Elliptic Curve Cryptography

Professors: Manish Gupta

Group Members and their Contribution

1. **Soham Viradiya(202101472)**
   Contribution:

2. **Jay Sabva(202101224)**
   Contribution:

3. **Shrut Kalathiya(202101479)**
   Contribution:

4. **Vasu Golakiya(202101487)**
   Contribution:

5. **Vandit Bhalani(202101478)**
   Contribution:

6. **Dev Changela(202101069)**
   Contribution:

7. **Swet Lakhani(202101218)**
   Contribution:

1 Introduction and History of Cryptography

The History of cryptography can be split into two eras:the classical era and the modern era. The turning point between the two occurred in 1977, when both the RSA algorithm and the Diffie-Hellman key exchange algorithms were introduced.These new algorithms were revolutionary because they represented the first viable cryptographic schemes where security was based on the theory of numbers; it was the first to enable secure communication between two parties without a shared secret.Cryptography went from being about securely transporting secret codebooks around the world to being able to have provably secure communication between any two parties without worrying about someone listening in on the key exchange.

Modern cryptography is founded on the idea that the key that you use to encrypt your data can be made public while the key that is used to to decrypt your data can be kept private. As such, these systems are known as public key cryptographic systems. The first, and still most widely used of these systems, is known as RSA — named after the initials of the three men who first publicly described the algorithm: Ron Rivest, Adi Shamir and Leonard Adleman.

What you need for a public key cryptographic system to work is a set of algorithms that is easy to process in one direction, but difficult to undo. In the case of RSA, the easy algorithm multiplies two prime numbers. If multiplication is the easy algorithm, its difficult pair algorithm is factoring the product of the multiplication into its two component primes. Algorithms that have this characteristic — easy in one direction, hard the other — are known as Trap door Functions. Finding a good Trapdoor Function is critical to making a secure public key cryptographic system. Simplistically: the bigger the spread between the difficulty of going one direction in a Trapdoor Function and going the other, the more secure a cryptographic system based on it will be.

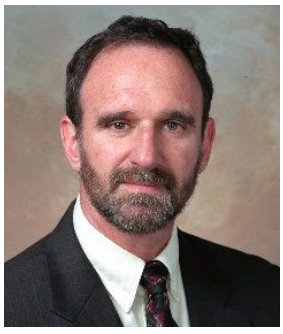


Figure 1: Martin Hellman

Figure 2: Rivet-Shamir-Adleman

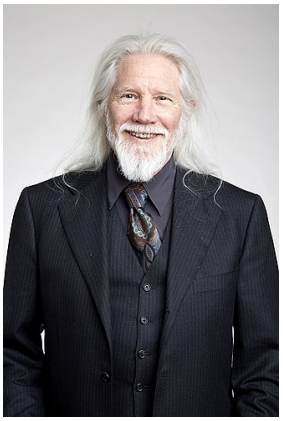


Figure 3: Whitefield Diffle

2 RSA algorithm

RSA(Rivet-Shamir-Adleman) is a public-key cryptosystem that is widley used for secure data transmission.The acronym "RSA" comes from the surname of Ron Rivest,Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977.In a public-key cryptosystem, the encryption key is public and distinct from the decryption key, which is kept secret(Private).An RSA user creates and publishes a public key based on two large prime number,along with an auxiliary value.The prime numbers are kept secret.Messages can be encrypted by anyone, via the public key,but can only be decoded by someone who knows the prime numbers. The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem".Breaking RSA encryption is know as the RSA problem.Whether it is as difficult as the factoring problem is an open question.There are no published methods to defeat the system if a large enough key is shared.RSA is a relatively slow algorithm.

Let's make this more concrete with an example. Take the prime numbers 13 and 7,their product gives us our maximum value of 91.Let's take our public encryption key to be the number 5. Then using the fact that we know 7 and 13 are the factors of 91 and applying an algorithm called the Extended Euclidean Algorithm, we get that the private key is the number 29

$$\boxed{public\,key\,(e)\,:5 \quad private\,key\,(d)\,:29 \quad maximum\,key\,(n)\,:91}$$

Let's use these values to encrypt the values to encrypt the message "CLOUD". In order to represent a message mathematically we have to turn the letters into numbers.A common representation of the Latin alphabet is UTF-8.Each character corresponds to a number.

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

Under this encoding,CLOUD is 67,76,79,85,68.Each of these digits are smaller than our maximum of 91,so we can encrypt them individually.Let's start with the first letter. For C=67 we have to multiply it by itself 5(public key) times to get the encrypted value.

$$\boxed{Encrypted\,Data\,E\,=A^e*(modn)} \qquad \boxed{Decrypted\,Data\,D\,=L^d*(modn)}$$

$$Encrypted Data C = 67^5 mod 91 = 58 \qquad Decrypted Data C = 58^{29} mod 91 = 67$$
$$Encrypted Data L = 76^5 mod 91 = 20 \qquad Decrypted Data L = 20^{29} mod 91 = 76$$
$$Encrypted Data O = 79^5 mod 91 = 53 \qquad Decrypted Data O = 53^{29} mod 91 = 79$$
$$Encrypted Data U = 85^5 mod 91 = 50 \qquad Decrypted Data U = 50^{29} mod 91 = 85$$

$EncryptedDataD = 68^5 mod 91 = 87 \qquad DecryptedDataD = 87^{29} mod 91 = 68$

So we can see that there no symmetric relation in keys so it will be harder to decrypt.

2.1 Encryption

Suppose alice wants to send a secret message to her friend Bob.The message is 'C'. How can she confidentially deliver this message,assuming that her sworn enemie Eve,might peek at its contents while it is in transit?

Example: The "Lock"(AKA the public key):(5,14) This pair of numbers is public,and is used by Alice(the sender) to encrypt messages. The "Key"(AKA the private key):(17,14) The first number in this pair of numbers is private,i.e. only known by Bob(the receiver).This pair is used to decrypt messages.

1. Alice encrypts the message "C" using (5,14).

   a.First,"C" is converted into an integer.In ASCII,"C" maps to the integer 67.But for simplicity,let's say it maps to 3 insted.

   b.Then,3 is encrypted $3^5 mod(14) = 5$

2. Alice sends the result,5,to Bob.if Eve peek and see this message,it tells them nothing.

3. Bob decrypts the result using (17,14).

   a.$5^17 mod(14) = 3$

   b.The result is 3.Bob maps 3 back to a character,which is "C",to get the original message.

2.1.1 Lock Key Generation

In our example,Bob genrates these two points of number.He gives everyone access to the firs pair of numbers(5,14),but keeps the second pair secret(specifically the first part 17,the second part 14 is in the first pair and thus is public) so only he can decrypt messages.

1. Bob Choose two prime numbers,p and q.In real life scenarios,they should be large(for security),but we"ll chooe small numbers to make things easier.

$$\boxed{p = 2 \quad q = 7}$$

2. Compute

$$\boxed{n = p * q}$$

3. Compute e(for encryption) such that:

$$\boxed{1 < e < (p-1)(q-1) \qquad and \qquad GCD(e,(p-1)(q-1) = 1}$$

4. Compute d (for decryption) such that:

$$\boxed{d = e^{-1} mod((p-1)(q-1))}$$

$$\boxed{d * 5 = 1 mod 6}$$

This means d can be 5,or 11,or 17,and so on.We"ll choose d=17. That's it,we're done

$$\boxed{The\ "Lock"\ : (e, n)\ =\ (5, 14)}$$

$$\boxed{The\ "Key"\ : (d, n)\ =\ (17, 14)}$$

As Shown before,senders can use the first pair of numbers to encrypt messages and the receiver can use the second pair of numbers to decrypt messages.

The basic principle underlying RSA is that for all integers 0 m ¡ n, it is practical to find positive integers e, d, and n such that: (m mod n) mod n = m and such that even if e and n are made public, it is extremely difficult to find out what d is.

The Takeaway is that you can take a number,multiply it by itself a number of times to get a random-looking number,then multiply that number by itself a secret number of times to get back to the original number. These factoring algorithms get more efficient as the size of the numbers being factored get larger.The gap between the difficulty of factoring large numbers and multiplying large numbers is shrinking as the number(The key's bit length) gets larger.As the resources available to decrypt numbers increase,the size of the keys need to grow even faster.This is not a sustainable situation for mobile and low-powered devices that have limited computational power.The gap between factoring and multiplying is not sustainable in the long term.

2.2 Issues with RSA

As of 2005, the largest number factored using general purpose methods is 663-bits long, using state of the art distributed methods. Experts feel that 1024-bit keys may become breakable in the near future(though disputed). 256-bit length keys are breakable in a few hours using a personal computer. The current recommended key-length is 2048-bits[wik]. Though this length may be insignificant for most personal computers in use, it causes low processing power portable devices like smartcards to become ine cient. There are constraints on processor word length, available memory and clock speeds in these devices. As the need for portable and secure identification slowly becomes a necessity, and as RSA key sizes will increase in proportion to the processor power available, there arises a need to devise a scheme which provied the same level of cryptographic security with smaller key lengths. ECC is one such scheme, described in the following section

All this means is that RSA is not the ideal system for the system for the future of cryptography.In an ideal Trapdoor Function,the easy way and the hard way get harder at the same rate with respect to the numbers in question

3 Diffie-Hellman key exchange

3.1 Introduction

Whitefield and Martin Hellman developed Diffie-Hellman key exchange Algorithms in 1976 to overcome the problem of key agreement and exchange. It was one of the first public-key exchange protocols as conceived by Ralph Merkle and named after Whitefield Diffe and Martin Hellman. It enables the two parties who want to communicate with each other to agree on a symmetric key(a key that can only be used for encrypting and decrypting).

This Algorithm can only be used for Key Exchange and not for encryption and decryption process.The Algorithm is based on mathematical principles.

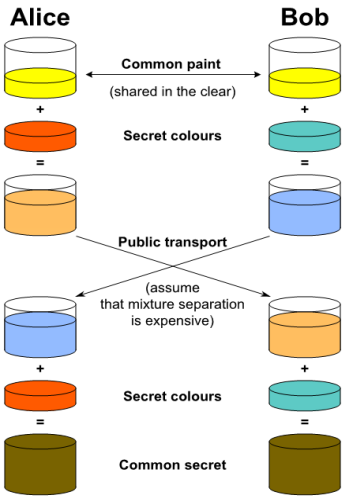
3.2 Reason to use Algorithm

When we are sending a key to receiver then it can be attacked in between(i.e. Insecurity may occurs). It allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel.

3.3 General Overview of an algorithm

Alice and Bob,publicly agree on an arbitrary starting color that does not need to be kept secret(but should be different every time).In this example,color is yellow.Each person also selects a secret color that they keep to themselves-in this case,red and cyan.The crucial part of the process is that Alice and Bob each mix their own secret color together with their mutually shared color,resulting in orange-tan and light-blue mixtures respectively, and then publicly exchange the two mixed colors.Finally,each of them mixes the color they received from the partner with their own private color.The result is a final color mixture(yellow-brown in this case) that is identical to their partner's final color mixture. If a third party listened to the exchange,it would only know the common color(yellow) and the first mixed colours(orange-tan and light-blue),but

Figure 4: Illustration of Diffie-Hellman key exchange



it would be difficult for this party to determine
the final secret color(yellow-brown).

3.4 Mathematical Explanation

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo p,where p is prime,and g is a primitive root modulo p.These two values are choosen in this way to ensure that the resulting shared secret can take on any value from 1 to$p-1$.Here is an example of the protocal,with non-secret values in blue,and secret values in red

1. Here Alice and Bob wants to share there secret but Eve is keeping watch on Them So Alice and Bob publicly agree to use a modulus p=23 and base $g=5$(primitive root of modulo 23).
2. Alice choose a secret integer $a=4$,then sends Bob $A=g^a \bmod p$
   - $A=5^4 \bmod 23$=4
3. Bob choose a secret integer $b=3$,then sends Alice $B=g^b \bmod p$
   - $B=5^3 \bmod 23$=10
4. Alice computes s= $B^a \bmod p$
   - s= $10^4 \bmod 23$=18
5. Bob computes s= $A^b \bmod p$
   - s= $4^3 \bmod 23$=18
6. Alice and Bob now share a secret

Both Alice and Bob arrived at the same values because under mod p,

$$\boxed{A^b \bmod p = g^{ab} \bmod p = g^{ba} \bmod p = B^a \bmod p}$$

More specifically

$$\boxed{(g^a \bmod p)^b \ mod \ p \ = \ (g^b \bmod p)^a \ mod \ p}$$

Only a and b are kept secret.All the other values – p, g, $g^a \bmod p$, and $g^b \bmod p$ – are sent in the clear. The strength of the scheme comes from the fact that $g^{ab} \bmod p = g^{ba} \bmod p$ take extremely long times to compute by any known algorithm just from the knowledge of p, g, $g^a \bmod p$, and $g^b \bmod p$. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel.

Of course, much larger values of a, b, and p would be needed to make this example secure, since there are only 23 possible results of n mod 23. However, if p is a prime of at least 600 digits, then even the fastest modern computers using the fastest known algorithm cannot find a given only g, p and $g^a \bmod p$.

Such a problem is called the **discrete logarithm problem**.The computation of g^a mod p is known as modular exponentiation and can be done efficiently even for large numbers. Note that g need not be large at all, and in practice is usually a small integer (like 2, 3, ...)

3.4.1 Secrecy Table

Alice	
Known	Unknown
p = 23	
$\alpha = 9$	
x = 4	y
a = 9^4(mod 23) = 6	
ka=16^4(mod 23) = 9	

Bob	
Known	Unknown
p = 23	
$\alpha = 9$	
y = 3	x
b= 9^3(mod 23) = 9	
kb=6^3(mod 23) = 9	

Eve	
Known	Unknown
p = 23	
$\alpha = 9$	
	x,y
a,b	
	ka,kb

After the introduction of RSA and Diffie-Hellman, researchers explored other mathematics-based cryptographic solutions looking for other algorithms beyond factoring that would serve as good Trapdoor Functions. In 1985, cryptographic algorithms were proposed based on an esoteric branch of mathematics called elliptic curves.

4 Elliptic curve Cryptography

elliptic-curve cryptography is a approach to public key cryptography based on the algebraic stricture of elliptic curves over finite fields.ECC allows smaller keys compared to non-EC cryptography(based on plain Galois Fields) to provide equivalent security.The technique was first proposed individually by Neal Koblitz and Victor Miller in 1985. The ECC is based on the *Elliptic Curve Discrete Logarithm* Problem,which is known NP-Hard problem. Elliptic curve cryptography algorithms entered wide use in 2004 to 2005. An elliptic curve is defined by the equation

$$\boxed{y^2 = x^3 + ax + b}$$

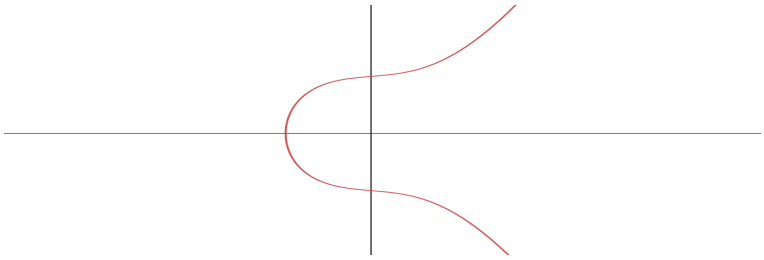


Figure 5: Elliptic Curve $y^2 = x^3 + ax + b$

Elliptic curves cryptography uses elliptic curves over the fintie field Z_p(where p is prime and $p > 3$) or Z_{2m} (where the fields size p=2m).This means that the field is a square matrix of size $p * p$ and the point on the curve is limited to integer coordinates within the field only.All algebric operation within the field (like point multiplication and addition) result in another point within the field.

The Elliptic curve equation over the finite field Zp

$$\boxed{y^2 = x^3 + ax + b(modp)}$$

The Bitcoin curve(secp256k1) equation over the finite field Zp

$$\boxed{y^2 = x^3 + 7(modp)}$$

Unlike RSA,which uses for its key space the integers in the range[0...p-1](the field Zp),the ECC uses the points{x,y} within the Galois field Zp(where x and y are integers in the range[0...p-1]

4.1 Mathematics

Given a curve,E,defined along some equation in a finite field(such as E:$y^2 = x^3 + ax + b$,point multiplicatioj is defined as the repeated addition of a point along that curve.denote as $nP = P + P + P... + P$ for some scalar n and a point $P = (x, y)$ that lies on the curve,E.

4.1.1 Point Operations

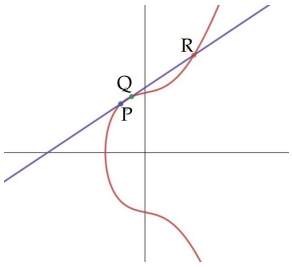


Figure 6: $P+Q+R=0$

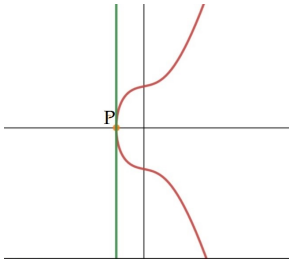


Figure 7: $P+P+0=0$

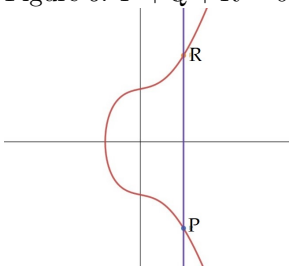


Figure 8: $P+R+0=0$

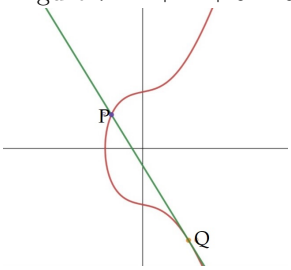


Figure 9: $P+Q+Q=0$

Point at Infinity
Point at infinity $\mathcal{O}$ is the identity element of elliptic curve arithmetic.Adding it to ant point result in that other point,including adding point at infinity to itself.That is

$$\boxed{\mathcal{O}+\mathcal{O}=\mathcal{O}}$$

$$\boxed{\mathcal{O}+\mathcal{P}=\mathcal{P}}$$

Point negation
Point negation is finding such a point,that adding it to itself will result in point at infinity($\mathcal{O}$).

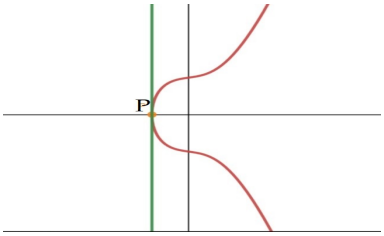


Figure 10: Point negation $P+P+0=0$

$$\boxed{\mathcal{P} + (-\mathcal{P}) = \mathcal{O}}$$

For elliptic curves that is a point with same x coordinate but negated y coordinate:

$$\boxed{(x, y) + (-(x, y)) = \mathcal{O}} \quad \boxed{(x, y) + (x, -y)) = \mathcal{O}} \quad \boxed{(x, -y) = -(x, y)}$$

Point addition
Given Two point in the set $E = (x, y) \mid y^2 + ax + b \cup \mathcal{O}$

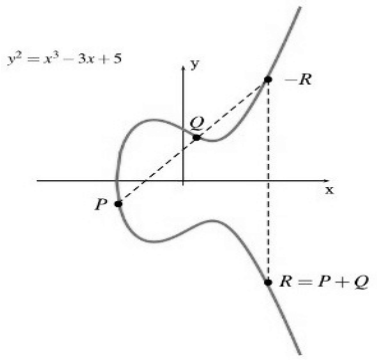


Figure 11: Point addition $P + Q = R$

With 2 distinct points,P and Q,addition is defined as the negation of the point resulting from the intersection of the curve,E,and the straight line defined by the points P and Q giving the point,R

$$\boxed{\mathrm{P} + \mathrm{Q} = \mathrm{R}}$$

Assuming the elliptic curve,E, is given by $y^2 = x^3 + ax + b$, so the slope of line (PQ) λ can be calculated as:

$$\boxed{\lambda = \frac{y_q - y_p}{x_q - x_p}}$$

$$\boxed{x_r = \lambda^2 - x_p - x_q} \quad \boxed{y_r = \lambda((x_p - x_r) - y_p}$$

These equations are correct when neither point is the point at infinity,O,and if the points have diffrent x coordinates(they're not mutual inverses).This is important for the Elliptic curve digital singature algorithm(ESDSA) where the hash value could be zero.

Point doubling
Where the point P and Q are coincident(at same coordinate),addition is similar,except that there is no well-defined straight line through P,so the operation is closed using limiting case,the tangent to the curve,E,at P.

$$\boxed{\lambda = \frac{3x_p^2 + a}{2y_p}}$$

where a is from the defining equation of the curve,E,above.

$$\boxed{x_r = \lambda^2 - 2x_p} \quad \boxed{y_r = \lambda((x_p - x_r) - y_p}$$

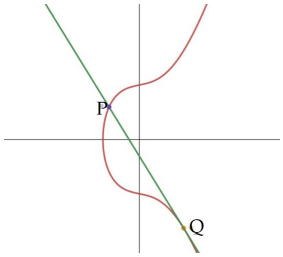


Figure 12: Point doubling $P + Q + Q = 0$

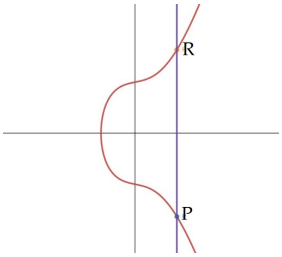


Figure 13: Point doubling $P + R + 0 = 0$

5 Commercialization

Permutations are frequently used in communication networks and parallel and distributed systems . Routing different permutations on a network for performance evaluation is a common problem in these fields. Many communication networks require secure transfer of information, which drives development in cryptography and network security. This area has recently become particularly significant because of the increased use of internet information transfers. Associated problems include protecting the privacy of transactions and other confidential data transfers and preserving the network security from attacks by viruses and hackers. Encryption process involves manipulations of sequences of codes such as digits, characters, and words. Hence, they are closely related to combinatorics, possibly with intelligent encryption process. For example, one common type of encryption process is interchanging–i.e., permuting parts of a sequence . Permutations of fast Fourier transforms are employed in speech encryption .

Queries in databases are multiple join operations that are permutations of the constituent join operations. Determining an optimal permutation that gives minimum cost is a common and important problem. Data mining or knowledge discovery in databases is a relatively new field that aims at distilling useful information, often from large databases. In this area, techniques employing symbolic AI can manipulate combinatorial sequences of atoms or information elements. Non-symbolic knowledge discovery techniques, such as genetic algorithms and genetic programming, most commonly deal with solutions in the form of sequences of bits, digits, characters, and sometimes Lisp program elements. Neural networks, another domain of non-symbolic AI, sometimes deal with combinatorial patterns. Knowledge discovery techniques under uncertainty, such as Bayesian networks, Dempster-Shafer theory, fuzzy logic, and rough set theory, may have combinatorial solutions.

Some other applications like Computer architecture, Computational molecular biology, Languages, Pattern analysis, Scientific discovery, Operations research, Simulation, Homeland security .

6 Evolution of Cryptography

Year	Event
1900 BC	Around 4000 years ago, the Eqyptians used to communicate by messages written in **hieroglyph**.
300 BC	"Arthshashtra",a classic work on statecraft written by Kautalya,describes the espionage service in India and mentions giving assignment to spies in "secret writing"
Wednesday	Rain will still linger for the morning. Conditions will improve by early afternoon and continue throughout the evening.

Symmetric Key Size(bits)	RSA-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