To what extent do modern cryptographic methods effectively secure online communications, transactions, and the internet?

Extended Essay
Mathematics Higher Level

Contents

1 An introduction into modern cryptography

Throughout history, beginning from the days of the Caesar cipher, individuals have wanted to exchange secret messages. To encrypt and decrypt these messages there has always been a secret **key** that both ends needed to have. This is called symmetric cryptography, wherein both parties, the sender and the receiver, have to have the key. This is very dangerous as if a malicious user got a hold of the key. They could decrypt the message very easily.

Hence, for the purposes of modern cryptography, messages are exchanged without the risk of such a key. This negates the risk associated with a stolen key, making it near impossible for malicious users to decipher messages.

This system is called public key cryptography. In it, there are two keys. One is public, everyone has it. The other one is private. The encrypted message is secured using a *public key*, and can only be decrypted using a that same user's *private key.*

1.1 Public Key Cryptography: A high level overview

Let us define two individuals who want to send each other a secret parcel. Let their names be Alice and Bob wherein Alice is the sender and Bob is the receiver.

1. First Bob sends an unlocked padlock to Alice (Bob would send that to anyone even someone he doesn't really trust). This is the public key. The only use of an unlocked padlock is to send Bob a parcel since Bob is the only one who has the key that can open the padlock.
2. Alice locks up the package she wants to send with the padlock Bob sent her. Only Bob can open the package now.
3. After receiving the page, Bob can open it with his private key.

This simple exchange is called the Public-Key Exchange and makes a foundation for Public-Key Cryptography.

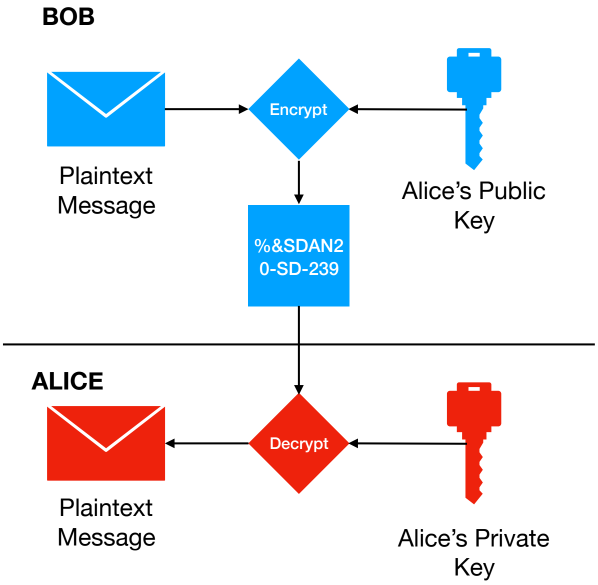


Figure 1: Public Key Exchange Visualization

1.2 Assumptions in the Public-Key Exchange

If we need to generate these keys we need to assume a few things that will be explored:

1. Generating large prime numbers of a particular bit-size, that is a particular number of digits, is easy.
2. It is easy to multiply two primes p and q and find their product $n = p \times q$
3. Given a product of primes n, it very difficult to recover the prime factors p and q.
4. Given $\phi(n)$, the $Message$, and e, it's easy to compute $C = (M \cdot e) \pmod{\phi(n)}$. Formally this is called modular exponentiation and can be be represented like,

$$Encryption(Message) = Ciphertext \tag{1.1}$$

5. With $\phi(n)$, e, C, and the prime factors p and q, it is easy to recover the value M such that $C = (M \cdot e) \pmod{\phi(n)}$. Formally this is the reverse of modular exponentiation – Modular root extraction – is easy given the decryption key,

$$Decryption(Encryption(Message)) = Message \tag{1.2}$$

6. For all other cases modular root extraction is difficult. Given just n, e, and c, but **not** the prime factors p, q, it is very difficult to recover the value of the $Message$.

The significance of these operations, to tackle the assumptions, will be explained and explored later in the essay.

2 Divisibility, modular arithmetic, and number theory

2.1 Prime Numbers and factorizing

Prime numbers, how difficult they are to find, and factor, make the basis of public key cryptography. A prime number $p \in \mathbb{Z}^+$ that cannot be formed by multiplying two numbers together with its only factors being 1 and itself.

Theorem 1 (Euclid's Theorem)**.** *The set of all prime numbers is infinite*[1].

Consider a finite list of all the primes $p_1 = 2 < p_2 = 3 < p_3 = 5 < ... < p_n$. For this list, let P be the total product of all the primes in this list, i.e., $P = p_1 \cdot p_2 \cdot p_3 \cdot ... \cdot p_n$. Consider $(P + 1)$,

1. $(P + 1)$ is a prime: there is at minimum one more prime.
2. $(P + 1)$ is not prime: there exists some prime p that divides $(P + 1)$. Also if p were on the list it would also divide P. Hence, it would also have to divide the difference $(P + 1) - P = 1$. Since no prime divided 1, p cannot be on the list since primes on the list divide P, so there exists one additional prime p.

Both cases show that it is impossible to create a finite list of primes.

Definition 1. Greatest Common Divisor ($gcd()$): The $gcd(a, b)$ where $a, b \neq 0$ is the largest possible integer that divides both of the integers a and b[2].

Definition 2. Coprime: Two integers, a and b are said to be coprime or relatively prime if their gcd is 1, that is, they have no common factors besides 1[2].

[1] Euclid, and Thomas Little Heath. *The Thirteen Books of Euclid's Elements.* Vol. 2, Cambridge University Press, 2015.

[2] Fannon, Paul et al. "2B: Greatest Common Divisor and Least Common Multiple." *Mathematics Higher Level for the IB Diploma Option Topic 10 Discrete Mathematics.* Cambridge University Press, 2013, pp. 19-23.

2.2 Modular Arithmetic

In modular arithmetic, the modulo function gives the remainder upon division by another number. Two numbers a and b are said to be congruent or equivalent if they give the same remainder when divided by a number n. Another way to say it would be that they are congruent $\mod n$ if their difference $(a-b)$ is an integer multiple of n, i.e., $\frac{(a-b)}{n} = k \in \mathbb{Z}$. For example, 15 and -9 give the same remainder when divided by 12, therefore,

$$a \equiv b \pmod{n} \iff m|a-b \tag{2.1}$$

This could be said to be a linear congruence if $a = qm + r$ and $b = lm + r$, that is,

$$a \equiv b \pmod{n} \iff \exists\, l, q, r \in \mathbb{Z} : a = qm + r \textit{ and } b = lm + r \tag{2.2}$$

Similarly, other rules for modular arithmetic[3] are as expected, if $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$, then:

- $ka \equiv kb \pmod{n}$ for all $k \in \mathbb{Z}$
- $a + c \equiv b + d \pmod{n}$
- $a - c \equiv b - d \pmod{n}$
- $ac \equiv bc \pmod{n}$
- $a^m = b^m \pmod{n}$

Division, however, is different[4]. Suppose we want to make both sides of a congruence divisible by d. We can subtract or add a multiple of n from one side of a congruence. Suppose if $a \equiv b \pmod{n}$ then $a \equiv b \pm n \pmod{n}$. This will allow us to make both sides divisible by d. This brings us to the three rules of division.

[3] Fannon, Paul et al. "5B: Rules of Modular Arithmetic." *Mathematics Higher Level for the IB Diploma Option Topic 10 Discrete Mathematics.* Cambridge University Press, 2013, pp. 53-55.

[4] Fannon, Paul et al. "5C: Division and Linear Congruences." *Mathematics Higher Level for the IB Diploma Option Topic 10 Discrete Mathematics.* Cambridge University Press, 2013, pp. 56-59.

- Consider when $a \equiv b \pmod{n}$ and d divides both a, b, and $gcd(d, m) = 1$, then $\frac{a}{d} \equiv \frac{b}{d} \pmod{n}$. For example, if $5x \equiv 15 \pmod{24}$ then $x \equiv 3 \pmod{24}$ as 5 is coprime with 24.
- When d and m have some common factors we need to change the modulo when dividing. If $a \equiv b \pmod{n}$ and d divides a, b, n then, $\frac{a}{d} \equiv \frac{b}{d} \pmod{\frac{n}{d}}$
- If $a \equiv b \pmod{n}$ and d divides a, b then, $\frac{a}{d} \equiv \frac{b}{d} \pmod{\frac{n}{gcd(d,n)}}$

2.3 Fermat's Little Theorem

Theorem 2 (Fermat's Little Theorem)**.** *Let p be a prime number and a any integer. Then $a^p - a$ is always divisible by p. It can also be written in modular arithmetic notation as*[5]*:*

$$a^p = p \pmod{p} \textbf{ or } a^{p-1} = 1 \pmod{p}$$

Consider the following.

$$\text{If, } a \equiv 0 \pmod{p}$$
$$\text{then evidently, } a^p \equiv 0 \pmod{p}$$
$$\text{therefore, } a^p \equiv a \pmod{p}$$

So it only remains to prove equation 2.3 where $a \not\equiv 0 \pmod{p}$, that is, a is not divisible by p.

Let there be a list of the first nonnegative numbers p multiplied by a,

$$0, 1, 2, 3, ..., p-3, p-2, p-1 \tag{2.3}$$

$$\Rightarrow 0, a, 2a, 3a, ..., (p-3)a, (p-2)a, (p-1)a \tag{2.4}$$

Reducing these new numbers $\pmod{p}$, we will get the original list, that is, we can use a property of the modulus function.

For the proof[6], let us take $a = 4$ and $p = 7$. This gives us the original list as $\{0, 1, 2, 3, 4, 5, 6\}$ and the new list as $\{0, 4, 8, 12, 16, 20, 24\}$. If we reduce this

[5] "Fermat's Little Theorem." *Brilliant Math & Science Wiki*, brilliant.org/wiki/fermats-little-theorem/.

[6] Burton, David M. *Elementary Number Theory.* 7th ed., McGraw-Hill, Higher Education, 2011.

list $\pmod 7$, we get $\{0, 4, 1, 5, 2, 6, 3\}$, evidently no two elements on the new list are equivalent $\pmod p$. It is also just the original list in a scrambled order.

From 2.4 we get, $0, a, 2a, 3a, ..., a(p-3), a(p-2), a(p-1)a$ reduced $\pmod p$ to a list of p so that each possible remainder $0, 1, 2, 3, ..., p-3, p-2, p-1$, must appear once, so it is in a scrambled order (the zero entities in this list can be disregarded and removed). The two lists have the same elements modulo p, that means they have the same products modulo p:

$$a \cdot 2a \cdot ... \cdot a(p-2) \cdot a(p-1) \equiv 1 \cdot 2 \cdot ... \cdot (p-2) \cdot (p-1) \pmod p \tag{2.5}$$

Which can be factorized to,

$$a^{p-1} \cdot 1 \cdot 2 \cdot ... \cdot (p-2) \cdot (p-1) \ \equiv 1 \times 2 \cdot ... \cdot (p-2) cdot (p-1) \pmod p \tag{2.6}$$

By subtracting,

$$a^{p-1} \cdot 1 \cdot 2 \cdot ... \cdot (p-2) \cdot (p-1) - 1 \cdot 2 \cdot ... \cdot (p-2) \cdot (p-1) \equiv 0 \pmod p \tag{2.7}$$

or,

$$(a^{p-1} - 1) \cdot 1 \cdot 2 \cdot ... \cdot (p-2) \cdot (p-1) \equiv 0 \pmod p \tag{2.8}$$

None of the factors $1, 2, ..., p-1$ is divisible by p since they are lesser than p. So we must have $a^{p-1} - 1$ divisible by p, which proves another form of equation 2.3 :

$$a^{p-1} - 1 = 0 \pmod p \tag{2.9}$$

2.4 Chinese Remainder Theorem

Theorem 3 (Chinese Remainder Theorem)**.** *If two numbers p and q are coprime, then the simultaneous linear congruencies $y \equiv a \pmod p$ and $y \equiv b \pmod q$ have a unique solution $n = \pmod{pq}$*[7].

For example, let us take the pair of equations,

$$\left.\begin{aligned} y &\equiv 2 \pmod 3 \\ y &\equiv 3 \pmod 5 \end{aligned}\right\}$$

[7] Fannon, Paul et al. "5D: Chinese Remainder Theorem." *Mathematics Higher Level for the IB Diploma Option Topic 10 Discrete Mathematics.* Cambridge University Press, 2013, pp. 59-62.

That gives us,

$$y \equiv 2 \pmod 3 \Rightarrow x = 2, 5, 8, 11, 14, 17, 20, 23, ...$$

$$y \equiv 3 \pmod 5 \Rightarrow x = 3, 8, 13, 18, 23, 28, ...$$

Doing this is a long process, and we only have 2 solutions, 8 and 23. In the first list, all numbers are +3, in the second list they are +5. So, to get number in both lists we need to +15. Therefore, all solutions are in the form $8 + 15k$, or $y \equiv 8 \pmod{15}$.

3 The RSA

RSA (Rivest-Shamir-Adleman)[8] is a very popular public-key cryptosystem used to data transmission on the internet and to secure sensitive transactions. Like introduced in section 1.1, it is an asymmetric cipher.

3.1 Euler's Theorem

The RSA relies on the Euler's totient function $\phi(n)$ for the computation of relatively prime numbers required for encryption and decryption.

Definition 3. Euler's totient function: $\phi(n)$ denotes the set of numbers $\leq n$ and which are relatively prime to n. In other words, $\phi(n)$ is the number of $m \in \mathbb{N}$ such that $1 \leq m < n$ and $gcd(m, n) = 1$[9].

For example, if we want to find $\phi(8) = 4$,

$$\begin{array}{ll} gcd(1,8) = 1 & gcd(5,8) = 1 \\ gcd(2,8) = 2 & gcd(6,8) = 2 \\ gcd(3,8) = 1 & gcd(7,8) = 1 \\ gcd(4,8) = 4 & gcd(8,8) = 8 \end{array}$$

From this we can derive a conclusion for a prime p as,

$$\phi(p) = p - 1 \tag{3.1}$$

Since all the integers $\mathbb{Z} < p$ are relatively prime to p. The figure 2 for this function till $n = 8000$ shows it clearly. There is an evident upper bound of the line $\phi(n) = n - 1$

[8] Rivest, R. L., et al. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, vol. 21, no. 2, 1 Jan. 1978, doi:10.21236/ada606588.

[9] Pettofrezzo, Anthony J., and Donald R. Byrkit. *Elements of Number Theory.* Prentice-Hall, 1970, p. 80.

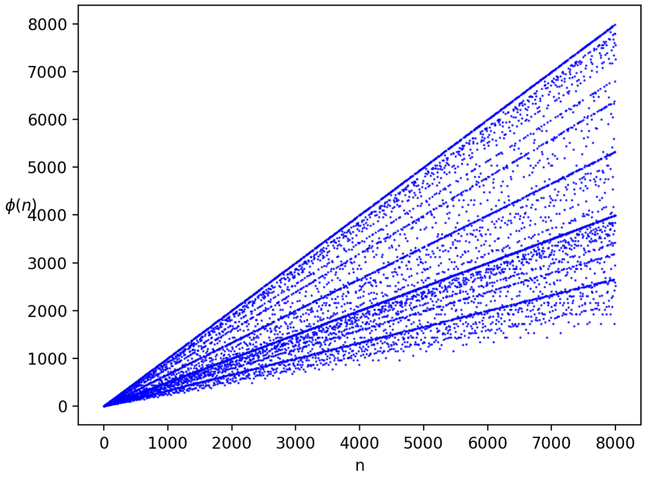


Figure 2: Graph of $\phi(n)$ for $n \leq 8000$

3.2 Encryption Process

The steps taken to encrypt a message using the RSA are as follows[10]:

1. Generate two large prime numbers p and q (this can be done using various methods such as Fermat Test or Miller-Rabin Test, but they are out of the scope of this essay).

2. Compute a modulus n such that:

$$n = p \cdot q \tag{3.2}$$

3. Compute $\phi(n)$ such that

$$\phi(n) = (p-1)(q-1) \tag{3.3}$$

4. Now, we can disregard p and q, such that we erase them from the system in question.

[10] Rivest, R. L., et al. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, vol. 21, no. 2, 1 Jan. 1978, doi:10.21236/ada606588.

5. Choose two numbers e and d such that, where e is the encryption key and is relatively prime to ϕ, i.e., $gcd(e, \phi(n)) = 1$, and d is the decryption key such that:

$$ed = 1 \pmod{\phi(n)} \tag{3.4}$$

   Therefore, it complies with Euler's Theorem where $\phi(n)$ is Euler's function which is the amount of numbers smaller than n that are coprime to it. This means that, $(p-1)(q-1)$ is coprime to n. The proof for the Little Theorem follows in the next page.

6. The couple (e, n) constitutes the public key, wherein n is called the modulus, and it signifies the number of digits the prime numbers are, and e the exponent.

7. The private key is d and may sometimes be written as (d, n).

8. Let the plaintext message be M such that $0 \leq M \leq (n-1)$. It is possible to convert a message to the decimal system using ASCII values (table in Appendix D).

9. M is then encrypted to ciphertext C according to the formula:

$$C = M^e \pmod{n} \tag{3.5}$$

3.3 Applying Fermat's little theorem to prove the RSA encryption and decryption

Fermat's Little Theorem as seen in equation 2.3 is,

$$a^p \equiv a \pmod{p} \tag{3.6}$$

Multiplying with a^{p-1},

$$a^{p-1} \times a^p \equiv a^{p-1} \times a \equiv a^p \equiv a \pmod{p} \tag{3.7}$$

Supposing we repeat this multiplication K times,

$$a^{K(p-1)} \times a^p \equiv a \pmod{p} \tag{3.8}$$

Regrouping considering $a^p = a^{p-1} \times a$,

$$\begin{aligned} a^{K(p-1)} \times a^{p-1} \times a &\equiv a \pmod{p} \\ \text{factoring } a^{p-1} \Rightarrow a^{(K+1)(p-1)} \times a &\equiv a \pmod{p} \\ \Rightarrow a^{(K+1)(p-1)+1} &\equiv a \pmod{p} \end{aligned} \tag{3.9}$$

Let $K+1=N$, since they are both constants,

$$a^{N(p-1)+1} \equiv a \pmod p \tag{3.10}$$

Which is true for any a and N

From equation 3.2, 3.3, and 3.4 we know that n is a product of two primes and e has no common factors with ϕ. Then we find the multiplicative inverse of e modulo ϕ, i.e. the number d, which is the decryption key. So, $ed = 1+$ a multiple of ϕ, or

$$ed = L(\phi) + 1 \tag{3.11}$$

From equation 3.5 we know that,

$$C = M^e \pmod n \tag{3.12}$$

The number C will be the encrypted text of M. To decrypt we compute $z \equiv C^e \pmod n \equiv M^{ed} \pmod n$. That means the plaintext z is the original message M again. To prove this, we can use equation 3.11 and 3.5, and taking $n = pq$,

$$M^{ed} = M^{L(\phi)+1} \equiv M \pmod p \tag{3.13}$$

$$\Rightarrow M^{ed} - M = \text{multiple of } p \tag{3.14}$$

and the same of q instead of p such that,

$$M^{ed} = M^{L(\phi)+1} \equiv M \pmod q \tag{3.15}$$

$$\Rightarrow M^{ed} - M = \text{multiple of } 1 \tag{3.16}$$

Therefore, $M^{ed} - M$ can only be a multiple of p and q if it is a multiple of its product n. Implying the equation mentioned above using the theorem 3 which is the Chinese Remainder Theorem mentioned in section 2.4,

$$M^{ed} \equiv M \pmod n \tag{3.17}$$

This equivalence only establishes the fact that M^{ed} and M have the same remainders when divided by n. Since M is positive and less than n, the remainder of dividing M^{ed} by n is always M. Therefore, to recover M, $C^e \pmod n$ must be computed.

The pair (e, n) make the public key. But, to decrypt you need the private key d and to compute,

$$C^d \pmod n \equiv M \tag{3.18}$$

d is the multiplicative inverse of e modulo ϕ. (e, n) is public knowledge, but ϕ is not, so d cannot be calculated. n is public knowledge though, but it is not factored into p, q. As stated, public key cryptography is safe due to the near impossible problem that is prime factorization of large prime numbers. Therefore since it is almost impossible to factor n into p and q, it is almost impossible to get ϕ which is $(p-1)(q-1)$, making RSA safe.

3.4 Example RSA encryption

Following the steps in section 3.2 we get,

1. $p = 7$ and $q = 11$
2. $n = 7 \times 11 = 77$
3. $\phi(77) = (7-1) \times (11-1) = 60$
4. Disregarding p and q...
5. We need to find a which satisfies the condition for in equation 3.3, that is, to a number with two prime factors ed, and is equivalent to 1 $\pmod{\phi(n)}$. The following are numbers which equal $1 \pmod{\phi(77)}$ or $1 \pmod{60}$, but all do not have factors, that must be tested for:

61	121	181	241	301
361	421	481	541	601
661	721	781	841	961
1021	1081	1141	1201	1261
1321	1381	1441	1501	1561
1621	1681	1741	1801	

   For the purpose of this example let us take 481 (although it does not matter), therefore, $e = 37$ and $d = 13$ since $13 \times 37 = 481$ and $481 \pmod{60} = 1$
6. $(37, 77)$ is sent out as the public key
7. $(13, 77)$ is kept as the private key

Now, we must convert our message into a number from 1 to $(n-1)$. Let us take the message "IB" in ASCII is 73 66. That is,

$$I = 73$$

$$B = 66$$

To encrypt we use equation 3.5, written in code in the Appendix A,

$$73^{37} \pmod{77}$$

$$\Rightarrow 17 \equiv 73^{37} \pmod{77}$$

and,

$$66^{37} \pmod{77}$$

$$\Rightarrow 66 \equiv 66^{37} \pmod{77}$$

Therefore $C = 17\ 66$.

Decrypting using equation 3.18,

$$17^{13} \equiv 73 \pmod{77}$$

and,

$$66^{13} \equiv 66 \pmod{77}$$

Giving our original message in ASCII as 73 66.This encryption and decryption method has been mathematically proven in section 3.3

3.5 Primality Tests

RSA relies on very large prime numbers for its keys. Due to the inherent nature of prime numbers, there is no such formula that has been devised that can give us a list of primes below n. This is what makes the RSA secure as it is very difficult to factor n into p, q. Therefore, the best way of finding that is testing each number for whether it is prime or not. For smaller numbers this is easy, but for larger numbers it is quite difficult, therefore primality tests are used. There are various primality tests such as Trial division and Sieve of Eratosthenes, but they are beyond the scope of this essay.

4 Elliptical Curve Cryptography

For applications, mainly those on the Internet and Blockchain, ECC or Elliptical Curve Cryptography might seem like better options due to two smaller bit sizes, that is a smaller n, while simultaneously being harder to crack. The reason for this will be explored further and can be seen in Appendix C.

4.1 Introduction

Elliptical curves are curves with the formula:

$$y^2 = x^3 + ax + b \pmod p \text{ for all } a, b \in F_p \tag{4.1}$$

Here y, x, a, b are all within F_p wherein is any finite field F_p : in any $\mathbb{R}, \mathbb{Q}, \mathbb{C}, \mathbb{Z}/p\mathbb{Z}$. The coefficients a and b give the curve its characteristics – they determine what points will be on the curve and are restricted to $\pmod p$. Furthermore, the number of points N on an Elliptical Curve is bounded by

$$p + 1 - 2\sqrt{p} \leq N \leq p + 1 + 2\sqrt{p} \tag{4.2}$$

Furthermore, it is known that Elliptical Curves, in F_p form an Abelian group also known Commutative group due to the following properties:

1. **Closure**: For all a, b in a set E, the result of the operation $a \cdot b$ is also in a set E
2. **Associativity**: For all a, b, c in a set E the equation $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
3. **Identity Element**: There exists an identity element i for all other elements in a set E such that $a \cdot i = i \cdot a = a$
4. **Commutative**: For all a, b in a set E such that $a \cdot b = b \cdot a$
5. **Distributive**: For all a, b, and c in a set E such that $a \cdot (b + c) = a \cdot b + a \cdot c$
6. **Inverse**: For a in a set E there exists an additive inverse $-a$ such that $a + (-a) = 0$ and a multiplicative inverse such that if $a \neq 0$ there is a a^{-1} such that $a \cdot a^{-1} = 1$

For the purpose of this essay, elliptical curves will be restricted in the modulo p domain such that $F_p : \mathbb{Z}/p\mathbb{Z}$.

For example, if $p = 3$, the additive and multiplicative fields in this finite field with order 7 would look like:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

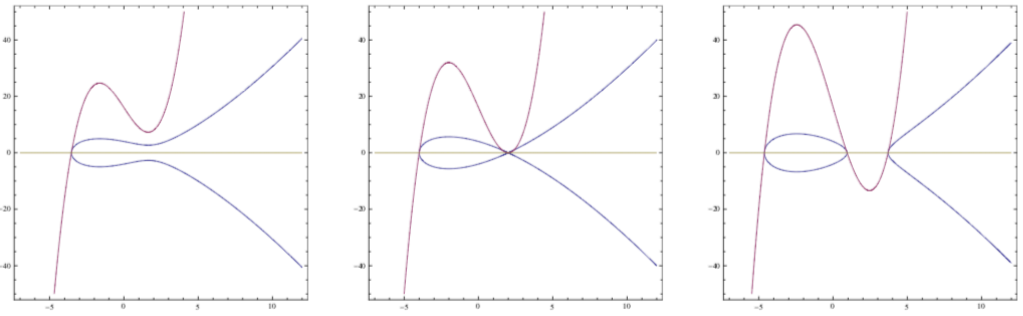
Table 1: Addition table

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Table 2: Multiplication table

4.2 Singularity case

Another condition is that the coefficients must be such that they avoid a singularity. A singularity could potentially lead to not having 3 distinct roots. In the diagrams[11] below the purple curve is that of $y = x^3 + Ax + B$ while the blue curve is that of $y^2 = x^3 + ax + b$. The value of $x^3 + ax + b$ must be positive for us to take a square root. Therefore, when the purple line is negative, there is no blue curve. So, if there is a blue curve, it will include both the positive and negative values of the square root.A singularity occurs when the purple curve is tangent to the x-axis.



The point of tangency will exist where the minimum of the curve $y = x^3 + Ax + B$, the purple line, is on the x-axis.
That is when:

$$\frac{dy}{dx} = 0$$

[11] Davis, Tom R. “Elliptic Curve Cryptography.” *Mathematical Circles* Topics, 3 Nov. 2013, www.geometer.org/mathcircles/ecc.pdf.

Taking the derivative of $y = x^3 + Ax + B$,

$$\frac{dy}{dx} = 3x^2 + A = 0$$

$$\text{For } x = \sqrt{\frac{-A}{3}} \text{ to be touching the } x\text{-axis, we need:}$$

$$\left(\sqrt{\frac{-A}{3}}\right)^3 + \left(\sqrt{\frac{-A}{3}}\right) A + B = 0$$

$$\sqrt{\frac{-A}{3}} \left(\frac{-A}{3} + A\right) = -B$$

$$\sqrt{\frac{-A}{3}} \left(\frac{2A}{3}\right) = -B$$

$$\frac{-4A^3}{27} = B^2$$

$$0 = 4A^3 + 27B^2 \tag{4.3}$$

From equation 4.3 we can see that if $4A^3+27B^2 = 0$ then there is a singularity, meaning there is a chance that is there are not 3 distinct roots and so the properties discussed in section 4.3 will not be valid.

Hence from now on we will assume the one condition the coefficients of elliptical curves must follow is that:

$$4A^3 + 27B^2 \neq 0 \tag{4.4}$$

4.3 Group Operations

Elliptical curves have many properties that are useful to making it so secure including but not limited to:

- A non-vertical line that intersects two points on the curve will always intersect a third point. Let us call that point $\mathcal{O}$. It is an artificial point that exists “at infinity”.
- A non-vertical line tangent to the curve will precisely intersect one other point on the curve.

Using these properties we can define two group operations[12]:

Point Addition: $P + Q = R$ is defined as the reflection through the x-axis of the third intersecting point R' on a line that includes P and Q.

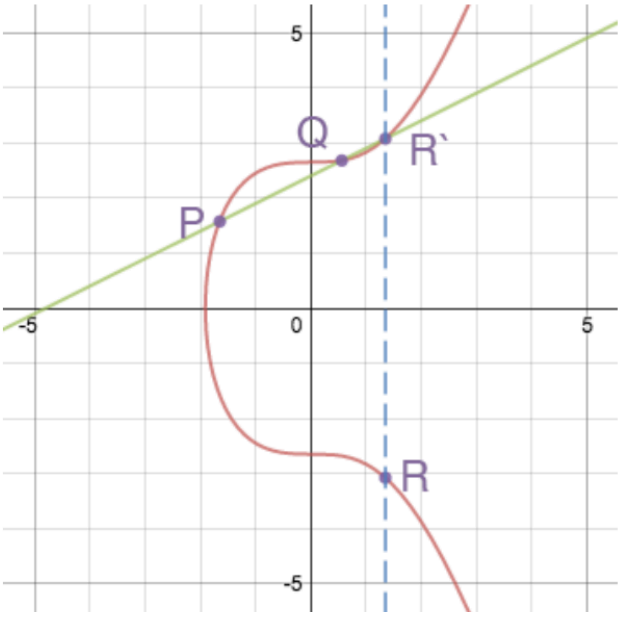


Figure 3: Point Addition

For all P, Q, and R in F_p, we have the following addition properties, taking from the properties of F_p defined in subsection 4.1:

$$
\begin{aligned}
P + Q &= Q + P \\
P + \mathcal{O} &= P \\
\mathcal{O} + \mathcal{O} &= \mathcal{O} \\
P + (Q + R) &= (P + Q) + R
\end{aligned}
$$

It also shows us that point doubling, explained below, is a special case of point addition wherein P is getting added to itself.

To algebraically derive the coordinates of R let us assume the line PQ is represented by a linear equation in the form $y = mx + c$, where m is the gradient and c is the y-intercept. Let the point P have coordinates (x_P, y_P)

[12] Davis, Tom R. "Elliptic Curve Cryptography." *Mathematical Circles Topics*, 3 Nov. 2013, www.geometer.org/mathcircles/ecc.pdf.

and the point Q have the coordinates (x_Q, y_Q). For the case where $P \neq Q$,

$$m = \frac{y_P - y_Q}{x_P - x_Q} \tag{4.5}$$

To find the intersection of the elliptical curve and the line PQ,

$$(mx + y_P)^2 = x^3 + Ax + B \tag{4.6}$$

Since (x_P, y_P), (x_Q, y_Q), and (x_R, y_R) are all solutions,

$$\begin{aligned} x^3 + Ax + B - (mx + y_P)^2 &= 0 \\ (x - x_P)(x - x_Q)(x - x_R) &= 0 \\ x^3 - x^2(x_P + x_Q + x_R) + x(x_P x_Q + x_Q x_R + x_P x_R) - x_P x_Q x_R &= 0 \end{aligned} \tag{4.7}$$

Matching coefficients gives us the following coordinates for R as (x_R, y_R),

$$\begin{aligned} x_R &= m^2 - (x_P + x_Q) \\ y_R &= m(x_P - x_R) - y_P \end{aligned} \tag{4.8}$$

Point Doubling: It is defined as $P + P = R = 2P$ and can be done by finding the line tangent to the point to be doubled, P, and then reflecting the intersecting point R' through the x-axis on the curve to get R.

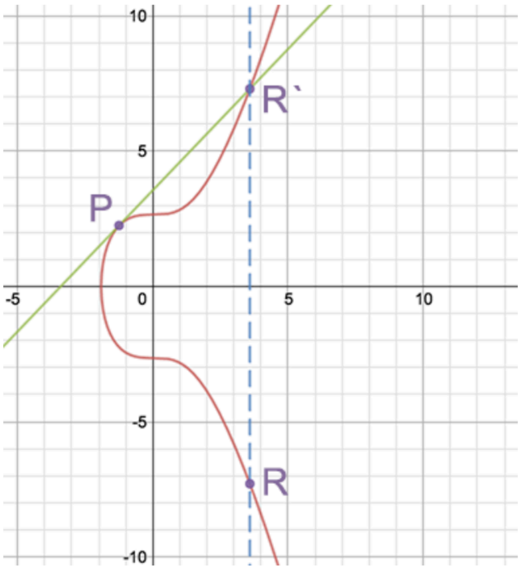


Figure 4: Point Doubling

For when $P = Q$, like in the case of Point Doubling, we can just take the derivative of the elliptical curve function (equation 4.1) at (x, y),

$$\begin{aligned} \frac{d}{dx}(y^2) &= \frac{d}{dx}(x^3 + ax + b) \\ \frac{dy}{dx}(2y) &= 3x^2 + a \\ \frac{dy}{dx} &= \frac{3x^2 + a}{2y} \end{aligned} \tag{4.9}$$

That means, if $P = Q$ at (x_P, y_P), then the gradient m at that point is,

$$m = \frac{3x_P^2 + a}{2y_P} \tag{4.10}$$

Which makes the coordinates of point R, (x_R, y_R),

$$\begin{aligned} x_R &= m^2 - 2x_P \\ y_R &= m(x_P - x_R) - y_P \end{aligned} \tag{4.11}$$

There is sometimes a special case of Point Addition or Point Doubling, when two points create a vertical point.

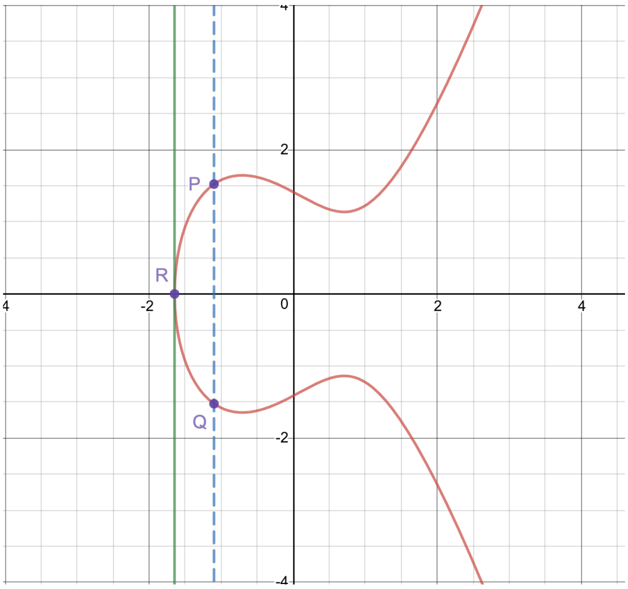


Figure 5: Vertical Point

The phenomenon shown in figure 5 happens in two cases:

1. If $P \neq Q$ and $x_P = x_Q$ then $P + Q = \mathcal{O}$
2. If $P = Q$ (represented by point R in figure 5) and $y_P = y_Q = 0$ then $P + Q = \mathcal{O}$

Together these properties of Point Addition and Point Doubling can be used for Scalar Multiplication $R = k \cdot P$ which is defined by $R = P + P + P$ (when $k = 3$), a number of times. This brings us to the definition of G known as the The Base Point or Generator Point such that for any point G on the curve, the set of all the points on that curve is $\{\mathcal{O}, G, G+G, G+G+G, ...\}$ and is called a cyclic subgroup of the points on the elliptical curve.

Definition 4. The Base Point or Generator Point: $G \in E(\mathbb{Z}/p\mathbb{Z})$ where E is an elliptical curve defined over modulo p, generates a cyclic subgroup. That is, any point in the subgroup can be computed through repeat addition of G[13].

Properties of the Generator Point:

- Order of the generator point: $ord(G) = n$, is the number of points in the group that it generates. It is also the smallest possible integer k such that $kG = \mathcal{O}$
- Cofactor: $h = \frac{|E(\mathbb{Z}/p\mathbb{Z})|}{n}$, that is the number of elements on the elliptical curve divided by the order of G. A cofactor of $n = 1$ is ideal since larger cofactors are more susceptible to attacks and are undesirable.

The methods used to find the generator point are beyond the scope of this essay as they involve the Lagrange Theorem and group theory.

4.4 Elliptical Curve Discrete Logarithm Problem

How do we go from this idea of an elliptical curve to a crytosystem that can secure everything on the internet? Looking at the assumptions in section 1.2 gives us a hint. A one-way encryption function is needed so it is easy to encrypt a message but very difficult to reverse-engineer that function. We have already encountered this in the idea of scalar multiplication.

[13] Yin, Xinchun, and Jinliang Zou. "A Parallel Base Point Choosing Algorithm of ECC on Binary Field." *International Conference on Systems and Informatics (ICSAI2012)*, May 2012, doi:10.1109/icsai.2012.6223543.

Definition 5. Elliptical Curve Discrete Logarithm Problem (ECDLP): Let E be an elliptic curve over a finite field K_p. Suppose there is a point $Q \in E(K)$, given the generator point G, it is currently computationally infeasible to compute k such that $Q = kG$ where k is the discrete logarithm of Q to the base G[14].

As seen from the definition above, what makes ECDLP so difficult to crack is the one-way nature of scalar multiplication regarding elliptical curve $\mathbb{E}$ in the finite field F_p. This one-way function or trapdoor function that secured the RSA was the factoring of primes, and for ECC it is the ECDLP. These two trapdoor or one-way functions are the basis for the security of the internet. Although primality testing has allowed for some progress to be maid when factoring prime numbers. On the other hand, there has been no such revolutionary progress for the ECDLP.

4.5 Elliptical Curve Diffie-Hellman protocol key exchange with an example

Domain parameters $\{p, a, b, G, n, h\}$ are available to both parties exchanging messages and to any third parties since it is on the public domain. An elliptical curve E is used for this process.

- p: field modulus of the curve E as defined over $\mathbb{Z}/p\mathbb{Z}$.
- a, b: curve parameters of E
- G: Generator Point
- n: $ord(G)$
- h: cofactor

Let us compute an example curve and all the parameters before beginning with the process.

For the purpose of the example let,

$$E : y^2 \equiv x^3 + 2x + 2 \pmod{17} \tag{4.12}$$

[14] Smart, N. P. "The Discrete Logarithm Problem on Elliptic Curves of Trace One." *Journal of Cryptology*, vol. 12, no. 3, June 1999, pp. 193196., doi:10.1007/s001459900052.

and therefore $G = (5, 1)$. Such small numbers are never used since they are not secure, but to illustrate the example it is sufficient.

Now we need to generate the cyclic group using G. The first step is to compute $2G$ which is $G + G$. To do this we can use the point doubling formula derived in equation 4.10:

$$m = \frac{3x_G^2 + a}{2y_G} \tag{4.13}$$

Since $G = (5, 1)$, $x_G = 5$ and $y_G = 1$ and $a = 2$ from 4.12:

$$m \equiv \frac{3(5^2) + 2}{2(1)} \equiv \frac{77}{2} \equiv 77 \cdot 2^{-1} \equiv 9 \cdot 9 \equiv 13 \pmod{17} \tag{4.14}$$

Note: $2^{-1} \pmod{17} \equiv 9$ *was computed using the Extended Euclidean Algorithm which is outside the mathematical scope of this essay.*

Next, we compute the x_{2G} and y_{2G} coordinates for $2G$ using the formula in 4.11

$$x_{2G} = m^2 - 2x_G \tag{4.15}$$

$$x_{2G} \equiv 13^2 - 2(5) \equiv 169 - 10 \equiv 16 - 10 \equiv 6 \pmod{17} \tag{4.16}$$

and,

$$y_{2G} = m(x_G - x_{2G}) - y_G \tag{4.17}$$

$$y_{2G} \equiv 13(5 - 6) - 1 \equiv -13 - 1 \equiv -14 \equiv 3 \pmod{17} \tag{4.18}$$

Therefore, the coordinates of $2G$ are $(6, 3)$. Just like this we need to compute the whole cyclic group till the last point $\mathcal{O}$. The cyclic group for our E and G is:

$$\begin{array}{lllll}
G = (5,1) & 5G = (9,16) & 9G = (7,6) & 13G = (16,4) & 17G = (6,14) \\
2G = (6,3) & 6G = (16,13) & 10G = (7,11) & 14G = (9,1) & 18G = (5,16) \\
3G = (10,6) & 7G = (0,6) & 11G = (13,10) & 15G = (3,16) & 19G = \mathcal{O} \\
4G = (3,1) & 8G = (13,7) & 12G = (0,11) & 16G = (10,11) &
\end{array}$$

By counting the number of points in the group we find that $n = 19$ and $h = 1$.

We are using the Elliptical Curve Diffie-Hellmann exchange since it is the most popular but other protocols are also in use today. The process for ECDH follows the typical public-key exchange structure we saw in section 1.1 with Bob and Alice:

1. Bob picks a private key such β such that $1 \leq \beta \leq n-1$ and computes the point $B = \beta G$ through scalar multiplication and which lies on the curve E. Let $\beta = 9$ then $B = 9G = (7, 6)$.

2. Alice picks a private key such α such that $1 \leq \alpha \leq n-1$ and computes the point $B = \alpha G$ through scalar multiplication and which lies on the curve E. Let $\alpha = 3$ then $A = 3G = (10, 6)$.

3. Bob and Alice swap the information about $A = (x_A, y_B)$ and $B = (x_B, y_B)$. This information is made public and acts as the public keys. However, this does not give the public information about α and β due to the ECDLP explained in section 4.4.

4. Alice multiplies Bob's point with her private key $P = \alpha \times B = \alpha\beta G$. This gives Alice $\alpha B = 3B = 3(9G) = 27G = 8G = (13, 7)$. Here $27G$ reduces to $8G$ since the order of the cyclic group was $n = 19$.

5. Bob multiplies Alice's point with his private key $P = \beta \times A = \beta\alpha G$. Similar to Alice, Bob computes $(13, 7)$

6. Bob and Alice now have the same point on the curve P that no one else has. They are free to use this information as they wish. For example, they can use the x-coordinate to encrypt messages. This is very secure since no third party has access to α or β and therefore cannot compute the point P due to the ECDLP.

5 Conclusion

Digital packets or messages make up all major internet processes. When we visit a website, buy something using a credit card, or send cryptocurrency over the internet, all our information is *encrypted* into these digital messages before being sent. As we have seen, RSA and ECC are two methods to secure these digital messages. **Do RSA and ECDH effectively secure digital packets?**

Going back to the assumptions in section 1.2, we can see that what we needed to assume as true allows these protocols to be secure. If someone wanted to decipher a message encrypted using the RSA, they would need the decryption key d. From the encryption process of the RSA (section 3.2) we know that e and n are shared as the public key to allow anyone to send the receiver a message.. To compute d from e and n is only possible if someone knew $\phi(n)$. Without the knowledge of the two very large primes p, q, it is very difficult to get $\phi(n)$. Intuitively, one would try to factor n into p and q. This is very difficult as one has to check for every number from 0 to $\sqrt{n}$ until they find p or q. For RSA key sizes which are usually 1024 to 2048 digits long (see Appendix B), this would take years and years, even for the fastest supercomputers. Another way to do it would be count the integers less than $n-1$ which satisfy $gcd(integer, n) = 1$. With modern RSA standards using n at least as large as 2^{1024}, this would take many years even with the fastest supercomputers.

On the other hand, ECDH, a protocol of ECC, is secure not only because of the difficulty of factoring large primes but also due to the ECDLP (section 4.4). There have been various attempts to partially solve this problem like baby-step-giant-step, Pollard rho and kangaroo, index calculus, and summation polynomials are all techniques used[15] but nothing that makes ECC unusable for even the most secure application. Furthermore, as seen in Appendix C, ECC can make use of smaller bit-sizes and provide and equivalent security to RSA.

[15] Galbraith, Steven D. and Pierrick Gaudry. "Recent progress on the Elliptical Curve Discrete Logarithm Problem. *Designs, Code, and Cryptography,* vol. 78, no. 1, 23 Nov. 2015, pp. 51-72., doi:10.1007/s10623-015-0146-7.

A Code used in encryption and decryption of RSA

Encryption in Python 3.7:

```python
e = 37
n = 77

public_key = [e,n]

plaintext = input("Enter a message: ")
plaintext = list(plaintext)

print(ord(plaintext[1]))

def encrypt(p):
        return pow(p, public_key[0], public_key[1])

for i in plaintext:
        print(encrypt(ord(i)))
```

Output: **17 66**

Decryption in Python 3.7:

```python
d = 13
n = 77

private_key = [d,n]

ciphertext = [17,66]


def decrypt(c):
        return pow(c, private_key[0], private_key[1])

for i in ciphertext:
        print(decrypt(i))
```

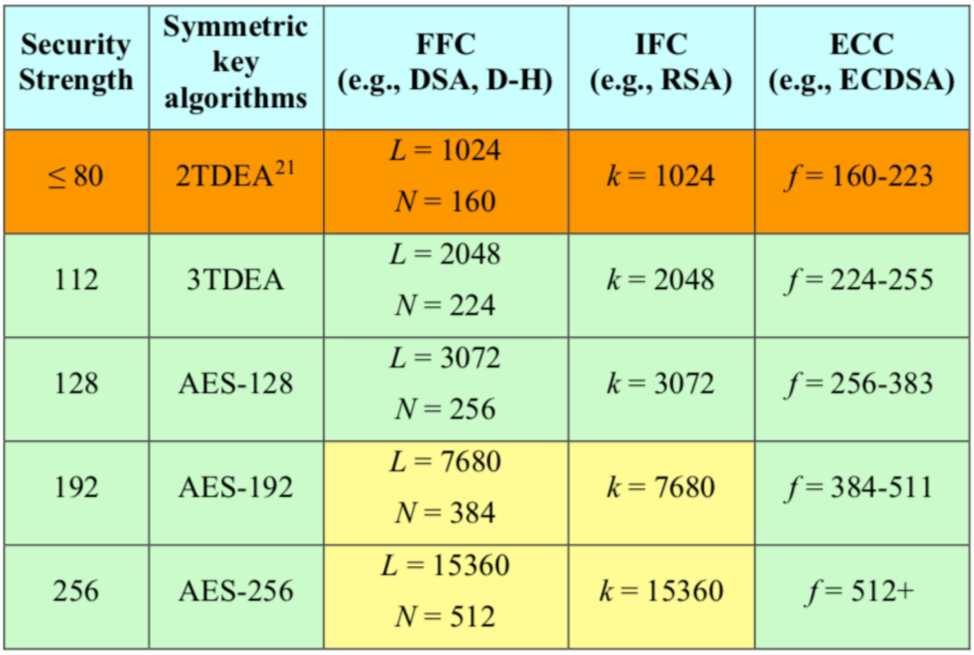
Output: **73 66**

B Recommended RSA parameters

The National Institute of Standards and Technology (NIST) part of the United States Department of Commerce recommends the key-size, that is the number of digits of n, to be **2048-bit** for the period of time from the year 2016 to 2030 in the NIST Special Publication 800-57[16].

C RSA vs. ECC, a comparison of key-size to security level

United States, Department of Commerce, National Institute of Standards and Technology. "NIST Special Publication 800-57 Part 1 Revision 4" *Recommendation for Key Management Part 1: General,* by Elaine Barker, Jan. 2016, nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf, p. 53.

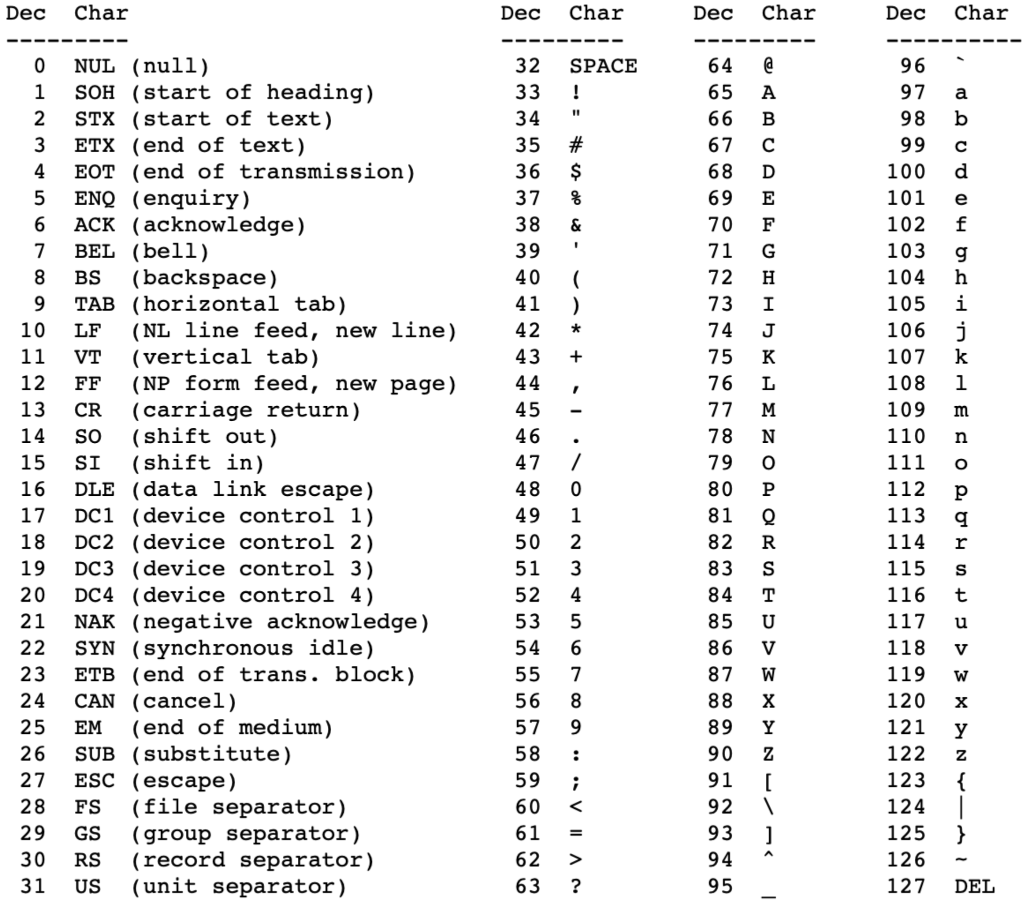


Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

[16] United States, Department of Commerce, National Institute of Standards and Technology. "NIST Special Publication 800-57 Part 1 Revision 4" *Recommendation for Key Management Part 1: General,* by Elaine Barker, Jan. 2016, nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf.

D ASCII values of most common symbols

Pattis, Richard E. *ASCII Table*, Carnegie Mellon University, www.cs.cmu.edu/ pattis/15-1XX/common/handouts/ascii.html.



Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(	72	H	104	h
9	TAB (horizontal tab)	41	)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[	123	{
28	FS (file separator)	60	<	92	\	124	|
29	GS (group separator)	61	=	93	]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL

References

Amara, Moncef, and Amar Siad. “Elliptic Curve Cryptography and Its Applications.” *International Workshop on Systems, Signal Processing and Their Applications, WOSSPA,* May 2011, doi:10.1109/wosspa.2011.5931464.

Bauer, Johannes. “Elliptic Curve Cryptography Tutorial.” *Johannes Bauer*, www.johannes-bauer.com/compsci/ecc/.

Bressoud, David M. “The RSA Public Key Crypto-System.” *Factorization and Primality Testing Undergraduate Texts in Mathematics,* 1989, pp. 4357., doi:10.1007/978-1-4612-4544-5_4.

Burton, David M. *Elementary Number Theory.* 7th ed., McGraw-Hill, Higher Education, 2011.

Corbellini, Andrea. “Elliptic Curve Cryptography: a Gentle Introduction.” *Andrea Corbellini Atom,* andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/.

Daubechies, Ingrid, and Shannon Hughes. “Lecture Notes: Cryptography Part 2.” *Math Alive,* Princeton University, web.math.princeton.edu/math_alive/index.shtml.

Euclid, and Thomas Little Heath. *The Thirteen Books of Euclids Elements.* Vol. 2, Cambridge University Press, 2015.

Fannon, Paul, et al. *Mathematics Higher Level for the IB Diploma Option Topic 10 Discrete Mathematics.* Cambridge University Press, 2013.

“Fermat’s Little Theorem.” *Brilliant Math & Science Wiki,* brilliant.org/wiki/fermats-little-theorem/.

Galbraith, Steven D., and Pierrick Gaudry. “Recent Progress on the Elliptic Curve Discrete Logarithm Problem.” *Designs, Codes and Cryptography,* vol. 78, no. 1, 23 Nov. 2015, pp. 5172., doi:10.1007/s10623-015-0146-7.

“Journey into Cryptography.” *Khan Academy,* www.khanacademy.org/computing/computer-science/cryptography#modern-crypt.

Kaliski, Burt. “The Mathematics of the RSA Public-Key

Cryptosystem." *Mathematics and Statistics Awareness Month,* Mathaware, www.mathaware.org/mam/06/Kaliski.pdf.

Lynn, Ben. "Number Theory." *Applied Cryptography Group,* Stanford University, crypto.stanford.edu/pbc/notes/numbertheory/crt.html.
Ouwehand, Martin. "The (Simple) Mathematics of RSA." *L'Autorit De Certification De L'EPFL,* certauth.epfl.ch/rsa/.

Pettofrezzo, Anthony J., and Donald R. Byrkit. *Elements of Number Theory.* Prentice-Hall, 1970.

Pierce, Robert. *Elliptic Curve Diffie Hellman,* YouTube, 10 Dec. 2014, www.youtube.com/watch?v=F3zzNa42-tQ.

Popyack, Jeffrey L. "RSA Calculator." *Introduction to Computer Science,* Drexel University, Oct. 1997,
www.cs.drexel.edu/ jpopyack/IntroCS/HW/RSAWorksheet.html.

Rivest, R. L., et al. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM,* vol. 21, no. 2, 1 Jan. 1978, doi:10.21236/ada606588.

Singh, Soram Ranbir, et al. "A Critical Review on Elliptic Curve Cryptography." *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT),* Sept. 2016, doi:10.1109/icacdot.2016.7877543.

Smart, N. P. "The Discrete Logarithm Problem on Elliptic Curves of Trace One." *Journal of Cryptology,* vol. 12, no. 3, June 1999, pp. 193196., doi:10.1007/s001459900052.

Yin, Xinchun, and Jinliang Zou. "A Parallel Base Point Choosing Algorithm of ECC on Binary Field." *International Conference on Systems and Informatics (ICSAI),* May 2012, doi:10.1109/icsai.2012.6223543.

Cryptosystem." *Mathematics and Statistics Awareness Month,* Mathaware, www.mathaware.org/mam/06/Kaliski.pdf.

Lynn, Ben. "Number Theory." *Applied Cryptography Group,* Stanford University, crypto.stanford.edu/pbc/notes/numbertheory/crt.html.
Ouwehand, Martin. "The (Simple) Mathematics of RSA." *L'Autorit De Certification De L'EPFL,* certauth.epfl.ch/rsa/.

Pettofrezzo, Anthony J., and Donald R. Byrkit.*Elements of Number Theory.* Prentice-Hall, 1970.

Pierce, Robert.*Elliptic Curve Diffie Hellman,* YouTube, 10 Dec. 2014, www.youtube.com/watch?v=F3zzNa42-tQ.

Popyack, Jeffrey L. "RSA Calculator." *Introduction to Computer Science,* Drexel University, Oct. 1997,
www.cs.drexel.edu/ jpopyack/IntroCS/HW/RSAWorksheet.html.

Rivest, R. L., et al. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM,* vol. 21, no. 2, 1 Jan. 1978, doi:10.21236/ada606588.

Singh, Soram Ranbir, et al. "A Critical Review on Elliptic Curve Cryptography." *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT),* Sept. 2016, doi:10.1109/icacdot.2016.7877543.

Smart, N. P. "The Discrete Logarithm Problem on Elliptic Curves of Trace One." *Journal of Cryptology,* vol. 12, no. 3, June 1999, pp. 193196., doi:10.1007/s001459900052.

Yin, Xinchun, and Jinliang Zou. "A Parallel Base Point Choosing Algorithm of ECC on Binary Field." *International Conference on Systems and Informatics (ICSAI),* May 2012, doi:10.1109/icsai.2012.6223543.