A Brief Introduction to Public-Key Cryptography

Project submitted to St. Xavier's College (Autonomous) for the degree of Bachelor's in science by-

Raktim Dey

Roll- 12

Department of Mathematics

St. Xavier's College (Autonomous) Kolkata

Declaration from the student

This project entitled “A brief Introduction to Public-Key Cryptography” is being submitted by me to St. Xavier’s College (Autonomous) Kolkata to finish the degree of Bachelor’s in Science which consists of research work done by me, under the supervision of Rabiul Islam, Assistant Professor (Mathematics). The research work is not original and is inspired by numerous books and internet resources mentioned in the Bibliography section.

Raktim Dey

Acknowledgements

I would like to thank a lot of people for helping me with this project, but then I'd be lying. Nevertheless, I would like to thank my advisor, Professor Rabiul Islam for helping me anytime he could and for also constantly making sure that I was working on my project.

Secondly, I'd like to thank Professor Gaurab Tripathi for accepting my calls for every simple doubts at all hours, helping me out with them and various other problems that I have faced.

Lastly, I'd like to thank the internet, my classmate Nafisa and my senior Anugata Di for helping me with my project.

CONTENTS

Introduction to Cryptography

Terminology

".. if there was occasion for secrecy, he wrote in cypher; that is, he used the alphabet in such a manner, that not a single word could be made out. The way to decipher these epistles was to substitute the fourth for the first letter, as d for a, and so for the other letters respectively." – Suetonius wrote this of Julius Caesar in "The lives of the Twelve Caesars".

Cryptography is the science of communicating messages in secret, i.e., by disguising messages. The process of hiding the messages is called enciphering and figuring it out is called deciphering. The study of mathematical techniques to decrypt the messages is called cryptanalysis.

The word cryptography comes from the Greek word kryptos and logos, meaning hidden and word respectively. The term was coined by James Howell in 1645. The first recorded cryptographic instance was written in stone almost four millennia ago by an Egyptian scribe who used hieroglyphic symbol substitution in his writing on a rock wall in the tombs of a nobleman of the time, Khnumotep.

The intention was not to disguise the inscription, but rather to add some majesty to his inscription of the nobleman's deeds. The most widely used cryptographic method was invented by Julius Caesar, known as Caesar Cipher.

Caesar Cipher

In this method, we assume the numerical equivalent of 26 alphabets as elements of $\mathbb{Z}_{26}$, where A=0, B=1,…Z=25.

According to Caesar, every fourth letter was to be replaced by the fourth one, i.e., D->A.

In mathematical terms, if the encoded alphabet is α, then the decoded alphabet is:

$\beta \equiv \alpha + 3 \pmod{26}$.

So, to recover, we write:

$\alpha \equiv \beta - 3 \pmod{26}$.

For example, let us say we want to encode the message – "CATS ARE BETTER THAN DOGS"

We find the numerical equivalent of every letter:

C A T S A R E B E T T E R T H A N D O G S

2 0 19 18 0 17 4 1 4 19 19 4 17 19 7 0 13 3 14 6 18

Now we add 3 to every number to obtain-

5 3 22 21 3 20 7 4 7 22 22 7 20 22 10 3 16 6 17 9 21

Converting into alphabets, we get:

FDWV DUH EHWWHU WKDQ GRJV

To decode it, we subtract 3 from every number and get an original message.

Enciphering and Deciphering functions

An enciphering function is an injective function E_e:M $\rightarrow \mathbb{C}$.

where the key e decides $E_e(m)=c \in \mathbb{C}$, and m is the message unit from M. In other words, how many letters are to be shifted in order to encode the message depends on the value of e.

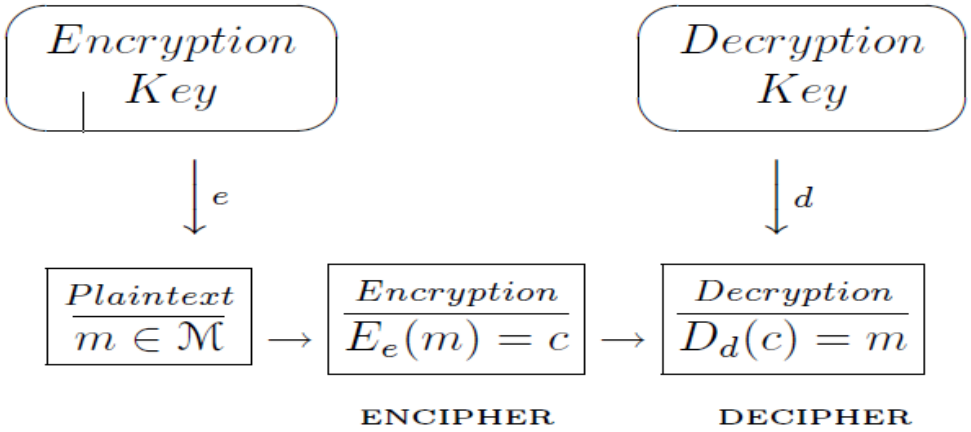
A deciphering function is also an injective function D_a: $\mathbb{C} \rightarrow M$,

$D_a(c)=m \in M$, where c is the encoded text.

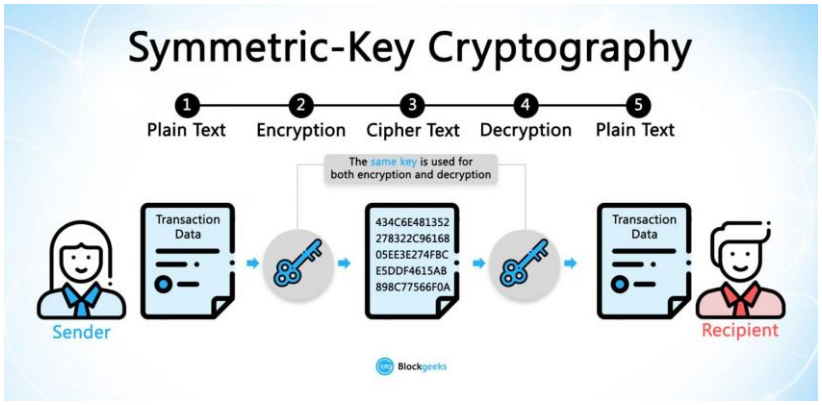
In Caesar cipher, e is taken as 3, a is the additive inverse of e in $\mathbb{Z}_{26}$, which is 23.

Cryptography

A cryptosystem consists of a set of enciphering functions $\{ E_e : e \in \Re \}$ and another set of deciphering functions $\{ D_a = E_e^{-1} : a \in \Re \}$, such that for each $e \in \Re$, $\exists d \in \Re$ such that $E_e = D_a^{-1}$, so $D_a \circ E_e(m) = m \ \forall m \in M$, where $\Re$ is an arbitrary set and M is a set of message units. 'e' and 'd' are called keys, and (e,d) is called a key pair.



An Illustrated Cryptosystem



In general, symmetric-key cryptosystems are easy to crack/decipher because for every key pair (e,d), we can easily calculate d if we know e and vice versa. But in asymmetric cryptosystem, we cannot very easily calculate d from e. Public-Key Cryptography is an example of asymmetric cryptosystem.

There are two types of symmetric cryptosystems, one of them is called a Block cipher.

Block cipher

In this method, the original message or plaintext is divided into blocks of a fixed length, all of the same length and encrypts each of them by the same encryption algorithm. Let us see how one of the earliest types of block ciphers- Hill cipher works.

Let the message be "KNIFE BELOW TABLE".

We divide the message into blocks of length n=2. Here we see that the number of letters are odd, so we add an extra letter Z in the end to make it even. So, we get:

K N I F E B E L O W T A B L E Z

The key of the hill cipher is an n*n matrix M where n is the block length, which must have an inverse in $\mathbb{Z}_{26}$, i.e., M must be non-singular.

Let us take M=$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}$.

So, $M^{-1}=1/5\begin{pmatrix}5 & -4\\-5 & 5\end{pmatrix}=21\begin{pmatrix}5 & -4\\-5 & 5\end{pmatrix}=\begin{pmatrix}105 & -84\\-105 & 105\end{pmatrix}=\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}(mod\ 26)$.

1/5 is taken as 21, because 21 is the inverse of 5 in $\mathbb{Z}_{26}$, as all the co-efficients are also taken modulo 26.

Now, we convert the blocks of length 2 into 2*1 matrices and pre-multiply them with M to encode them.

Writing the numerical equivalent of alphabets in matrix form, we get:

KN=$\begin{pmatrix}10\\13\end{pmatrix}, IF=\begin{pmatrix}8\\5\end{pmatrix}, EB=\begin{pmatrix}4\\1\end{pmatrix}, EL=\begin{pmatrix}4\\11\end{pmatrix}, OW=\begin{pmatrix}14\\22\end{pmatrix}, TA=\begin{pmatrix}19\\0\end{pmatrix}, BL=\begin{pmatrix}1\\11\end{pmatrix}, EZ=\begin{pmatrix}4\\25\end{pmatrix}$.

Now, we encode them by-

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}10\\13\end{pmatrix}=\begin{pmatrix}102\\115\end{pmatrix}=\begin{pmatrix}24\\11\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}8\\5\end{pmatrix}=\begin{pmatrix}60\\65\end{pmatrix}=\begin{pmatrix}8\\13\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}4\\1\end{pmatrix}=\begin{pmatrix}24\\25\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}4\\11\end{pmatrix}=\begin{pmatrix}75\\71\end{pmatrix}=\begin{pmatrix}23\\19\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}14\\22\end{pmatrix}=\begin{pmatrix}54\\50\end{pmatrix}=\begin{pmatrix}2\\24\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}19\\0\end{pmatrix}=\begin{pmatrix}95\\95\end{pmatrix}=\begin{pmatrix}17\\17\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}1\\11\end{pmatrix}=\begin{pmatrix}60\\59\end{pmatrix}=\begin{pmatrix}8\\7\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}5 & 4\\5 & 5\end{pmatrix}\begin{pmatrix}4\\25\end{pmatrix}=\begin{pmatrix}16\\15\end{pmatrix}(mod\ 26).$$

Arranging the letters, we get the encrypted message-

YLINYZXTCYRRIHQP

In order to decrypt, we multiply the 2*1 matrices with M^{-1} to decode the encrypted matrices-

$$\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}\begin{pmatrix}24\\11\end{pmatrix}=\begin{pmatrix}218\\13\end{pmatrix}=\begin{pmatrix}10\\13\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}\begin{pmatrix}8\\13\end{pmatrix}=\begin{pmatrix}-70\\5\end{pmatrix}=\begin{pmatrix}8\\5\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}\begin{pmatrix}24\\25\end{pmatrix}=\begin{pmatrix}4\\1\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}\begin{pmatrix}23\\19\end{pmatrix}=\begin{pmatrix}4\\11\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}\begin{pmatrix}2\\24\end{pmatrix}=\begin{pmatrix}14\\22\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix}1 & 20\\25 & 1\end{pmatrix}\begin{pmatrix}17\\17\end{pmatrix}=\begin{pmatrix}19\\0\end{pmatrix}(mod\ 26).$$

$$\begin{pmatrix} 1 & 20 \\ 25 & 1 \end{pmatrix}\begin{pmatrix} 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 1 \\ 11 \end{pmatrix} (mod\ 26).$$

$$\begin{pmatrix} 1 & 20 \\ 25 & 1 \end{pmatrix}\begin{pmatrix} 16 \\ 15 \end{pmatrix} = \begin{pmatrix} 4 \\ 25 \end{pmatrix} (mod\ 26).$$

Arranging the integers sequentially and converting them into alphabets, we get back

KNIFEBELOWTABLEZ. So, we see one of the most important applications of matrix in real life- encoding and decoding messages by multiplication.

This project also contains various methods of decryption that have been introduced over time by several mathematicians of our age, introductions to the Discrete Log problem, the Diffie-Hellman Key Exchange, methods like Baby Step-Giant Step and Index Calculus method to solve these important problems which has several applications in the field of network security. Also, Elliptic Curve Cryptography, the faster method of solving the above problems while using relatively less time is also mentioned.

All problems solved in this project to further explain the many methods of decrypting ciphers have been solely done by me with the help of Python Programming Language and not copied or influenced by any reference material mentioned in the bibliography or otherwise.

Hence, any student interested in learning more about Cryptography can use this document as a reference material.

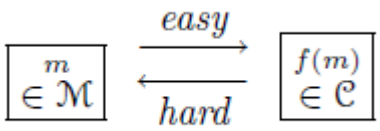
PUBLIC-KEY CRYPTOGRAPHY

Let's assume that Bob and Alice, two spies who haven't met before, want to communicate secret information with each other but they are in different countries. They could use any symmetric cryptographic system, but since they haven't met before, so it's not possible for them to have agreed to a key. Now Alice could send a courier to Bob with the secret key locked in a box but that might take a few days, which isn't plausible either. So, here comes the idea of Public-Key Cryptography.

Alice generates two keys, one called the public key, and the other the private key. Alice sends the public key to Bob, so Bob can encrypt the message and send it to Alice, and then only Alice can decrypt it using the private key. This is called an asymmetric cryptosystem.

Definition: One-way functions

A function f: M→ $\mathbb{C}$ is called a one-way function if and only if f is one-one, and f(m) is "easy" to compute ∀m∈M, but for some c∈$\mathbb{C}$, $f^{-1}(c)=m$ is computationally infeasible, i.e., we can calculate f, but it is almost impossible to find f^{-1}.



It has never actually been proved that one-way functions actually exist, because there is no rigorous definition of "computationally easy" or "computationally infeasible".

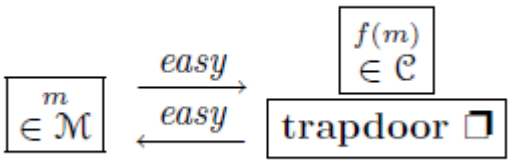
Let us see a common example of illustrating a one-way function: coin-flipping over the telephone:

(1) Alice and Bob both know a one-way function f but not f^{-1}.
(2) Bob selects some integer x, calculate y=f(x) and sends it to Alice. We assume that" Heads" refers to x being odd, "Tails" even.
(3) Alice calls either Heads or Tails.
(4) Bob tells her if her guess is correct.
(5) Bob sends x to Alice and she calculates f(x) to verify if Bob was being honest.

It's security depends upon f being one-one and producing y with equal probability of being odd or even.

Definition: Trapdoor one-way functions

A trapdoor one-way function is also a one-way function f: M→ ℂ, satisfying an extra property that there exists information, called trapdoor, which makes it feasible to find m∈M such that f(c)=m, for any c∈ℂ, which is to find f^{-1}, but without the trapdoor, this task is computationally infeasible.



The basic idea of Diffie-Hellman problem makes use of one-way functions, which we shall see later.

In order to explain the idea of "computationally infeasible" to evaluate the inverse function, let's take the example of n=pq, where p and are large primes. If we are only given the value of n, evaluating Φ(n)[=(p-1)(q-1)], no. of integers less than and coprime with n, it is the same as factoring n. If we have (p-1)(q-1), then we can find p and q by calculating p+q=n-(p-1)(q-1)+1, and p-q-=$\sqrt{(p+q)^2-4n}$, so we get p=$\frac{(p+q)+(p-q)}{2}$, $q=\frac{(p+q)-(p-q)}{2}$.

The Discrete-Log problem

The generalised discrete log problem states that given a finite cyclic group G of order n, where G is generated by a, or G=<a>, and for some b ∈G, we need to find a unique e, 1≤e≤n-1, such that $a^e \equiv b$.

In our case, we are more concerned with a more specific case when G=Zp*, where p is an odd prime.

Let a be a generator of Zp*, b ∈Zp*. We are to find a unique non-negative integer e≤p-2(Since order of Zp* is p-1), such that $a^e \equiv b$(mod p). Then e is called index of b to the base of a mod p, or e≡ $\log_a$ b (mod p).

So, given a, b and a prime p, the problem of finding unique e<=p-2 is the Discrete Log Problem (DLP)-

e≡ $\log_a$ b (mod p). If p is properly chosen, this is a very difficult problem to solve. There are several methods of solving a Discrete Log problem. We now discuss one of those methods-

The Baby Step-Giant step method

This is a common method for solving discrete logs when p is not very large.

So, when G is a cyclic group of order n, <a>=G, b ∈G, we are to solve for x, where

x≡ $\log_a$ b (mod p).

The algorithm is:

(1) Find s =⌊√n⌋, where ⌊ ⌋ represents the greatest integer function.
(2) Baby step: For i=0, 1,…s-1, find a^ib(mod n).
(3) Giant step: For j=1, 2,…s, find a^{js}(mod n).
(4) Search the lists to find i and j such that a^ib= a^{js}. If so, then x=js-i(mod n), which is $\log_a$ b (mod n).

We now give an example to illustrate this process:

Let the problem be: $5^x \equiv 12$(mod 307).

We write it as x≡ $\log_5$ 12(mod 307). Here, a=5, b=12, p=307.

We find s=$\lfloor\sqrt{307}\rfloor=17$.

We calculate a^ib(mod n), i=0,1,..16.

i	0	1	2	3	4	5	6	7	8
a^ib(mod n)	12	60	300	272	132	46	230	229	224

i	9	10	11	12	13	14	15	16

$a^i b \pmod n$	199	74	63	8	40	200	79	88

Now we calculate $a^{js} \pmod n$, j= 1,2,..s=17.

j	1	2	3	4	5	6	7	8	9
$a^{js} \pmod n$	139	287	290	93	33	289	261	53	306

j	10	11	12	13	14	15	16	17
$a^{js} \pmod n$	168	20	17	214	274	18	46	254

We see that $a^5 b = 46 = a^{15s}$,

So x=(16*17)-5=267.

Hence, $267 \equiv \log_5 12 \pmod{307}$,

Or $5^{267} \equiv 12 \pmod{307}$.

Index Calculus algorithm

If p is a large prime, Baby Step- Giant Step method becomes inefficient, so in that case, we use index calculus method to solve the Discrete log problem,

$a^x \equiv b \pmod p$.

The algorithm is:

1. Choose a factor base B of small primes.
2. Compute $g_r \pmod p$ for many random values of r and try to factor the results using only primes from B.
3. Use combinations of the successes from Step 2 to evaluate log(q) for all primes q in B.
4. Compute $h \cdot g_r \pmod p$ for random values of r and try to factor these using only primes from B, If this happens, evaluate log(h) using the values of log(q) for $q \in B$.

To illustrate this process further, let us take an example:
Let the problem be: $3^x \equiv 31 \pmod{1579}$.

We write it as: $x = \log_3 31 \pmod{1579}$.

We select a base set of primes B = {2,5,7}.

Calculating $3^i \pmod{1579}$ for arbitrary integers i, we get:

$3^{537} \equiv 2.$

$3^{1238} \equiv 5.$

$3^{963} \equiv 7.$

So, in other words,

Log 2=537,

Log 5=1238,

Log 7=963.

Now, calculating 31×3^i for arbitrary i to get a value which is a product of powers of our base primes, we get:

$31\times3^{17}=560=2^4\times5^1\times7^1.$

Or, $31=2^4\times5^1\times7^1\times3^{-17}=(3^{537})^4\times3^{1238}\times3^{963}\times3^{-17}=3^{4332}.$

By Fermat's theorem, $3^{1578}\equiv1 \pmod{1579}$.

Hence, $3^{4332}\equiv(3^{1578})^2\times3^{1176}\equiv3^{1176}.$

Thus, $3^{1176}\equiv31 \pmod{1579}$,

or $\log_3 31\equiv1176 \pmod{1579}$.

RSA Cryptography

RSA is a type of asymmetric cryptosystem created by Ron Rivest, Adi Shamir and Leonard Adleman in 1977. The security of RSA relies on the practical difficulties of factoring n=pq, where p and q are large primes.

RSA Key Generation

Let us assume that Alice wants to send a secret message to Bob through a public communication system. So, Alice can't use a symmetric cryptosystem to send the message to Bob. Then-

(1) Bob finds two large primes p and q, where p≠q.
(2) Bob calculates n=pq and $\varphi(n)$=(p-1)(q-1).
(3) Bob chooses a random e, 1<e< $\varphi(n)$ and gcd(e, $\varphi(n)$)=1.
(4) Now, bob calculates a unique d, 1<d< $\varphi(n)$ such that ed=1(mod $\varphi(n)$) using Extended Euclidean algorithm, which states that gcd(a,b)=m means there exists some integers x and y such that ax+by=m.
(5) Bob tells e and n to Alice through a public database and keeps d, p, q and $\varphi(n)$ private. So (e,n) is the public key and d is the private key.

The problem for any eavesdropper Eve is to find d to decode the message. The problem depends on the difficulty of factoring n.

ENCIPHERING

Let us assume that the plaintext message *m* be converted to its numerical form, where m<n, gcd(m,n)=1. If m>n, the message is broken down into smaller blocks of length l, where n must lie between 26^l and 26^{l+1}.

Step 1: Alice obtains the public key (n,e) from the database.
Step 2: She enciphers m by calculating c≡ m^e(mod n) and sends it to Bob.

DECIPHERING
Bob receives message c and calculates m≡ $c^d(mod\ n)$.

The above process makes sense only when we explain how the deciphering works. So, we prove that part here.

Lemma: If n=pq, where p and q are primes, p≠q, and let there be integers e and d such that 1<e< $\varphi(n)$, 1<d< $\varphi(n)$, and ed=1(mod $\varphi(n)$). Then for all integers m,

$$m^{ed} \equiv m(mod\ n).$$

Case 1: When gcd(m,n)=1.

Then $m^{ed} \equiv m^{1+k\varphi(n)}$=m.($(m^{\varphi(n)})^k$ =m.1≡m(mod n).

Case 2: m>n, gcd(m,n)=n.

Then n divides m, and so m≡0(mod n).
So, $m^{ed} \equiv 0 \equiv \mathrm{m}(\mathrm{mod\ n})$.

Case 3: gcd(m,n)=p.

Then m ≡ 0(mod p).
Since gcd(m,q)=1,
By Fermat's theorem, $m^{q-1} \equiv 1$(mod q).
So, $m^{ed} = m^{1+(p-1)(q-1)}$ ≡m.1=m(mod q).
So, p and q both divides m^{ed}-m.
Hence, m^{ed} ≡m(mod n).

Let us give an example of this method to help understand it better:

Let the secret message be- WE ARE COMPROMISED MISSION ABORT

We choose n=p.q=57593.2281=131369633 and e=7. So $\varphi(n)$=(p-1)(q-1)=131309760.
Using extended Euclidean Algorithm, we find d=112551223.
Here, n lies between $26^5\ and\ 26^6$, so we will break the message into blocks of length 5.
Thus, converting the words into their numerical equivalent, we get:

WEARE=22×26^4+4×26^3+0×26^2+17×26^1+4×26^0=10124222.

COMPR=2×26^4+14×26^3+12×26^2+15×26^1+17×26^0=1168535.

OMISE=14×26^4+12×26^3+8×26^2+18×26^1+4×26^0=6614456.

DMISS=3×26^4+12×26^3+8×26^2+18×26^1+18×26^0=1587734.

IONAB=8×26^4+14×26^3+13×26^2+0×26^1+1×26^0=3910661.

ORTAA=14×26^4+17×26^3+19×26^2+0×26^1+0×26^0=6709300.

Now, after calculating the integer values of the messages, we encode it by raising it to the power of e modulo $\varphi(n)$.

10124222^7=30663662(mod ϕ(n)).

1168535^7=5837742(mod ϕ(n)).

6614456^7=58587825(mod ϕ(n)).

$1587734^7 = 118208897 (\mathrm{mod}\ \phi(n))$.

$3910661^7 = 103301908 (\mathrm{mod}\ \phi(n))$.

$6709300^7 = 108921494 (\mathrm{mod}\ \phi(n))$.

Now, converting the numbers into their alphabetical equivalent, we get:

$30663662 = 2\times26^5+15\times26^4+2\times26^3+16\times26^2+11\times26^1+16\times26^0 \equiv$ CPCQLQ.

$5837742 = 12\times26^4+21\times26^3+3\times26^2+18\times26^1+14\times26^0 \equiv$ MVDSO.

$58587825 = 4\times26^5+24\times26^4+5\times26^3+10\times26^2+9\times26^1+23\times26^0 \equiv$ EYFKJX.

$118208897 = 9\times26^5+24\times26^4+17\times26^3+15\times26^2+6\times26^1+1\times26^0 \equiv$ JYRPGB.

$103301908 = 8\times26^5+18\times26^4+1\times26^3+11\times26^2+12\times26^1+8\times26^0 \equiv$ ISBLMI.

$108921494 = 9\times26^5+4\times26^4+9\times26^3+4\times26^2+12\times26^1+6\times26^0 \equiv$ JEJEMG.

Thus, our encoded message becomes

CPCQLQMVDSOEYFKJXJYRPGBISBLMIJEJEMG.

This can be solved only by our key.

Now, let us see an example of decoding a message using RSA.

Let the encoded message be – HPUICIYQGVRQQWM.

We are given n=10765283, $\phi(n)=10758720$, l=5.

We break the encoded message into blocks of length 5, and then expressing the alphabets into their numerical equivalent, we get:

$HPUIC = 7\times26^4+15\times26^3+20\times26^2+8\times26+2\times26^0 = 3476202$.

$IYOGV = 8\times26^4+24\times26^3+16\times26^2+6\times26+21\times26^0 = 4088625$.

RQQWM=$17\times26^4+16\times26^3+16\times26^2+22\times26+12\times26^0$=8061208.

We choose e=11.

Then, by using Extended Euclidean Algorithm, we get-

11d+10758720 ϕ(n)=1.
Solving, we get d=1956131.

Raising the integer to the power d to decode them, we get:

3476202^d=$3476202^{1956131}$ =185453.

4088625^d=$4088625^{1956131}$ =82856.

8061208^d=$8061208^{1956131}$ =333944.

Now, converting them back into alphabetical form, we get:

185453=$10\times26^3+14\times26^2+8\times26+21\times26^0$=KNIV.

82856=$4\times26^3+18\times26^2+14\times26+20\times26^0$=ESOU.

333944=$19\times26^3+0\times26^2+0\times26+0\times26^0$=TAAA.

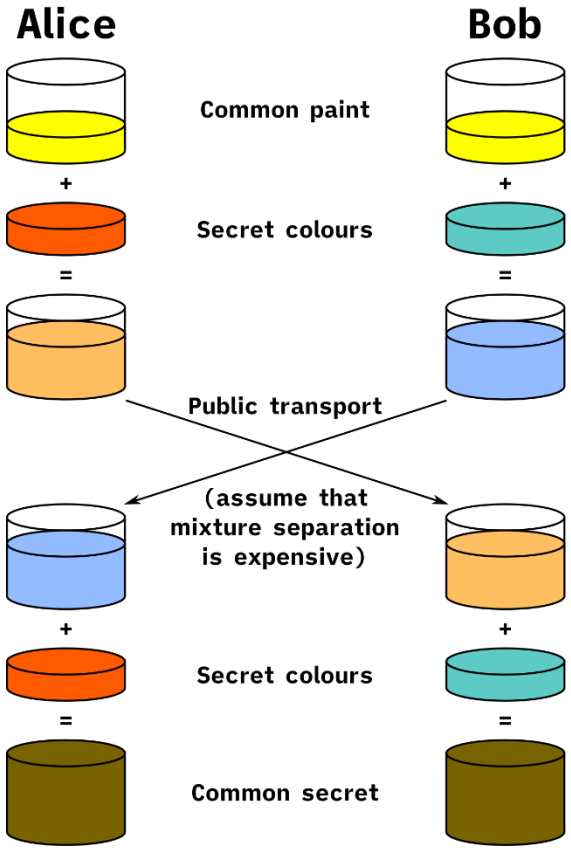
Hence, we get- KNIVESOUTAAA.

Removing the additional A's from the end, we get- KNIVES OUT.

DIFFIE-HELLMAN KEY EXCHANGE

Imagine for a moment that you have a perfect cryptography system, completely impenetrable and totally unreadable by anyone who intercepts an encrypted message. If you need to send classified information to agents thousands of miles away., they need the decoding key to read the message. This key cannot be communicated by public communicating devices because it can be intercepted. This is called the Key Exchange problem.

In 1976, Whitfield Diffie and Martin Hellman proposed a key exchange method that can be done publicly but nevertheless maintain the secrecy of the key. We will now describe this idea. Like RSA, which is another method of solving the "key" problem, all of the communications are done over public channels, like advertisements in newspaper, or even over public telephone booths (the ones that existed pre-Android). This key exchange problem is protected by the difficulty of computing the Discrete logarithms.



A simple example of the technique of Diffie-Hellman Key Exchange

(1) Alice and Bob agree on a large prime p and a primitive root g mod p (meaning, g is a generator of the multiplicative group Zp*).
(2) Alice chooses a secret integer a and calculates $h_1 \equiv g^a$(mod p).
(3) Bob chooses a secret integer b and calculates $h_2 \equiv g^b$(mod p).
(4) Alice sends h_1 to Bob and Bob sends h_2 to Alice.
(5) Alice computes k≡ h_2^a=$\left(g^b\right)^a = g^{ba}$(mod p).

(6) Bob computes k≡ h_1^a=$(g^a)^b = g^{ab}$ (mod p).
So, **k** becomes the secret key that both Alice and Bob can agree upon.

EL-GAMAL CRYPTOSYSTEM

In 1984, Tahar El-Gamal designed a public-key cryptosystem whose security is closely related to the difficulty of DLP.
The setup is given by:

(1) Alice chooses a prime p and a primitive root g (mod p).

(2) Alice chooses an n, 2≤n≤p-1, calculates h≡ g^x(mod p).

ENCRYPTION:

(1) Bob chooses a random number y, 2≤y≤p-1 and calculates r≡ g^y(mod p).

(2) Bob takes his message m and calculates c≡ $m.h^y$(mod p).

DECRYPTION:

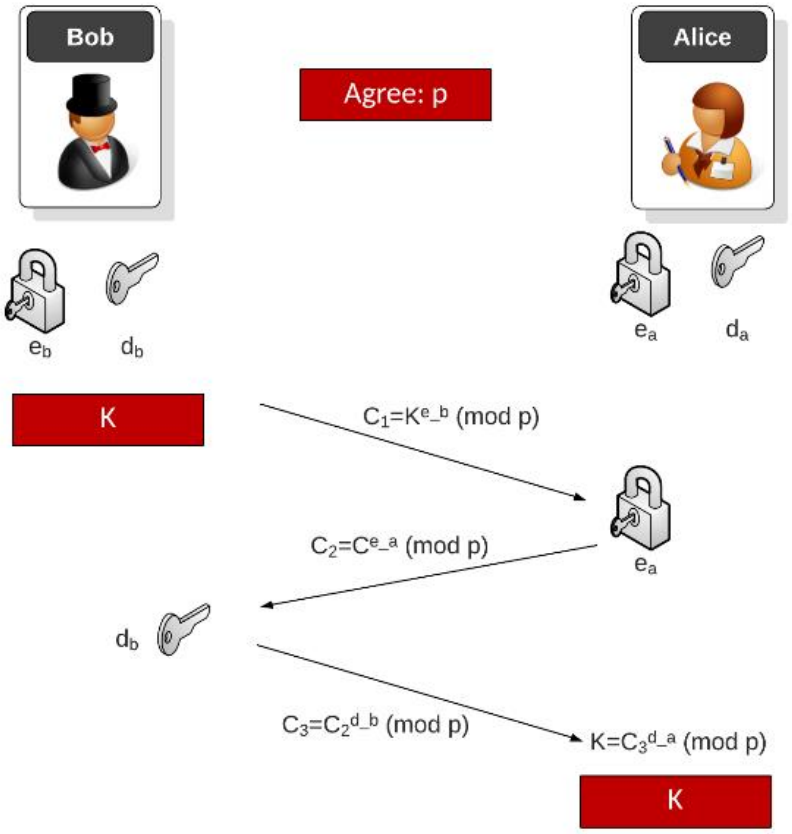
Alice computes m≡ $c.r^{-n}$(mod p) and sends r and c to Alice.

The above method works, because $c.r^{-n} \equiv (m.h^y).(g^y)^{-n} \equiv g^{ny}.m.g^{-ny} \equiv m(mod\ p)$.

If anyone else knew x, they could decrypt the message, but that depends on solving the DLP: x≡$log_g h$(mod p).

MASSEY-OMURA ENCRYPTION

Let us assume that Alice wants to send a message “m” to Bob, where m is the numerical equivalent of the message she is going to send, such that m∈ $\mathbb{Z}_{p^n}^*$, where p is a prime and n is a natural number.



The algorithm is given by:

(1) Alice and Bob choose e_A and e_B respectively, where e_A and e_B lies between 2 and $p^n - 1$, and gcd ($e_A, p^n - 1$), gcd($e_B, p^n - 1$)=1.

(2) Alice and Bob compute d_A=e_A^{-1}(mod p^n −1), d_B=e_B^{-1}(mod p^n −1) by using Extended Euclidean Algorithm.

(3) Alice calculates m^{e_A} and sends it to Bob.

(4) Bob calculates $m^{e_A e_B}$ and sends it to Alice.

(5) Alice calculates $m^{e_A e_B d_A}$=m^{e_B} and sends to Bob.

(6) Bob computes $m^{e_B d_B}$=m.

So, Bob successfully obtains “m” from Alice.

ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curves

General equation of an Elliptic curve E is:

$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_5$, where a_1,a_2,a_3,a_4,a_5 are constants.

$$=>\left(y+\frac{a_1x}{2}+\frac{a_3}{2}\right)^2=x^3+\left(a_2+\frac{a_1^2}{2}\right)x^2+\left(a_2+\frac{a_1a_3}{2}\right)x+\frac{a_3^2}{y}+a_5.$$

$=> y_1^2=x^3+\alpha x^2+\beta x+\gamma$,

Where $y_1=y+\frac{a_1x}{2}+\frac{a_3}{2}$, $\alpha=a_2+\frac{a_4}{4}$: $\beta=a_4+\frac{a_1a_3}{2}$, $\gamma=\frac{a_3^2}{4}+a_5$.

Or, $y^2=x^3+Ax+B$, which is the most common form, where $A, B \in K$, an arbitrary field.

In general, this field K can be the set of all real numbers or complex numbers, but in the perspective of Cryptography, we take this field as Zp, for some prime p.

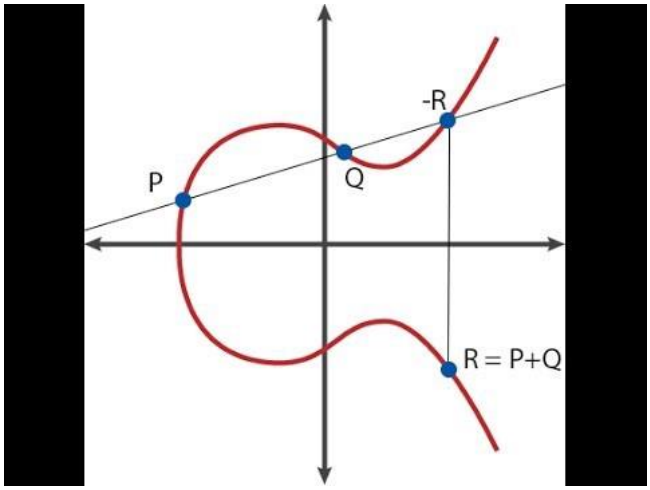
If we want to draw an elliptic curve whose points lie in a field L, we call the image of L as E(L), defined as:

$E(L)=\{\infty\}\cup\{(x,y)\in L\times L: y^2=x^3+Ax+B\}$, where A and B are constants.

We have to add the point infinity to the curve in the same way we add infinity to the set of all complex numbers to obtain c_∞ for stereographic projection of the Riemann sphere.

Group Law

Every symbol isn't always what it appears to be. That must be the most obvious real-life interpretation of cryptography ever. Similarly, "adding" two points on an arbitrary curve is not addition in the conventional sense. Let us explore the concept of this new type of addition:



We take two points on the elliptic curve E, name them P and Q. We draw a line segment joining P and Q and extend it beyond both P and Q until the line cuts E at another point. We call this new point -R. Now, we take reflection of this point about x-axis, i.e., we replace the ordinate of the point by its additive inverse. We call this R. So, R=P+Q.

So, to sum it up, let E: $y^2=x^3+Ax+B$ be an elliptic curve and P_1, P_2 be two points on E, where $P_1=(x_1, y_1)$, $P_2= (x_2, y_2)$. Addition of two points is defined as:

If $x_1 \neq x_2$,

Then slope of line joining these points $m=\frac{y_2-y_1}{x_2-x_1}$.

Then, $x=m^2-x_1-x_2$, $y=m(x_1-x)+y_1$.

If $x_1=x_2$, $y_1 \neq y_2$,

Then $P_1+P_2=\infty$.

If $x_1=x_2$, $y_1=y_2$, $y_1 \neq 0$, i.e., $P_1=P_2$.

Then, $m=\frac{3x_1^2+A}{2y_1}$.

$x=m^2-x_1-x_2$,

$y=m(x_1-x)- y_1$.

If $P_1=P_2$ and $y_1=0$,

then $P_1+P_2=\infty$.
And finally, $P_1+\infty=P_1 \ \forall P_1 \in E$.

As we discussed before, for its implementation in Cryptography, we usually take our field to be $\mathbb{Z}_p$ or $\mathbb{Z}_q, q = p^n$ for some prime p.

Hasse's Theorem

Let E be an elliptic curve over a finite field $\mathbb{Z}_q, q = p^n$. Then,

$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{Z}_q) \leq q + 1 + 2\sqrt{q}$.

Integer times a point

Let k be an integer and P be a point on an elliptic curve.

kP is given by:

(1) We take a=k, B=∞, C=P.

(2) If a is even, a=$\frac{a}{2}$, B=B, C=2C.

(3) If a is odd, a=a-1, B=B+C, C=C.

(4) If a≠0, we go to step 2.

(5) We get kP.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Let E be a given elliptic curve and P be a primitive element of E, and Q be some arbitrary point on E. The Elliptic Curve Discrete Logarithm Problem is to find an integer k, where 1≤k≤ #E, such that P+P+…+P(k times)=kP=Q.
There are various ways to tackle the ECDLP problem. We mention some of them here:

Baby Step- Giant Step

The algorithm is:

(1) Find s =$\lfloor\sqrt{n}\rfloor$, where ⌊ ⌋ represents the greatest integer function and find sP.

(2) Baby step: For i=0, 1,…s-1, find iP(mod n).

(3) Giant step: For j=1, 2,…s, find Q-jsP.(mod n).

(4) Search the lists to find i and j such that iP=Q-jsP. If so, then Q=(i+js)P=kP.

Now we add an example to explain the method better:

Let G=E(z_{599}), where E is given by $y^2=x^3+1$. We take P= (60,19), Q = (277,239).

So, we have A=0, B=1, n=599.

We calculate s=$\lfloor\sqrt{n}\rfloor\approx 25$.

We now calculate iP, for i=1, 2,. . . . 25.

We get: (60,19), (305,527), (329,543), (340,353), (32,254), (77,359), (263,114), (40,551), (344,587), (199,302), (562,506), (97,457), (492,339), (4,444), (87,218), (24,446), (279,532), (455,65), (62,463), (44,538), (520,77), (409,229), (68,565), (75,305), (351,183).

Now, we calculate Q-jsP, where j=1,2,…25.

We get: (403,91), (221,269), (500,185), (569,127), …. And we stop the calculation at j= 10 where Q-jsP=(24,446), which matches the value of iP for i=16.

Thus, we get 16P=Q-10.25P.

=>Q=16P+250P=266P.

So, we get k=266.

Pollard's ρ method

This is another method for calculating discrete logarithms, the problem being to solve for x such that xP=Q, where P and Q are points on some elliptic curve E over some field of order p.

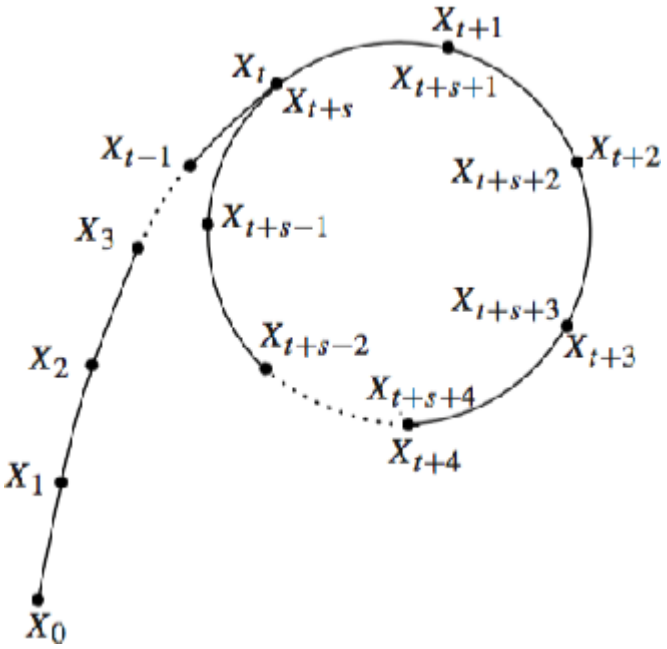
Here, the problem is to find integers a,b,A,B such that aP+bQ= AP+BQ.

Once the integers are found, we write:

(a-A)P=(B-b)Q.

So, (a-A)P=(B-b)xP.

Eliminating P, we get x= (a-A) ×(B-b)$^{-1}$(mod p).



Pollard's ρ method

CONCLUSION

Security benefits of Public-Key Cryptography are:

(1) Confidentiality- As the content is encrypted with an individual's public key, it can only be decrypted with the individual's private key, ensuring only the intended recipient can decrypt and view the contents.
(2) Integrity- Part of the decryption process involves verifying that the contents of the original encrypted message and the new decrypted match, so even the slightest change to the original content would cause the decryption to fail.

Comparison between ECC and RSA

The biggest differentiator between ECC and RSA is key size compared to cryptographic strength.

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes

It can be deduced from the chart that ECC is able to provide the same cryptographic strength as an RSA-based system with much smaller key sizes. For example, a 256-bit ECC key is equivalent to an RSA key of size 3072 bytes. The latest, most secure symmetric algorithms used by TLS use at least 128-bit keys, so it makes sense that the asymmetric keys provide at least this level of security.

BIBLIOGRAPHY

1) RSA and Public-Key Cryptography by Richard A. Mollin
2) Course in Number Theory and Cryptography by Neal Koblitz
3) An Introduction to Number Theory with Cryptography by James S. Kraft and Lawrence C. Washington
4) Elliptic Curves Number Theory and Cryptography by Lawrence C. Washington.