is fixed. Let us assume that g^x is public knowledge but only you know x and I claim that I have found it. I demonstrate my claim in the following manner :

(i) I generate a random $r < N$ and send you g^r.
(ii) You toss a coin. According to whether it is heads or tails, you ask me to reveal r or $x + r$. Once again, I can answer both questions only if I knew x. Therefore, after several trials, you will be able to know with high probability if I commit a fraud.

In conclusion

In 1994, Peter Shor brought the idea of a *quantum computer* on which one could perform computations like factorisation and finding discrete logs using polynomial-time algorithms. Until date, one does not know whether a quantum computer can indeed be built. In another development in 1996, Adleman showed that it was feasible to use techniques from molecular biology to solve some NP-complete problems. The problem instance was encoded in molecules of DNA, and the steps of the computation were performed with standard protocols and enzymes. The fastest supercomputers today can perform 10^{12} operations per second while it is thought plausible that a DNA computer performs more than 10^{20} operations per second. Until date, nobody knows whether it is possible to build a DNA computer. However, it cannot be denied that if either of these computers becomes a reality one day, then that day the existing public key cryptosystems are likely to be rendered insecure.

3 Primality testing and factorisation

All the public key cruptosystems require keys which involve producing large prime numbers. How does one do that? The general approach of a primality test is the following : find a random odd number, use the test to check whether it is composite and if it is start over again. In other words, if a primality test returns a value as composite, it is certainly composite but in most of the tests, if the value is returned as a prime, it is only a probable candidate for a prime. Although popularly called 'primality tests', these 'probabilistic primality tests' should really be called 'compositeness tests'. Of course, there are obvious (deterministic) tests which can tell us for certain if a number is prime but they may not be time-efficient. Only very recently this fundamental problem of the existence and construction of a polynomial-time deterministic algorithm for primality testing has been proved by three Indians - Agrawal, Kayal and Saxena. However, it should be borne in mind that most of the polynomial-time probabilistic algorithms are more powerful and are considered safe enough to be in usage today. Moreover, there are some deterministic algorithms which are slightly worse than polynomial time but are still powerful enough to determine primality of a 100-digit number in a few seconds on a powerful computer. Such an algorithm was found by Adleman-Pomerance-Rumely. Interestingly, the Miller-Rabin test (to be described below) turns out to be a deterministic one if we assume the so-called generalised Riemann hypothesis. In 2002, 3 Indians - Agrawal, Kayal & Saxena astonished the world by coming up with a very simple algorithm to test primality which is polynomial time, deterministic and does not depend on any unproved hypotheses. It must be borne in mind though, that the probabilistic algorithms are already powerful enough for all practical purposes.

Carmichael numbers and pseudoprimes

Fermat's little theorem tells us that $a^{n-1} \equiv 1 \bmod n$ when n is prime and $(a, n) = 1$. However, this may happen for a composite n also for some a coprime to it. For instance, $341 = 31 \times 11$ satisfies $2^{340} \equiv 1 \bmod 341$. This statement is rephrased as saying that 341 *is a pseudoprime to the base* 2. Moreover, 2 is called a *Fermat liar* for 341. Note that $3^{340} \equiv 56 \not\equiv 1$

mod 341; so, 341 is not pseudoprime to the base 3 and, therefore, indeed composite. When we conclude compositeness using this analysis, note that this does not give us factors. Now, can it happen that an odd composite number is a pseudo-prime to the base a for *every* a coprime to n ? Indeed, it can, and such n are called Carmichael numbers. The first example is $561 = 3 \times 11 \times 17$. There is a very neat (and easy to prove) characterisation of Carmichael numbers :

A composite number n is a Carmichael number if, and only if, it is square-free and each prime divisor p of n satisfies the property $(p-1)|(n-1)$.
Proof.
If $n = p_1 \cdots p_r$ with $n-1$ divisible by $p_i - 1$ for $i = 1, \cdots, r$, then for each $(a, n) = 1$, $a^{n-1} = (a^{p_i - 1})^? \equiv 1 \bmod p_i$ for all i, which shows that $a^{n-1} \equiv 1 \bmod n$.
Conversely, suppose n is a Carmichael number. Let $p|n$ be a prime. Let a be a primitive root mod p. Since $a^{n-1} \equiv 1 \bmod n$ by hypothesis, we have $a^{n-1} \equiv 1 \bmod p$. Therefore, $p-1$ (being the order of a mod p) divides $n-1$. Moreover, if $p^2|n$, then for a primitive root a mod p^2, the property $a^{n-1} \equiv 1 \bmod p^2$, shows similarly that $p(p-1) = O(a)$ divides $n-1$. But then p divides $n-1$, an impossibility.

It was proved only a few years back that there are infinitely many Carmichael numbers - in fact, there are at least $N^{2/7}$ Carmichael numbers less than N if $N >> 0$. Therefore, a primality test just based on Fermat's little theorem would be wrong infinitely many times; that is, each Carmichael number would be returned as a prime number by such a test. The Miller-Rabin test uses a modification of pseudoprimes called strong pseudooprimes.

Miller-Rabin test

This is in wide usage especially for RSA. Let n be an odd prime; write $n - 1 = 2^s r$ with r odd. For $(a, n) = 1$, we have $a^{2^{s-1}r} \equiv \pm 1 \bmod n$. Thus, either a satisfies at least one of the conditions :
$a^r \equiv 1 \bmod n$ or $a^{2^i r} \equiv -1$ for some $0 \leq i < s$.
A composite n which satisfies this last-mentioned property is called a *strong pseudoprime to the base a*. One also calls such a base *a strong liar* for n. When n is *not* a strong pseudoprime to some base a (that is, if each of the

$s+1$ congruences fails), then evidently n is composite, and a is known as a *strong witness* to the compositeness of n.
For example, the Carmichael number 561 has 2 as a strong witness. This is so because $560 = 16 \times 35$ and $2^{35} \equiv 263, 2^{2\times 35} \equiv 166$, and $2^{4\times 35} \equiv 67 \bmod 561$. Also $2^{8\times 35} \equiv 1 \bmod 561$.
Miller-Rabin test starts by picking a random $a < n-1$ and checking whether $a^r \bmod n$ is ± 1. If it is, then one concludes that n (passes the test and) is a (probable) prime; move to the next a. If it is not ± 1, keep squaring (upto $s-1$ times) and checking until we reach -1. If it does, then again n passes the test and is a probable prime; move to the next a. If -1 is never reached, then n must be composite.
We shall see now that at the most 1/4-th of the numbers $a < n$ can be strong liars for a composite n. Thus, after d iterations, the probability that the Miller-Rabin test concludes primality of a composite n is at the most $(1/4)^d$.

If n is composite, then the set $\{1, 2, \cdots, n-1\}$ contains at the most $(n-1)/4$ strong liars.
Proof.
Among strong liars for n, there are those a which satisfy one of the s congruences $a^{2^i r} \equiv -1$ for some $0 \leq i < s$. Indeed, if $a < n$ is a strong liar for n which satisfies $a^r \equiv 1 \bmod n$, the strong liar $b = n - a$ satisfies $b^r \equiv -1 \bmod n$. Let d be the maximum value of i for which $a^{2^i r} \equiv -1 \bmod n$ for some $(a, n) = 1$. Then, we have the inclusions $A \leq B \leq C \leq D$ of the following four subgroups of $(Z/nZ)^*$:

$$A = \{a : a^{2^d r} = 1\},$$

$$B = \{a : a^{2^d r} = \pm 1\},$$

$$C = \{a : a^{2^d r} \equiv \pm 1 \ mod\ p^{n_p}\ \forall\ p|n\}, and$$

$$D = \{a : a^{n-1} = 1\}.$$

An easy counting of indices can be done which tells us that the index of B in $(Z/nZ)^*$ is at least 4.

Miller-Rabin test is deterministic if we assume GRH (the generalised Riemann hypothesis.
Proof.
We shall just give a sketch. A consequence of the GRH is that for any prime

$p \geq 3$, the least quadratic nonresidue is $< 2(log\ p)^2$. We show that if the Miller-Rabin test is performed for all $a < 2(log\ n)^2$ then it finds a strong witness for n.
Let us consider only the case when n has two prime divisors p, q with $p-1$ and $q-1$ having different 2-adic valuations; say $v_2(p-1) > v_2(q-1)$. Take a to be a quadratic nonresidue mod p with $a < 2(log\ p)^2$. Recall our notation $n-1 = 2^s r$. Now $a^{(p-1)/2} \equiv -1 \bmod p$ since a is a quadratic nonresidue mod p. Evidently, then $v_2(p-1)$ is also the power of 2 dividing the order of a in $(Z/pZ)^*$. In particular, the order of a in $(Z/pZ)^*$ is even.
Suppose, if possible, that the test produces a as a strong liar. Then, either $a^r \equiv 1$ or $a^{2^i r} \equiv -1 \bmod n$ for some $0 \leq i < s$. The former case is an impossibility since it would imply that a has odd order in $(Z/pZ)^*$. In the latter case, suppose the congruence $a^{2^i r} \equiv -1 \bmod n$ holds. This implies that the order of a in $(Z/pZ)^*$ as well as in $(Z/qZ)^*$ have v_2 equal to $i+1$. But, the power of 2 dividing the order of a mod q is at the most $v_2(q-1)$ (this is true for any prime). Therefore, we have

$$v_2(Ord_p(a)) = v_2(p-1) > v_2(q-1) \geq v_2(Ord_q(a))$$

which is a contradiction. The other case when $v_2(p-1) = v_2(q-1)$ can be analysed in a similar fashion once one chooses a with the quadratic residue symbols of a mod p and mod q being different.

Deterministic algorithms for special sequences

The AKS algorithm is recent; the above one is deterministic only under GRH. However, there are some algorithms which are deterministic and work efficiently for special sequences. One such is the Lucas-Lehmer deterministic polynomial-time test for the sequence of Mersenne numbers $2^s - 1$. It is based on the fact that $2^s - 1$ is prime if, and only if, s is prime and the sequence $u_0 = 4, u_{k+1} = u_k^2 - 2 \bmod n$ satisfies $u_{s-2} = 0$. Note that the first condition "s is prime" can be checked by checking for factors until $\sqrt{s}$ and this is polynomial-time (i.e., polynomial in log n).

Pocklington's primality test

This test determines primality of n when $n-1$ has a big prime factor. An analogue of this using elliptic curves has also been formulated. Suppose $n-1$

has a prime divisor $q > \sqrt{n}$. Then, one can determine whether n is prime by using the following fact :
If there exists a such that $a^{n-1} \equiv 1$ mod n and $(a^{(n-1)/q} - 1, n) = 1$, then n must be prime.
Proof.
If not, then let $p \leq \sqrt{n}$ be a prime factor. Then, $q > p - 1$ implies that $qr \equiv 1 \bmod p - 1$ for some r. But then $a^{(n-1)/q} \equiv a^{r(n-1)} \equiv 1 \bmod p$, which gives $(a^{(n-1)/q} - 1, n) \geq p$, a contradiction.

This fact can be used in the following manner to test the primality of n :
Write $n - 1 = ab$ where $(a, b) = 1$, $a > b$ and suppose the factorisation of a is known. If, for each prime $p|a$, we can find a_p with $a_p^{n-1} \equiv 1$ mod n, and $(a_p^{(n-1)/p} - 1, n) = 1$, then and only then n is prime.

Example.
Let $n = 105554676553297$ whose primality we wish to test. Now $n - 1 = 2^4.3.1048583.2097169 = ab$ where $b = 2^4$ say. If we take $a_3 = a_{1048583} = a_{2097169} = 2$, we conclude that n is prime, if we assume that $p = 1048583$ and $q = 2097169$ are primes. Thus, we have reduced primality testing to smaller numbers; that is, we have done what is known as a *down run*. Writing $p-1 = 2.29.101.179 = ab$ where we have set $a = 29.101$, the choices $a_2 = a_{101} = 2$ prove primality of p (for primes up to 1000, primality can be checked by hand). Similarly, as $q - 1 = 2^4.3.43691$, it can be seen that $a_3 = 5, a_{43691} = 2$ proves that q is prime if we know that $r = 43691$ is prime. Since $r - 1 = 2.5.17.257 = ab$, taking $a = 257$ and $a_{257} = 3$, we conclude that 43691 is a prime.

Agrawal-Kayal-Saxena algorithm

We mention very briefly their algorithm. As we have seen, most algorithms start with Fermat's little theorem. For instance, we have the algorithm which is a polynomial rephrasing of the fact about binomial coefficients noted in the beginning :
n is prime if and only if $(X - 1)^n \equiv X^n - 1$ modulo n.
This is infeasible on the first glance because of having to compute the polynomial $(X-1)^n$ which takes $n-1$ multiplications. However, this is not serious because of an old Indian method of Pingala from 200 BC; one does repeated squaring. That is, if $n = 2^{i_1} + \cdots + 2^{i_k}$, then $(X-1)^n = (X-1)^{2^{i_1}} \cdots (X-1)^{2^{i_k}}$ which is obtained by repeated squaring of $X - 1$. However, a more serious

problem with the above is to compute n coefficients in order to check the validity of the congruence $(X-a)^n \equiv X^n - a \bmod n$. The basic idea of the A-K-S algorithm is to make it feasible by evaluating both sides modulo a polynomial of the form $X^r - 1$. Their algorithm would take $O(r^2 log^3 n)$ time to verify $(X-a)^n \equiv X^n - a \bmod (X^r - 1, n)$. Note that it may be that $(X-a)^n \not\equiv X^n - a \bmod n$ but still $(X-a)^n \equiv X^n - a \bmod (X^r - 1, n)$. Also, as there are composites also which satisfy this congruence, one has to choose r and a suitably. One general comment to note is that it is far easier to test a polynomial for irreducibility mod p than to test primality of a natural number. In a nutshell, here is the A-K-S algorithm :

A-K-S algorithm to check primality of n
Step I
Check if n is a perfect power a^b with $a, b \geq 2$. Declare it composite if yes; if not go to the next step.
Step II
Choose the smallest prime r not dividing n such that the order of n modulo r has a prime divisor $q > [2\sqrt{r} log n] + 2$.
Step III
For $2 \leq a \leq l := [2\sqrt{r} log n] + 1$, check if a divides n. If yes, then n is composite. If not go to the next step.
Step IV
With r, l as above, check for each $a \leq l$, if
$(X-a)^n \equiv X^n - a$ modulo $(X^r - 1, n)$. If the congruence is not satisfied for some a, declare n is composite. If it is satisfied for all a, declare n prime.

The beauty of this is that the correctness of the algorithm (that is, the fact that a composite number is not declared to be a prime by the algorithm) can be checked quite easily using only elementary number theory. Let us indicate how.
Assume that a composite number n passes through the test is and declared a prime. Then, $(X-a)^n \equiv X^n - a$ modulo $(X^r - 1, n)$ for $a = 1, 2, \cdots, l$. We may replace n in modulo $(X^r - 1, n)$ by any divisor of n. In particular, let p be a prime divisor of n; then $(X-a)^n \equiv X^n - a$ modulo $(X^r - 1, p)$ for $a = 1, 2, \cdots, l$. But, we have also $(X-a)^p \equiv X^p - a$ modulo $(X^r - 1, p)$ for $a = 1, 2, \cdots, l$. We can easily see now that :
$(X-a)^{p^i n^j} \equiv X^{p^i n^j} - a$ modulo $(X^r - 1, p)$ for $a = 1, 2, \cdots, l$ and for all $i, j \geq 0$.

Look at the set $L = \{p^i n^j : 0 \leq i, j \leq [\sqrt{r}]\}$. As it has $(\sqrt{r}+1)^2 > r$ elements, at least two of them must be equal modulo r. Say, $m_1 = p^{i_1} n^{j_1} = m_2 = p^{i_2} n^{j_2} = m_1 + kr$ with $(i_1, j_1) \neq (i_2, j_2)$. We get :
$(X-a)^{m_2} \equiv (X-a)^{m_1}$ modulo $(X^r - 1, p)$.
We claim that $m_1 = m_2$. If we show this, then we would have n to be a power of p. But then $n = p^t$ with $t > 1$ as it is composite and then the first step would have declared it composite. Therefore, the correctness is proved modulo showing that $m_1 = m_2$.
The proof of this uses finite fields and is simple but we give only some details here. Basically, one looks at an irreducible polynomial $f(X)$ dividing $X^{r-1} + \cdots + X + 1$ mod p. We have the condition :
$(X-a)^{m_2} \equiv (X-a)^{m_1}$ modulo $(f(X), p)$.
Therefore, each element of the form $X - a$ in the field $\mathbf{Z}_p[X]/(f(X))$ satisfies the polynomial equation $g(X) := X^{m_1} - X^{m_2} = 0$. Thus, all elements of the set

$$S = \{\prod_{a=1}^{l} (X-a)^{e_a} : e_a = 0 \ \ or \ \ 1\}$$

satisfy the polynomial $g(X)$. The basic task is to show that S has 2^l elements; that is, all the products there are distinct mod $f(X)$. Once this is shown, the observation that $m_1, m_2 \leq n^{2\sqrt{r}}$ would lead to a contradiction unless $g(X)$ is the zero polynomial. This is because $n^{2\sqrt{r}} < 2^l$ by the definition of l and the polynomial $g(X)$ which has degree $< 2^l$ would have 2^l roots (all elements of S) in the field $\mathbf{Z}_p[X]/(f(X))$.

The A-K-S algorithm has been modified by H.Lenstra to make it more efficient.

Factorisation

For a composite number n, the primality tests we discussed (except the silly one where one checks for factors until $\sqrt{n}$) can help only in deciding whether it is composite and not in finding any factors.
As we pointed out while discussing RSA, there are some cases where factorisation can be done efficiently (these are cases to be avoided in RSA). For example, if $n = pq$ with the primes p, q close to each other. This means that $n = ((p+q)/2)^2 - ((p-q)/2)^2$ is close to a square. Thus, one starts with

$t > \sqrt{n}$ and $t \sim \sqrt{n}$ and computes $t^2 - n$ to check whether it is a square. One method which works more generally is the following one.

Pollard's rho method

This is a method adapted to finding small factors of a composite number which is not a prime power. It depends on random maps between finite sets. If we are attempting to find a factor of n, let $\mathcal{F}_n$ denote the set of all functions from $\{1, 2, \cdots, n\}$. Thus, note that the probability of choosing a particular function is $1/n^n$. For such a function f, the algorithm attempts to find *duplicates/collisions* among an orbit of a point x_0 under iterates of f. Of course, since the set is finite, any such sequence is eventually cyclic. The hope is that when there is a collision, say $x_i = x_j$ (where these are the i-th and j-th iterates of x_0), the GCD of n and the difference $x_i - x_j$ is > 1. The choice of x_0, f etc. can be varied in the event of failure. To describe the algorithm, let us first describe a directed graph that one can associate to f. The graph is defined simply by drawing an edge from each point x to $f(x)$. Some points have no predecessors but each point leads by a unique path to a unique cycle. Starting with a point x_0, the number of edges one has to travel to hit a cycle is called the *tail length* of x_0 and the number of edges in that terminating cycle is known as the *cycle length* of x_0. The total of these lengths (the rho length) is denoted by $\rho(x_0)$ - this is the origin of the name 'rho method'. Here are a few facts which are known about the expectations of the various parameters :

Facts on expectations.
As $n \to \infty$, for a random point x_0 and a random function f, the expected :
(i) number of points with no predecessors is n/e,
(ii) tail length is $\sqrt{n\pi/8}$,
(iii) cycle length is $\sqrt{n\pi/8}$, and
(iv) rho length is $\sqrt{n\pi/2}$.

Thus, note that the expected number of inputs starting with any point x_0 before we hit an x_i which is a duplicate, is $\sqrt{n\pi/2}$ as $n \to \infty$.

In Pollard's rho method, one usually chooses the function $f(x) = x^2 + 1 \mod n$. Starting with $x_0 = y_0 = 2$, one computes $x_1 = f(x_0), y_1 = f(f(x_0))$ and also $d = (n, x_1 - y_1)$. If $1 < d < n$, then the algorithm terminates with the assertion that we have successfully found a factor $n > d > 1$ of

n. If $d = 1$, compute x_2, y_2 etc. in the obvious manner. Check whether $(n, x_2 - y_2)$ is a proper divisor. Proceeding in this manner, if $d = n$, at some point, then the algorithm terminates as a failure. Assuming that the function $f(x) = x^2 + 1 \bmod n$ behaves randomly, the expected time to find a nontrivial factor is $O(n^{1/4})$ which is much larger than polynomial-time in log n. Thus, systems like RSA which depend on the difficulty of factorisation are not yet vulnerable.

Here is an example to factorise $n = 455459 = 743 \times 613$ (we write x, y, d for the x_i, y_i and the GCD $(x_i - y_i, n)$) :

x	y	d
5	26	1
26	2871	1
677	179685	1
2871	155260	1
44380	416250	1
179685	43670	1
121634	164403	1
155260	247944	1
44567	68343	743

Pollard's $p-1$ method

Let n be a number which we wish to find a factor of and suppose p is a prime divisor of n such that all prime divisors of $p-1$ are 'small.' Then, one can find a factor of n with a high probability as follows. Fix a bound B and consider $k = B!$. One may also choose k differently but the main point is to consider it to be a multiple of most of the integers upto B. The idea is that if $p|n$ is a prime such that all prime factors of $p-1$ are $\leq B$, then $(p-1)|k$. Thus, one would have $a^k \equiv 1 \bmod p$, so that $(a^k - 1, n)$ would be a multiple of p. Thus, to perform the algorithm, we start with a random $1 < a < n-1$ and compute $a^k \bmod n$ and $d = (a^k - 1, n)$. If $d = 1$ or n, then the algorithm fails. This is unlikely to happen if n has at least two large prime factors. If the algorithm fails, repeat with another a (and another k if necessary).
In cryptological parlance, one says that $p-1$ *is B-smooth* if all prime factors of $p-1$ are $\leq B$. Thus, this method works when $p-1$ is B-smooth for a suitable B.

Here is a way to write the algorithm :
(i) Start with some bound B.
(ii) Choose a random a with $1 < a < n$ and compute $d = (a, n)$. If $d > 1$, return the value of d; this is a factor.
(iii) For each prime $p \leq B$, compute $d = (a^{p^l} - 1, n) \bmod n$ where $p^l \leq n < p^{l+1}$.
(iv) If $d = 1$ or $d = n$, the algorithm terminates as a failure. If not, return the value of d as a factor.

Here is an example where $n = 19048567, B = 19$ and we start with the choice $a = 3$ (note that $(3, n) = 1$) :

p	l	a
2	24	2293244
3	15	13555889
5	10	16937223
7	8	15214586
11	6	9685355
13	6	13271154
17	5	11406961
19	5	554506

Note that $(554505, n) = 5281$. This is a prime and so is the factor $n/5281 = 3607$. Also note that $5280 = 2^5.3.5.11$ which means that the prime factors of $p - 1$ are all ≤ 19 whereas $3606 = 2.3.601$ whose prime factor 601 is much larger. The time taken is polynomial in $Blog(n)$.

4 Elliptic curves - a whirlwind tour

In this section, we recall the theory of elliptic curves rather superficially. While this is by no means an introduction to the subject in any reasonable sense, the discussion does give some details for someone who wants to get into elliptic curve cryptography. Actually, we do not een use all of the results in this section in the applications to cryptology since we do not get too deep into elliptic curve cryptography in these lectures.
Formally, an elliptic curve over a field K is a non-singular, projective plane curve of genus 1 having a specified base point $O \in E(K)$. In simpler language, an elliptic curve is the set of solutions in $\mathbb{P}^2(K)$ of an equation of the form $Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$ with $a_i \in K$, with one of three partial derivatives non zero at any given solution. That the first definition implies the second can be proved using the so-called Riemann-Roch theorem. If char $K \neq 2, 3$, then (using a change of variables), an elliptic curve can also be described as the set of solutions set $\mathbb{P}^2(K)$ of an equation of the form $Y^2Z = X^3 + aXZ^2 + bZ^3$, where the cubic $X^3 + aX + b$ has distinct roots. The above form of the equation is known as the Weierstrass form. In the above definition, a canonical point $O \in E(K)$ is the "point at infinity" $(0, 1, 0)$; this is the only point with $Z = 0$.

The name 'elliptic curve' comes from the fact that these equations arise when one tries to measure the of an ellipse. If E is an elliptic curve over the field $\mathbb{C}$ of complex numbers (that is, if the coefficients of the equation are from $\mathbb{C}$), then $E(\mathbb{C})$ can also be thought of as a complex manifold of dimension one. Thus, $E(\mathbb{C})$ is a compact Riemann surface of genus one and hence, a complex torus of dimension one. i.e., $\mathbb{C}/\Lambda$ where Λ is a lattice in $\mathbb{C}$. This follows from the classical theory of elliptic functions. In other words, there is an isomorphism of Riemann surfaces $E(\mathbb{C})$ and $\mathbb{C}/\Lambda$. As we know, the (singly periodic) trigonometric functions parametrize arc length of the unit circle, and, therefore, the addition formulae of trigonometric functions give the addition formulae for the arc length on the unit circle. For elliptic curves over $\mathbb{C}$, there are the (doubly periodic) elliptic functions which parametrize the arc length. There are addition formulae for elliptic functions which lead to analogous formulae for arc length of the ellipse.
There is an invariant which classifies elliptic curves up to isomorphism over an algebraically closed field. This is the j-invariant of E. Let us assume

that characteristic of K is $\neq 2, 3$ for simplicity of notation. If $E(a,b)$: $Y^2Z = X^3 + aXZ^2 + bZ^3$ is an elliptic curve, one defines the j-invariant to be $j(E) = 1728\frac{4a^3}{4a^3+27b^2}$. Note that :

(i) *$E(a,b) : Y^2Z = X^3 + aXZ^2 + bZ^3$ defines an elliptic curve over K (with $O = (0,1,0) \in \mathbb{P}^2(K)$) if, and only if, $a, b \in K$ and $4a^3 + 27b^2 \neq 0$.*
(ii) *Every elliptic curve over K is isomorphic to $E(a,b)$ for some $a, b \in K$.*
(iii) *An elliptic curve over K is isomorphic to the curve $E(a,b)$ if, and only if, there exists $t \in K^*$ such that its Weierstrass form can be written as $Y^2Z = X^3 + t^4aXZ^2 + t^6bZ^3$.*
So, note that it makes sense to define $j(E) = j(E(a,b))$ if E is isomorphic to $E(a,b)$.
(iv) *(when $K = \bar{K}$) $j(E) = j(E')$ if, and only if, $E \cong E'$.*

Group law

The most important aspect of elliptic curves is that the set of points form an abelian group. Over $\mathbb{C}$, this just comes from the addition formula for the Weierstrass p-function. Over a general field K, the first thing to notice is that there is a point (exactly one with $Z = 0$) on it viz., the point $\mathcal{O}$ with projective co-ordinates $(0,1,0)$. This will be our identity element. However, it is much more convenient to work with the affine curve; that is, where each point (X,Y,Z) on the projective curve corresponds to the unique point $(x = X/Z, y = Y/Z)$ *excepting the point $\mathcal{O}$ which has no corresponding point since $Z = 0$ there.* In that case, the point $\mathcal{O}$ is thought of as 'the point at infinity'. Once again, we assume the characteristic of K is $\neq 2, 3$ and that the (affine) elliptic curve E is given by solutions of the Weierstrass equation $y^2 = x^3 + ax + b$ with $a, b \in K$ and $4a^3 + 27b^2 \neq 0$ along with 'a point $\mathcal{O}$ at infinity'. Let us now define the 'sum of two points of E'. We first define $P + \mathcal{O} = P$ for any P. The line joining two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ (the tangent in case $P_1 = P_2$) meets the cubic in a third point unless $x_2 = x_1$ - note that the latter case happens only if $y_2 = \pm y_1$. If $(x_2, y_2) = (x_1, -y_1)$, define the sum $P_1 + P_2$ to be $\mathcal{O}$. In the case when $x_1 \neq x_2$, the third point Q of intersection of the line P_1P_2 with the curve is taken to satisfy

$$P_1 + P_2 + Q = \mathcal{O}.$$

The line P_1P_2 has the equation $y = mx+c$ where $m = \frac{y_2-y_1}{x_2-x_1}$ and $c = \frac{y_1x_2-y_2x_1}{x_2-x_1}$.

Thus, when $x_1 \neq x_2$, one has $P_3 = (x_3, y_3)$ with

$$x_3 = m^2 - x_1 - x_2 \ , \ y_3 = -mx_3 - c.$$

Finally, when $P_1 = P_2 \neq \mathcal{O}$, the tangent line is $y = mx + c$ with $m = \frac{3x_1^2+a}{2y_1}$ and $c = \frac{-x_1^3+ax_1+2b}{2y_1}$. Thus, in this case $P_1 + P_2 = 2P_1 = (x_3, y_3)$ with $x_3 = m^2 - 2x_1 \ , \ y_3 = -mx_3 - c$ and m, c as last stated.
Note the important point that the co-ordinates of P_3 are also in K. Notice also that even if the co-ordinates of P_i are integers for $i = 1, 2$, and $K = Q$, the co-ordinates of P_3 are only rational; they may not be integers. It should be noted that in this geometric definition of the group law, associativity is not obvious.
We point out one convention which is always used; when E_1, E_2 are elliptic curves defined over some field K, one means by a homomorphism from E_1 to E_2, a homomorphism on the $\bar{K}$-points; that is, from $E_1(\bar{K}) \to E_2(\bar{K})$.

Torsion points of Elliptic curves

Let E be an elliptic curve defined over an algebraic number field K. The most basic result on them is the so-called Mordell-Weil theorem. It asserts that the group $E(K)$ is a finitely generated abelian group. The finite group of torsion points as well as the rank of $E(K)$ are both rather mysterious objects - the latter more so. We mention in passing that although one expects curves of arbitrarily large ranks to exist over a nummber field K, this is unknown as yet. As everyone knows, elliptic curves over number fields form the link between a concrete classical number-theoretic problem like FLT and the modern, technically powerful Langlands's program (via modular forms and Galois representations). That elliptic curves over number fields play a role in elementary number-theoretical problems can be seen even without going to FLT etc. An ancient Greek problem was to determine all *congruent* numbers n; that is, natural numbers n which occur as the area of a right triangle with rational sides. Equivalently, for which n is there an arithmetic progression $x^2 - n, x^2, x^2 + n$ of rational squares ? As it turns out (not very hard to see this), *n is a congruent number if, and only if, the rank of $E(Q)$ for the elliptic curve $E : y^2 = x^3 - n^2x$ is positive.*
Let K be an arbitrary field and E be an elliptic curve defined over K. A point P of $E(K)$ such that $nP = O$, for some integer $n \geq 1$, is called a torsion point of E over K. Here, we have denoted by nP the point $P + \cdots + P$ added

$-n$ times (if $n < 0$). If $K = \bar{K}$, and $n \neq 0$, then the *n-torsion subgroup* is defined as

$$E[n] := \{P \in E(K) : nP = O\}.$$

One has:
If $n \neq 0$ is not a multiple of the characteristic of K, then

$$E[n] \cong \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}.$$

Over an algebraically closed field whose characteristic does divide n, it can happen that $E[n]$ is not as above.
Also, over a field K which is not algebraically closed, the group of n-torsion can be different (of course, it must be a subgroup of the above group).

Remarks
(i) If E is an elliptic curve defined over $\mathbb{R}$, then one may consider $E(\mathbb{R}) \cap E[n]$. It turns out that this is either a cyclic group or it is isomorphic to $\mathbb{Z}/2 \times \mathbb{Z}/2m$ for some $2m$ dividing n. This can be seen quite easily on using the so-called Weil pairing and the fact that ± 1 are the only roots of unity in $\mathbb{R}$.
(ii) If E is an elliptic curve defined over $\mathbb{Q}$, then the subgroup $E(\mathbb{Q})_{tor}$ of all points of finite order in $E(\mathbb{Q})$ is a finite group (by the Mordell-Weil theorem). Using (i), it is either a cyclic group or it is isomorphic to $\mathbb{Z}/2 \times \mathbb{Z}/2m$ for some m.
(iii) It is a far more difficult problem to determine which subgroups of $\mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}$ occur as *n-torsion subgroups* of an elliptic curve over K. For example, Mazur proved that over $\mathbb{Q}$, only finitely many (exactly 15) groups can occur as torsion groups.
(iv) Over general number fields K, it is a result of Merel that the order of the torsion group $E(K)_{tor}$ is bounded purely in terms of the degree $[K : \mathbb{Q}]$.

We saw in the above discussion that for each integer n and any elliptic curve E, there is a map $[n] : E(K) \to E(K)$ defined by $P \mapsto nP$. This is an example of an isogeny, when $n \neq 0$. In general :
A non constant morphism $\phi : E_1 \to E_2$ between elliptic curves E_1 and E_2 such that $\phi(\mathcal{O}) = \mathcal{O}$ is called an isogeny.
Therefore, an isogeny must be surjective and must have finite kernel. In fact, the rigidity property of projective varieties implies that an isogeny must be a group homomorphism. Moreover, as a trivial consequence of the fact about the fibres of a morphism of curves, we see that, if $\phi : E_1 \to E_2$ is an isogeny, then, $\#\ \mathrm{Ker}\phi = \deg_{sep} \phi$.

One denotes by $\mathrm{Hom}(E_1, E_2)$, the abelian group of all isogenies from E_1 to E_2.

Elliptic curves over finite fields

Let E be an elliptic curve E defined over a finite field F_q. The groups $E(F_q)$ are the groups which arise in cryptological applications. In fact, typically they will be elliptic curves over Q and one would reduce the coefficients mod p for a prime p and consider the elliptic curve over F_p. Given E over Q, this can be done for almost all primes p. The most important information encoded in E over F_q is the order $|E(F_q)|$. Given an equation for E, to find the order of this group is a nontrival problem. A theorem of Hasse tells us that this order must be within an error of $2\sqrt{q}$ from the number $q+1$ of points on the projective line on F_q. The order of $E(F_q)$ can be computed in terms of a certain isogeny known as the Frobenius isogeny which is defined as the map $(x, y) \mapsto (x^q, y^q)$ on points $(x, y) \neq \mathcal{O}$. If E is defined over F_q and $\pi_{q,E} : E \to E$ is the Frobenius morphism, then note that $E(F_q) = \mathrm{Ker}\ (1 - \phi_{q,E})$. The Frobenius isogeny is purely inseparable, of degree q. The trace t of the Frobenius plays a fundamental role in the theory of elliptic curves over F_q. We shall see in a while that a result of Hasse tells us that $|t| \leq 2\sqrt{q}$. One has, for every $(x, y) \in E$,

$$(x^{q^2}, y^{q^2}) + [t](x^q, y^q) + [q](x, y) = \mathcal{O}.$$

To discuss these aspects in some detail, we recall some notions which will be used.

Remarks
(a) The notion of a dual isogeny is defined by the characterizing property: *Let $\phi : E_1 \to E_2$ be an isogeny. Then, there is a unique isogeny $\hat{\phi} : E_2 \to E_1$ satisfying $\hat{\phi} \circ \phi = [\deg \phi]$. $\hat{\phi}$ is called the dual of ϕ.*
A more down-to-earth description of $\hat{\phi}$ is as follows:

$$\hat{\phi}(y) = [\deg_{insep}\phi]\{\sum_{z \in \phi^{-1}(y)} z - \sum_{w \in \mathrm{Ker}\phi} w\} = [\deg\phi](z)$$

for any $y \in E_2$ and any $z \in \phi^{-1}(y)$.
(b) For $K = C$, an isogeny $\phi : C/L \to C/L'$ has degree $d = [L' : \phi(L)]$.

Thus, $dL' \subseteq \phi(L) \subseteq L'$. Then, $\hat{\phi} : C/L' \to C/L$ is the map d/f where ϕ is 'multiplication by f'.

As we noted, an isogeny has finite kernel. On the other hand, here is a rather startling fact:
Let E_1 and E_2 be isogenous elliptic curves defined over F_q. Then $\#E_1(F_q) = \#E_2(F_q)$.

Some important facts on dual maps are :
(i) $\widehat{\phi + \psi} = \hat{\phi} + \hat{\psi}$
(ii) $\widehat{[n]} = [n]$
(iii) deg $[n] = n^2$
(iv) deg $\hat{\phi} = \deg \phi$
(v) $\hat{\hat{\phi}} = \phi$
(vi) deg $(-\phi) = \deg \phi$
(vii) $d(\phi, \psi) :=$ deg $(\phi + \psi) -$ deg $\phi -$ deg ψ is symmetric, bilinear on Hom(E_1, E_2) , where $E._1, E_2$ are elliptic curve over a field, and
(viii) deg $\phi > 0$ for any isogeny ϕ.

Hasse's theorem - Riemann hypothesis for elliptic curves

For an elliptic curve E defined over a finite field F_q, the most important parameter and the most obvious one that one can think of is the number of points in $E(F_q)$. Let us heuristically estimate $\#E(F_q)$; this will be one (corresponding to the point at infinity) more than the number of solutions (x, y) of the equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ with $x, y \in F_q$. Each value of x yields at the most two values of y and thus $\#E(F_q) \leq 1 + 2q$. Heuristically, one might expect a random quadratic equation (for y in terms of x) to have a solution with probability $1/2$. Thus, perhaps $\#E(F_q) \sim q+1$. As a matter of fact, we shall prove that this is true for any E upto an error of $2\sqrt{q}$ i.e., $|\#E(F_q) - q - 1| \leq 2\sqrt{q}$. This is a theorem of Hasse and, when rewritten in terms of the so-called zeta function of E, turns out to be analogous to the classical Riemann hypothesis.

Let E be an elliptic curve defined over F_q. Let $\pi_{q,E} : E \to E$ denote the Frobenius endomorphism. Recall that $\pi_{q,E}$ is a purely inseparable isogeny with $\deg \pi_{q,E} = q$.

Riemann hypothesis for elliptic curves (Hasse 1934).

Let E be an elliptic curve defined over F_q. Then,

$$|\#E(F_{q^n}) - 1 - q^n| \leq 2q^{n/2} \quad \forall n \geq 1.$$

Tate modules and the Weil pairing

We first recall the Tate module and its relation to isogenies on an elliptic curve. Let E be an elliptic curve defined over F_q. These are crucially used not only in the Weil conjectures for elliptic curves but have also been used in elliptic curve cryptography (the Menezes-Okamoto-Vanstone attack on elliptic curve DLP). Suppose ℓ is a prime not dividing q. We know that the ℓ^n-division points of E i.e., $E[\ell^n] \stackrel{d}{=} \text{Ker } [\ell^n]$ is $\simeq Z/\ell^n \times Z/\ell^n$. The inverse limit of the groups $E[\ell^n]$ with respect to the maps $E[\ell^{n+1}] \stackrel{[\ell]}{\to} E[\ell^n]$ is the *Tate module* $T_\ell(E) = \lim_{\leftarrow} E[\ell^n]$. Since each $E[\ell^n]$ is naturally a Z/ℓ^n-module, it can be checked that $T_\ell(E)$ is a $Z_\ell(= \lim_{\leftarrow} Z/\ell^n)$-module. It is clearly a free Z_ℓ-module of rank 2.

Evidently, any isogeny $\phi : E_1 \to E_2$ induces a Z_ℓ-module homomorphism $\phi_\ell : T_\ell(E_1) \to T_\ell(E_2)$. In particular, we have a representation : End $(E) \to M_2(Z_\ell); \phi \mapsto \phi_\ell$, if $\ell \not| q$. Note that End $E \hookrightarrow$ End $T_\ell(E)$ is injective because if $\phi_\ell = 0$, then ϕ is 0 on $E[\ell^n]$ for large n i.e., $\phi = O$.

Finally, let us recall the *Weil pairing*. This is a non-degenerate, bilinear, alternating pairing

$$e : T_\ell(E) \times T_\ell(E) \to T_\ell(\mu) \stackrel{d}{=} \lim_{\leftarrow} \mu_{\ell^n} \cong Z_\ell.$$

It has the important property that $e(\phi x, y) = e(x, \hat{\phi} y)$.

Weil conjectures for elliptic curves

In 1949, A. Weil made a series of general conjectures about varieties defined over finite fields. Let us use the notation $K_n = F_{q^n}$. If V is a projective variety defined over K_1 (i.e., the zero set of a collection of homogeneous polynomials with coefficients from K_1), we want to keep account of the number $\#V(K_n)$. The natural way to do this is by means of a generating function which codifies the data. This is known as the zeta function of V and is defined as the formal power series

$$Z(V/K_1; T) = \exp\left(\sum_{n=1}^{\infty} \#V(K_n) \frac{T^n}{n}\right)$$

Note that $\#V(K_n) = \frac{1}{(n-1)!}\frac{d^n}{dT^n}\log Z(V/K_1;T)]_{T=0}$. The reason for defining the zeta function in this manner is that the series $\sum_{n\geq 1}\#V(K_n)\frac{T^n}{n}$ often looks like the log of a rational function of T.

The following result is crucial for the Weil conjectures for elliptic curves and uses Weil pairings in its proof.
Let $\phi \in$ End (E) and $\ell \not| q$ be a prime. Then,

$$\begin{aligned} det\ \phi_\ell &= deg\ \phi, \\ tr\phi_\ell &= 1 + \deg\phi - deg\ (1-\phi). \end{aligned}$$

In particular, det $\phi_\ell, tr\phi_\ell$ are independent of ℓ, and are integers.

A consequence of this result is the evident fact that the characteristic polynomial of ϕ_ℓ has coefficients in Z when $\ell \neq$ char F_q.
Write det $(T.Id - \phi_\ell) = (T-\alpha)(T-\beta); \alpha, \beta \in C$.
Moreover, $\forall\ \frac{m}{n} \in Q$, we get

$$\det\ \left(\frac{m}{n}Id - \phi_\ell\right) = \frac{1}{n^2}det(mId - n\phi_\ell) = \deg(m - n\phi)\frac{1}{n^2} > 0.$$

This implies $\alpha = \bar{\beta}$. Noting, by triangularising, that det $(Id.T - \phi_\ell^n) = (T-\alpha^n)(T-\beta^n)$, we get:

Theorem *For all $n \geq 1$, $\#E(K_n) = 1 - \alpha^n - \bar{\alpha}^n + q^n$ where $|\alpha| = q^{1/2}$. In particular, $Z(E/K_1;T) = \frac{1-aT+qT^2}{(1-T)(1-qT)}$, where $a \in Z$ and $1 - aT + qT^2 = (1-\alpha T)(1-\bar{\alpha}T)$. Further, $Z\left(E/K_1; \frac{1}{qT}\right) = Z(E/K_1;T)$.*

Remark Putting $\zeta_{E/F_q}(s) = Z(E/K_1; q^{-s})$, one has

$$\zeta_{E/F_q}(s) = \frac{1 - aq^{-s} + q^{1-2s}}{(1-q^{-s})(1-q^{1-s})} = \zeta_{E/F_q}(1-s).$$

Note that the Riemann hypothesis for $Z(E/K_1;T)$ is equivalent to the fact that the zeroes of $\zeta_{E/F_q}(s)$ are on the line $Re(s) = \frac{1}{2}$.

Supersingularity

Supersingular curves are a special class of elliptic curves which arise naturally. One of the properties they have is that their definition forces them to be

defined over a small finite field and, over any field, there are only finitely many elliptic curves isogenous to a supersingular one. An elliptic curve E defined over a field of characteristic $p > 0$ is said to be *supersingular* if $E[p] = O$.
The following characterisation of supersingular elliptic curves is useful and not hard to prove.

Let K be a perfect field of characteristic $p > 0$. Then, the following statements are equivalent:
(a) E is supersingular.
(b) $[p] : E \to E$ is purely inseparable and $j(E) \in F_{p^2}$.
(c) $E[p^r] = \{O\}$ for some $r \geq 1$.
(d) $E[p^r] = \{O\}$ for all $r \geq 1$.
(e) $End_{\bar{K}}(E)$ is an order in a quaternion division algebra over Q.

For $p = 2, Y^2 + Y = X^3$ is the unique supersingular curve.
For $p > 2$, we have :

Theorem *Let $K = F_q$ with char $K = p$ odd.*
(i) Let $f(X) \in K[X]$ be a cubic polynomial with distinct roots in $\bar{K}$ and E be the elliptic curve defined by the equation $Y^2 - f(X) = 0$. Then E is supersingular $\Leftrightarrow$ coefficient of X^{p-1} in $f(X)^{\frac{p-1}{2}}$ is 0.
(ii) Consider the Deuring polynomial $H_p(t) = \sum \begin{pmatrix} \frac{p-1}{2} \\ i \end{pmatrix} t^i$. Let $\lambda \in \bar{K}, \lambda \neq 0, 1$. Then, the elliptic curve $E : Y^2 = X(X-1)(X-\lambda)$ is supersingular $\Leftrightarrow H_p(\lambda) = 0$.

As a corollary, here is a criterion for an elliptic curve over a field of positive characteristic to be supersingular.

Corollary *Let $K = F_p$. Then the elliptic curve E_λ defined by the equation $Y^2 = X(X-1)(X-\lambda)$ is supersingular if and only if $\#E_\lambda(F_p) = p + 1$.*

(Mass formula) $\frac{p-1}{24} = \sum \frac{1}{Aut(E)}$ *where the sum is over isomorphism classes of supersingular elliptic curves over a field of characteristic $p > 0$.*

Structure of $E(F_q)$

In this section, we finally point out what possible groups can arise as groups of rational points of elliptic curves over finite fields:
A group G of order $N = q + 1 - m$ is isomorphic to $E(F_q)$ for some elliptic

curve E over F_q if, and only if one of the following holds:
(i) $(q, m) = 1, |m| \leq 2\sqrt{q}$ *and* $G \cong Z/A \times Z/B$ *where* $B/(A, m-2)$.
(ii) q *is a square,* $m = \pm 2\sqrt{q}$ *and* $G = (Z/A)^2$ *where* $A = \sqrt{q} \mp 1$.
(iii) q *is a square,* $p \equiv 1(3), m = \pm\sqrt{q}$ *and* G *is cyclic.*
(iv) q *is not a square,* $p = 2$ *or 3,* $m = \pm\sqrt{pq}$ *and* G *is cyclic.*
(v) q *is not a square,* $p \not\equiv 3(4)$, $m = 0$ *and* G *is cyclic*
or q *is a square,* $p \not\equiv 1(4), m = 0$ *and* G *is cyclic.*
(vi) q *is not a square,* $p \equiv 3(4), m = 0$ *and* G *is either cyclic or* $G \cong Z/M \times Z/2$ *where* $M = \frac{q+1}{2}$.

5 Elliptic curve cryptosystems

Problems like the discrete log and others discussed make sense for any finite abelian group. Of course, the complexity depends on the group. For instance, the discrete log is easy for the additive group Z/nZ whereas it is hard for F_q^*. Our choice of group should be such that the group operations are simple enough to program on a computer and the elements of the groups should be stored conveniently on a computer. A general-purpose algorithm due to Pohlig & Hellman shows that a problem like the discrete log problem on a group reduces easily to solving the problem over a prime order subgroup. Thus, our group should have large prime order subgroups in order that cryptosystems based on such a group is not vulnerable. In 1986, Miller and Koblitz came up with the idea of basing cryptosystems on the group $E(F_q)$ for an elliptic curve E. Indeed, it turns out that the discrete log problem over such a group is, at present, orders of magnitude harder than the corresponding problem on a finite field of comparable cardinality. There are other groups like the class groups of orders in number fields based on which, cryptosystems have been studied. *It can be proved that these cryptosystems are as hard to break as factoring integers is.* However, the group operation here is rather complex to perform quickly enough. Once the DLP is analysed over elliptic curves, it is easy to see that it can be used to do a Diffie-Hellman key exchange. Before we start studying cryptosystems based on elliptic curves, we point out why elliptic curves are better to use than, say, F_q^*. First of all, there are many curves over the same field (there is a choice of about q^2 over F_q) and, for the same level of security as, say RSA, the key sizes needed are much smaller. For instance, an elliptic curve group over a 160-bit prime compares favourably with F_p^* for a 500-bit prime. Another advantage is due to the following fact. The so-called index calculus method can be used fruitfully for attacking the discrete log problem in F_p^*'s because there is a wide choice for the so-called factor bases since Q^* is infinitely generated (there are infinitely many prime numbers). But, as $E(Q)$ is finitely generated, such choices of factor bases may not be possible.

We shall not discuss analogues of RSA involving elliptic curves as the performance is worse in comparison with RSA and it has not been proved that breaking them is equivalent to factoring. We discuss the discrete log problem and the 'Baby step giant step' method of solving it. There is another method depending on random walks known as the *method of tame and wild*

kangaroos which is of comparable efficiency but we do not discuss it here. We also point out that cryptosystems like El Gamal, Massey-Omura etc. can be described over elliptic curves.

Representing plaintext on E

Before proceeding further, we would like to discuss how plaintext can be represented as points on an elliptic curve. Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over F_q where $q = p^r$ is assumed to be large and odd. Since this will be a probabilistic method of representing plaintext on $E(F_q)$, we fix a number k and we would be bounding the probability of failure by $1/2^k$. In practice, $k = 50$ does the job. Let M be so large that the plaintext is represented as a string of integers m with $0 \leq m < M$. On the other hand, the finite field is taken to be so large that $q > Mk$. We write the integers $< Mk$ in the form $mk + j$ with $1 \leq j \leq k$. These can be regarded as distinct elements of F_q. This sets up a 1-1 correspondence from the set of integers in $[0, Mk]$ to a set of elements in F_q. A number $mk + j$ would be the x co-ordinate of a point on $E(F_q)$. We compute the square-roots of $x^3 + ax + b$ in F_q if they exist. If they do, call one of them y and we have a point $P_m = (x, y)$ on $E(F_q)$. If there is no square-root in this case, then change j to $j + 1$ and proceed. If we can get a square by the time j reaches k, we can recover $m = [(\tilde{x} - 1)/k]$, where $\tilde{x}$ is the integer corresponding to the element x of F_q. Since $x^3 + ax + b$ will be a square half of the time, the probability of failing to find a square as j varies, is $1/2^k$.

Baby step giant step for DLP

Of course, this method is not special for elliptic curves but is common to all finite abelian groups G. Given elements $Q = mP$, one would like to determine the discrete log m. If the order of G is n, write $m = [\sqrt{n}]_0 a + b$ with $0 \leq a, b < [\sqrt{n}]_0$ where we have written $[*]_0$ for the upper ceiling function. Then

$$Q - bP = a[\sqrt{n}]_0 P.$$

One computes a table of values of $R_b = Q - bP$ as b ranges from 0 to $[\sqrt{n}]_0 - 1$. This is called the table of baby steps. After this, the giant steps $S_a = a[\sqrt{n}]_0 P$ are computed for a between 0 and $[\sqrt{n}]_0 - 1$. After computation of each S_a,

one checks with the table of baby steps to see if there is a match. Thus, a match is reached in $[\sqrt{n}]_0$ giant steps. The main drawback of the method is that one needs to store $[\sqrt{n}]_0$ group elements.

Here is an example :
Consider the elliptic curve

$$E : y^2 = x^3 + 71x + 602$$

over the field F_{1009}. We wish to solve for m satisfying $Q = (592, 97) = mP = m(32, 737)$. The point P has order 53 and the whole group $E(F_{1009})$ has order 1060. We need to work just in this subgroup of order 53 generated by P. Thus, in this example, $n = 53$. Note that $[\sqrt{n}]_0 = 8$, there are 8 baby steps. These values $R_b = Q - bP$ are tabulated below.

The table of baby steps $R_b = Q - bP$ are as follows :

b	R_b
0	(592,97)
1	(728,450)
2	(537,344)
3	(996,154)
4	(817,136)
5	(365,715)
6	(627,606)
7	(150,413)

The table of giant steps $S_a = 8aP$ are as follows :

a	S_a
1	(996,855)
2	(200,652)
3	(378,304)
4	(609,357)
5	(304,583)
6	(592,97)

Thus, we see there is a match for $a = 6, b = 0$ which means that $m = 8a+b = 48$.

Pohlig-Hellman's reduction of DLP - an example

The discrete log problem on any finite abelian group can easily be reduced to prime power order subgroups using the Chinese remainder theorem. Further, suppose $P \in G$ has order p^r, and that we wish to solve $Q = mP$. Note that m is determined mod p^r. One considers $Q' = p^{r-1}Q = mP'$ where $P' = p^{r-1}P$ has order p. If we solve $Q' = mP'$, we can determine m mod P. Call this m_0. One can then look at $Q - m_0P = (\frac{m-m_0}{p})(pP)$. As before, we can solve for $\frac{m-m_0}{p}$ mod p. Call this m_1. Then, we would have $m \equiv m_0 + m_1 p \bmod p^2$. Proceeding in this manner, we would have all the p-adic digits of m.

Let us look at an example to see how the whole procedure of applying Chinese remainder theorem, baby step giant step etc. works.
We consider once again the elliptic curve

$$E : y^2 = x^3 + 71x + 602$$

over the field F_{1009}. Suppose we wish to solve $Q = mP$ where $Q = (190, 271)$ and $P = (1, 237)$. One can compute the order of P and find it to be 530. As $530 = 2.5.53$, we need to solve this discrete log problem mod 2, mod 5 and mod 53.
To solve mod 2, look at $Q_2 = 265Q = mP_2$ where $P_2 = 265P$. It is easily computed that the points Q_2, P_2 of order 2 are the same, viz., $(50, 0)$. One finds that the solution of $(50, 0) = m(50, 0)$ is $m \equiv 1 \bmod 2$.
To solve mod 5, we compute $Q_5 = 106Q = (639, 849)$, $P_5 = 106P = (639, 160)$. Thus, $P_5 = -Q_5$, and so $m \equiv -1 \bmod 5$.
To solve mod 53, we compute $Q_{53} = 10Q = (592, 97)$ and $P_{53} = 10P = (32, 737)$. In this case, the solution of $Q_{53} = mP_{53}$ is not so easy as 53 is a somewhat large prime. Thus, one uses the baby step giant step method. This was exactly the example we gave to illustrate this method and we obtained $m \equiv 48 \bmod 53$.
Combining all these, one has by the Chinese remainder theorem the solution $m = 419$ for the original DLP.

Choice of E, F_q etc. for security of DLP

(i) *The group $E(F_q)$ should have a subgroup of a large prime order.*
'Large' here means that it should withstand the square-root finding algorithms in existence. At present, a 160-bit prime provides as much security as a public key cryptosystem based on F_p^* with 1000-bit key length.
(ii) *Avoid supersingular and anomalous curves.*

The reason is a method of attack (MOV) developed by Menezes, Okamaoto and Vanstone which embeds certain elliptic curves over F_q in the multiplicative group of F_{q^l} for some l, where the DLP is easier to attack. They use Weil pairing on $E[n]$. This MOV attack can work (as shown by Balasubramanian and Koblitz) only when the elliptic curve is supersingular or when the order of q mod n $(= |E(F_q)|)$ is small, or when $q = n = p$ (such curves are called anomalous). Thus, in addition to avoiding supersingular and anomalous curves, one needs to choose curves for which the order of q mod n is large.

El Gamal, Massey-Omura over E

Given our earlier formulations which were as general as possible, it is quite clear how to describe over elliptic curve groups the various cryptosystems like El Gamal and Massey-Omura. Also, the Diffie-Hellman key exchange can be done fruitfully as we know that the DLP is intractable if we make proper choices of E, q etc.
Indeed, for Diffie-Hellman, Alka and Beena exchange a secret key P as follows. They choose a public E, F_q and a point $Q \in E(F_q)$ of large enough size (comparable to $|E(F_q)|$). Then, Alka and Beena choose their private keys to be random integers a and b in the range $(1, |E(F_q)|)$. They make public the points aQ and bQ respectively. Both of them can compute abQ which they will take as their private key P.

Elliptic curve factoring method

The most popular method is the ECM method due to H.W.Lenstra which is analogous to Pollard's $p-1$ method. The special fact which allows for a wide choice of elliptic curves mod p is :

Lemma.
There is a constant $1 > c > 0$ such that among all pairs of points $(a, b) \in F_p \times F_p$ with $4a^3 + 27b^2 \neq 0$, there are cp^2 points (a, b) such that the curve

$$E : y^2 = x^3 + ax + b$$

over F_p satisfies $| \, |E(F_p)| - p| < \sqrt{p}$.
Thus, there is a probability of c of choosing an elliptic curve whose order is in the above interval.

Lenstra's ECM works as follows. To conveniently describe the method, let us consider the case $n = pq$ where p, q are primes. One considers an elliptic curve E over the *ring* Z/nZ. One needs to be careful as the ring has zero divisors. So, one writes the group law in terms of projective co-ordinates. By the Chinese remainder theorem, $E(Z/nZ) \cong E(F_p) \times E(F_q)$. Since there is a large amount of choice of E possible, we are more likely (than the usual Pollard's $p-1$ method) to find a curve for which the order of $E(F_p)$ is a B-smooth number for some reasonable B. First, one finds a curve with a projective Z/nZ-point; that is,

$$E_{a,b} : y^2z = x^3 + axz^2 + bz^3$$

and $(x, y, z) \in E_{a,b}(Z/nZ)$. If we assume that the discriminant Δ of the curve is coprime to n, then $E_{a,b}$ has good reduction mod p (and mod q). Considering it as a curve over F_p, Hasse's theorem gives

$$| \, |E_{a,b}(F_p)| - p - 1| < 2\sqrt{p}.$$

Choosing a constant B, let $k(B) = LCM(1, 2, \cdots, B)$.
We compute $k(B)(x, y, z)$ mod n, say, (x_B, y_B, z_B). If $|E_{a,b}(F_p)|$ divides $k(B)$, then p divides z_B and we may be able to find factor as (z_B, n). In practice, such a factor is found while computing the multiple $k(B)(x, y, z)$ where some inversion will be impossible due to the zero divisors in Z/nZ. We demonstrate this by an example now.

Example.
Let $n = 187$ and let E be $y^2 = x^3 + x + 25$. Consider the point $P = (0, 5)$. Take $B = 3$. Then $k(B) = 6$. Recall that the slope m of the line joining two points $(x_1, y_1), (x_2, y_2)$ on $y^2 = x^3+ax+b$ is given as $m = (y_2-y_1)/(x_2-x_1)$ or $m = (3x_1^2+a)/2y_1$ according as whether the points coincide or not. Then, the point P_3 which is their sum is given as $x_3 = m^2-x_1-x_2, y_3 = m(x_1-x_3)-y_1$.
To compute $2P$ in our case, we find $m = 1/10 \equiv -56$ mod 187; thus $2P = (-43, 18)$.
Now, to compute $4P = 2P + 2P$, we have $m = -62/36 \equiv 71$ mod 187 and $4P = (78, -7)$.
To compute $6P = 2P + 4P$, we find $m = 25/66$. Note that 66 cannot be inverted since $(66, 187) = 11$. We have thus found a factor.

Elliptic curves for primality

The principal method is an analogue of Pocklington's method which we discussed; this is due to Goldwasser & Kilian. We do not discuss it in detail but point out the result that it is based :

Let E be an elliptic curve over Z/nZ where $(n, 6) = 1$. Suppose we can find a point $P \in E(Z/nZ)$ and an integer m such that $mP = 0$ but for some prime divisor $l > (n^{1/4} + 1)^2$ of m, one has $(m/l)P \neq 0$. Then n is prime. If neither of the two multiplications mP and $(m/l)P$ can be made, then one has found a proper factor of n as in ECM. Further, if n is indeed prime, such a point P as aove does exist if $E(Z/nZ)$ has order m which has a prime factor $l > (n^{1/4} + 1)^2$.

Counting the order

It is clear that generating elliptic curves over finite fields whose orders are suitable for our applications is very important in any of the protocols involving elliptic curves. The method due to Rene Schoof was path-breaking and is currently the basis of any efficient scheme for counting points; we will discuss it very briefly here.
Now, Hasse's theorem gives $|E(F_q)| = q + 1 - t$ where $|t| < 2\sqrt{q}$.
The key point of Schoof's algorithm is to count the trace t of the Frobenius mod primes $l \leq l_{max}$ where l_{max} is the smallest prime for which $\prod\{l : l$ prime, $l \leq l_{max}\} > 4\sqrt{q}$.
One can then apply the Chinese remainder theorem to obtain the order of $E(F_p)$.