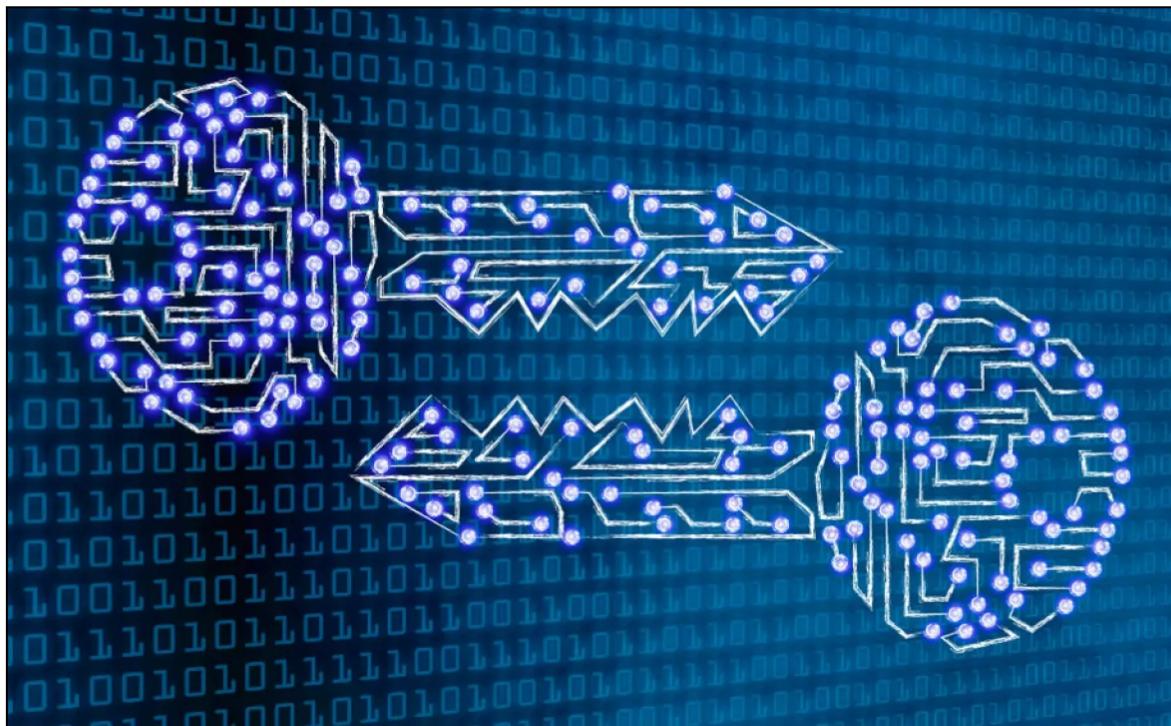


Criptografia

De l'antiguitat al futur de la Criptografia



Raúl Cano Esbrí

Pau Villanúa Hidalgo

Roc Rodríguez Moral

Batxillerat Tecnològic

Yolanda Redondo

INS Torre del Palau

2022

Aquest és el treball de recerca del Raúl Cano, Pau Villanúa i Roc Rodriguez, juntament amb la Yolanda Redondo com a tutora.

En aquest treball parlem de la criptografia, tant de la seva història com del seu present i futur i hem creat una pàgina web en la que es pot trobar tota la teoria juntament amb la part pràctica de programació i la nostra pròpia creació d'un mètode de xifratge.

http://raul.cano-esbri.com/tdr_criptografia/

“Anyone who tries to create his or her own cryptographic primitive is either a genius or a fool. Given the genius/fool ratio of our species, the odds aren’t very good.”

Bruce Schneier. Criptògraf nord-americà

ÍNDEX

INTRODUCCIÓ	4
1. Introducció a la Criptografia	5
2. Història de la criptografia	6
2.1. Criptografia a l'Antiguitat	6
2.2. Grècia i l'Imperi Romà	7
2.3. Edat Mitjana	8
2.4. Xifrat Polialfabètic	9
2.4.1. Xifrat d'Alberti	9
2.4.2. Xifrat de Vigenère	9
2.5. Edat Moderna	10
2.6. La segona guerra mundial	11
2.6.1. Màquina Enigma	11
2.6.1.1. Funcionament	13
2.7. El boom dels 80 i 90	16
2.8. En l'actualitat	16
3. Criptografia Clàssica	18
3.1. Xifratge Monoalfabètic	18
3.2. Xifratge Polialfabètic	18
3.3. Xifrat Cèsar	19
3.4. Xifrat Vigenère	19
3.5. Xifrat Alberti	20
3.6. Anàlisi de freqüència	21
4. Criptografia Moderna	23
4.1. Clau simétrica	23
4.1.1. Data Encryption Standard - DES	23
• Estructura Bàsica	24

• La funció (F) de Feistel	25
• Generació de subclaus	26
4.1.2. Xifratge de flux	26
4.1.3. Xifratge en bloc	27
4.1.3.1. Seguretat	27
4.1.3.2. Tamany dels Blocs	28
4.1.3.3. Tipus d'encriptació de Blocs	28
• Electronic Code Book (ECB)	28
- Avantatges d'utilitzar ECB:	29
- Desavantatges d'utilitzar ECB:	29
• Cipher Block Chaining (CBC)	29
- Avantatges de CBC	31
- Desavantatges de CBC	31
• Comparació ECB i CBC	32
4.2. Clau asimètrica	33
4.2.1. Principals algorismes	34
4.2.1.1. RSA	34
Creació de les claus	34
4.2.1.2. ECDSA	35
4.2.1.3. EdDSA	36
4.2.1.4. ElGamal	37
Exemple	37
4.2.2. Firma digital	38
5. Aplicacions	39
5.1. Aplicacions de xifratge	39
5.2. Aplicacions firma electrònica	40
5.3. Certificats digitals	42
• Xifrar comunicacions	45
• Signar missatges i documents	45

• Identificació davant un sistema o autenticació d'usuaris	45
PART PRÀCTICA	47
Introducció	47
1. Programació d'una pàgina web	47
1.1. Xifratge Cèsar	47
1.2. Xifratge Vigenère	49
1.3. Encriptació Asimètrica: RSA	50
1.4. Pàgina Web	55
2. Encriptació amb notes musicals	56
CONCLUSIONS	60
BIBLIOGRAFIA	62
WEBGRAFIA	62
BANC D'IMATGES	65
ANNEXOS	67
ANNEX 1: TERMINOLOGIA	67
ANNEX 2: QUADRAT DE POLIBI	69
ANNEX 3: FUTUR DE LA CRIPTOGRAFIA	71
1. Criptografia Quàntica	71
a. Criptografia Quàntica:	74
b. Criptografia Post-Quàntica:	75

INTRODUCCIÓ

Vam escollir aquest treball perquè ens va semblar un tema molt interessant del qual sabíem realment poc, donat això, vam decidir dividir el treball en dues parts, una part teòrica en la que hem investigat sobre la història de la encriptació i els mètodes d'encriptació actuals, i una part pràctica en la que utilitzem aquests coneixements adquirits durant la part teòrica i hem creat un programa per a encriptar textos mitjançant els diferents mètodes de xifratge, tant clàssics com moderns.

El nostre objectiu del treball era conèixer els orígens de la criptografia i els primers mètodes de xifratge, com el xifrat cèsar o el xifrat vigenère i descobrir com han anat evolucionant fins arribar a mètodes de xifratge més moderns, com el xifratge de blocs o l'encriptació de clau pública.

Una pregunta que ens vam fer des d'un inici era quines són les diferències entre la criptografia clàssica i la moderna, quina diferència hi ha en la seguretat dels diferents sistemes i, la que probablement és més important, quin serà el futur de la criptografia en quant a la seguretat dels actuals mètodes d'encriptació i la creació de nous mètodes amb l'aparició de ordinadors més potents i fins i tot l'aparició d'ordinadors quàntics, els quals suposarien una revolució en el camp de la criptografia similar a la que es va presenciar amb l'aparició dels primers ordinadors.

A mesura que investigavem i apreniem sobre els diferents mètodes d'encriptació vam decidir crear un nosaltres mateixos, utilitzant els conceptes aprenuts, però que alhora fos diferents a tots els que havíem vist durant la creació de la part teòrica, és així com vam tenir la idea de crear un mètode de xifratge que al encriptar un text donés com a resultat una sèrie de notes musicals representades sobre un pentagrama.

1. Introducció a la Criptografia

En l'actualitat vivim en l'anomenada era tecnològica, la qual avança molt ràpidament i ho fa amb evolucions contínues, les actualitzacions dels nostres sistemes s'acceleren de manera exponencial com mai havíem vist. La societat occidental ha apostat al cent per cent per una metodologia funcional de caràcter digital, és possible que ens trobem a les portes del final de l'era manual.

El futur se'ns presenta, i és un futur que aglutina ordinadors i processadors informàtics, bases de dades, sistemes digitals, intel·ligència artificial i que es desprèn del que és vell i "antiquat" per fomentar una política consumista de masses, "compra l'última versió i compra-la ara, aprofita i aconsegueix la felicitat que necessites". Davant d'aquest canvi constant la conclusió és clara i precisa, adaptar-se o morir, i és que es tracta en essència d'un procés de selecció natural un cop més, els individus que s'adaptin a les dificultats que se'ls presenten, sobreviuran i s'acabaran imposant com a la nova espècie dominant.

Dit d'una altra manera, aquesta era digital se sosté per mitjà de grans pilars transversals que aglutinen dades, sí perquè el futur són dades, doc's bé, aquells que posseeixin els coneixements que faciliten el control, d'atribució, emmagatzematge i venda d'aquest nombre incalculable de dades, que són necessaris per assegurar l'èxit del futur que estem llaurant, aquells seran autosuficients i exitosos davant aquesta nova realitat. El mot que aglutina tots els afers esmentats anteriorment s'anomena criptografia i és la raó del fet que els nostres comptes bancaris siguin segurs, o que vostè pugui escriure un whatsapp al seu amic, i que aquest missatge sigui encriptat impossibilitant l'atac de possibles intrusos.

Tant les majors superpotències com els estats més petits d'aquest planeta, basen la defensa de la seva nació en confiança d'aquest model que aglutina i estructura dades. Davant d'aquesta realitat, nosaltres entenem i concebem la necessitat de saber i manejar aquesta doctrina que parteix de pur càlcul matemàtic, però que també desenvolupa enginy i creativitat.

2. Història de la criptografia

Moltes vegades, sigui o bé per ignorància, o bé per desconeixement, tendim a pensar que la paraula criptografia va lligada a les missions d'espionatge internacional, les guerres, així com a organismes o entitats com la NASA o la Interpol.

Però la realitat està molt més lluny d'això, ja que només cal pensar que quan ens connectem a serveis com Gmail o WhatsApp estem establint una comunicació segura, i, per tant, xifrada amb el nostre ordinador o dispositiu mòbil i els servidors informàtics. Per tal d'assegurar la nostra privacitat, els nostres missatges estan codificats mitjançant algorismes complexos que busquen precisament evitar la intrusió d'un agent extern a la xarxa, el qual podria posar en risc la seguretat de les nostres dades.

La necessitat de mantenir la informació a resguard d'ulls curiosos no és nova, és molt més antiga del que, potser, ens podem arribar a imaginar. Com a primer pas per a entendre com funciona la criptografia, fem un breu repàs als seus gairebé quatre mil anys d'història.

La criptografia s'encarrega, precisament, de xifrar o codificar missatges per a evitar que el seu contingut pugui ser llegit per un tercer no autoritzat; és a dir, la generació de codis i algorismes de xifratge que busquen ofuscar la informació i protegir-la d'"ulls curiosos" és la comesa principal d'aquesta disciplina.

2.1. Criptografia a l'Antiguitat

Es coneix l'existència de tècniques criptogràfiques primitives des de temps remots, ja que la majoria de civilitzacions antigues semblen haver-les usat d'una forma o una altra. El reemplaçament de símbols, la forma més bàsica de criptografia, es pot trobar tant en antigues escriptures mesopotàmiques com egípcies. L'exemple més antic conegut d'aquesta manera de criptografia es va descobrir en la tomba d'un noble egipci anomenat Khnumhotep II, que va viure fa aproximadament uns 3.900 anys.

El propòsit del reemplaçament de símbols en la inscripció de Knhumhotep II no era ocultar informació, sinó incrementar el seu atractiu lingüístic. El cas més antic conegut de criptografia enfocada a protegir informació confidencial, és el d'un escriba mesopotàmic de fa 3.500 anys que va emprar la tècnica per a ocultar una fórmula per a setinat de ceràmica en una tauleta d'argila.



Il·lustració 1: Missatge xifrat a una tauleta d'argila.

2.2. Grècia i l'Imperi Romà

En períodes posteriors de l'antiguitat, la criptografia seria àmpliament utilitzada per a la protecció d'importants informacions militars, una funció que encara avui dia compleix. A la ciutat grega d'Esparta, els missatges s'encriptaven en ser escrits en un pergamí col·locat en un cilindre d'una mesura particular, la qual cosa feia que el missatge fos indecodificable fins que el recipient l'enrotllava en un cilindre similar. De manera semblant, se sap que els espies de l'antiga Índia feien servir missatges codificats ja en el segle II aC.



Il·lustració 2: Escítala amb missatge encriptat.

Probablement, la criptografia més avançada del món antic va ser la dels romans. Un exemple destacat de criptografia romana, coneguda com el xifratge del Cèsar, consistia a canviar les lletres d'un missatge encriptat sobre la base d'un cert

nombre de posicions en l'alfabet llatí. Si es coneixia el sistema i el nombre de posicions que havien de moure's les lletres, qualsevol recipient podia descodificar amb èxit el missatge secret.

2.3. Edat Mitjana

Al llarg de l'edat mitjana, la criptografia es tornaria cada vegada més important, però els xifratges per substitució (el xifratge del Cèsar és un exemple) continuarien sent l'estàndard. La criptoanàlisi, la ciència encarregada de resoldre codis i xifratges, va començar a posar-se al nivell d'una encara relativament primitiva ciència criptogràfica. Al-Kindi, un cèlebre matemàtic àrab, desenvoluparia entorn del 800 dC una tècnica coneguda com a anàlisi de freqüència, que deixava en situació de vulnerabilitat als xifratges per substitució.

a	b	c	d	e	f	g	h	ij	l	m	n	o	p	q	r	sz	t	uv	y
ð	ƿ	ƿx	æ	x	ȝ	ȝn	ȝ	ȝz	ȝr	ȝt	ȝp	ȝw	ȝf	ȝȝ	ȝk	ȝȝ	ȝȝ	ȝȝ	
o	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	
ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	ȝ	
								wȝ											
⇒ groupe nul									Cx par -or que ð vous										

Il·lustració 3: Exemple de llenguatge de xifratge antic.

Per primera vegada, la gent que intentava desxifrar missatges encriptats tenia a la seva disposició un mètode sistemàtic per a aconseguir-lo, la qual cosa va obligar la criptografia a evolucionar per a continuar sent útil.

2.4. Xifrat Polialfabètic

2.4.1. Xifrat d'Alberti

En 1465, Leone Alberti va desenvolupar el xifratge polialfabètic, considerat la solució contra la tècnica d'anàlisi de freqüència d'Al-Kindi. En un xifratge polialfabètic, el missatge es codifica utilitzant dos alfabets diferents.

Un és l'alfabet en què el missatge original s'escriu, mentre el segon és un alfabet totalment diferent, en el qual el missatge es mostra després de ser codificat. En combinació amb els xifratges de substitució tradicionals, els xifratges polialfabètics incrementaven enormement la seguretat de la informació codificada. Tret que el lector conegués l'alfabet en què el missatge havia estat originalment escrit, l'anàlisi de freqüència resultava inútil.

2.4.2. Xifrat de Vigenère

El xifratge Vigenère és un xifratge basat en diferents sèries de caràcters o lletres del xifratge Cèsar formant aquests caràcters una taula, anomenada taula de Vigenère, que s'usa com a clau. El xifratge de Vigenère és un xifratge polialfabètic i de substitució.

El xifratge Vigenère s'ha reinventat moltes vegades. El mètode original va ser descrit per Giovan Battista Belaso al seu llibre de 1553 "La xifra del Sig. Giovan Battista Belaso", qui va construir el xifratge basant-se en la taula recta de Trithemius, però va afegir una clau repetida per canviar cada caràcter entre els diferents alfabets. No obstant això, va ser incorrectament atribuït al segle XIX a Blaise de Vigenère, a partir d'un treball realitzat el 1583, i per això encara se'l coneix com el "xifrat Vigenère".

2.5. Edat Moderna

Nous mètodes per a codificar informació serien també desenvolupats durant el Renaixement, incloent-hi un primerenc mètode popular de codificació binari inventat en 1623 pel cèlebre erudit Sir Francis Bacon.

La ciència criptogràfica continuaria progressant en els següents segles. Un notable avanç en criptografia seria descrit, però potser mai construït, per Thomas Jefferson en la dècada de 1790. El seu invent, conegut com a roda de xifratge, consistia en 36 anells de lletres en rodes móbils, que podien ser utilitzats per a aconseguir codificats complexos. Aquest concepte era tan avançat que serviria com base de la criptografia militar americana fins al període de la Segona Guerra Mundial.



Il·lustració 4: Roda de xifratge.

2.6. La segona guerra mundial

Durant una guerra, és de vital importància que, en cas que l'enemic intercepti un missatge, li sigui impossible llegir el seu contingut. Per a aconseguir-ho, durant la Segona Guerra Mundial i gràcies a l'avanç de la tecnologia es van crear nous mètodes d'encriptació.

Com durant la Segona Guerra Mundial els missatges es transmetien mitjançant l'ús de la ràdio, qualsevol persona amb un receptor i la freqüència adequada pot escoltar aquests missatges.

Per a evitar que els enemics interceptessin aquests missatges es van haver d'inventar nous mètodes d'encriptació. En l'antiguitat, l'encriptació es feia a mà, i avui en dia s'utilitzen ordinadors, però durant la Segona Guerra Mundial, abans de la invenció dels ordinadors, es van crear noves màquines per a facilitar la tasca d'encriptació i dificultar la desencriptació.

Els alemanys van crear la que posteriorment seria la màquina d'encriptar més famosa de la història: La Màquina Enigma.

2.6.1. Màquina Enigma

La Segona Guerra Mundial portaria amb si l'exemple perfecte de criptografia analògica: la màquina Enigma. Igual que la roda de xifratge, aquest dispositiu, emprat per les potències de l'Eix, emprava rodes rotatories per a codificar un missatge, fent que fos virtualment impossible llegir-lo sense una altra màquina Enigma. Primerenques formes de tecnologia informàtica serien usades per a eventualment ajudar a trencar el xifratge d'Enigma. L'exitós desencriptat dels missatges d'Enigma es considera un component clau de la posterior victòria aliada.



Il·lustració 5: Màquina Enigma.

La màquina Enigma va ser inventada per Arthur Scherbius, un enginyer alemany expert en electromecànica, amb la intenció de millorar els sistemes de criptografia utilitzats pels exèrcits. El febrer del 1918 va patentar la seva idea que consistia a utilitzar un xifrat similar al xifrat de Vigenère. Com que Scherbius no tenia els recursos per poder-lo produir es va associar amb Willi Korn, propietari de l'empresa Enigma Chiffiermaschinen AG. Tots dos junts van millorar el disseny i el van aplicar al xifrat de secrets comercials, el 1923 el van presentar a l'Exposició Postal Internacional de Berlín.

En un inici, aquesta màquina podia ser comprada per qualsevol, però, a poc a poc, va anar guanyant importància dins de les forces armades, fins que l'exèrcit es va apoderar completament de la màquina i la van retirar del mercat.

Quan Enigma va passar al control de l'exèrcit, es va afegir un quart rotor en algunes de les màquines per a fer-les més segures. Tot i que les màquines que havien estat comercialitzades no s'assemblaven gaire a la Enigma definitiva i havien sigut retirades, algunes van acabar en mans dels exèrcits dels aliats i van permetre que eventualment s'acabessin desxifrant els missatges encriptats per Enigma.

2.6.1.1. Funcionament

La Màquina Enigma encripta els missatges mitjançant encriptació de substitució, de forma similar a l'encriptació Cèsar, però de forma molt més elaborada. Cada vegada que s'encripta una lletra, canvia tot el mètode d'encriptació, tot i que escriquis dues vegades la mateixa lletra, aquesta serà encriptada de formes diferents i tindran com a resultat lletres diferents.

Una màquina Enigma està formada per diverses parts, com ara un teclat, una placa de llum, rotors i circuits electrònics interns. Algunes màquines, com les utilitzades pels militars, tenen característiques addicionals, com ara un tauler de clavilles.



Il·lustració 6: Parts de la Màquina Enigma

Cada vegada que es prem una lletra, un o més rotors es mouen per a crear una nova configuració dels rotors, aquesta configuració és la que determina com

s'encriptarà la lletra i quina serà la lletra resultant. Una vegada s'ha encriptat la lletra, s'il·lumina la lletra resultant en la placa de llum.

Cada mes, els operadors d'Enigma rebien llibres de codis que especificaven quins paràmetres utilitzaria la màquina cada dia. Cada matí el codi canviaria.

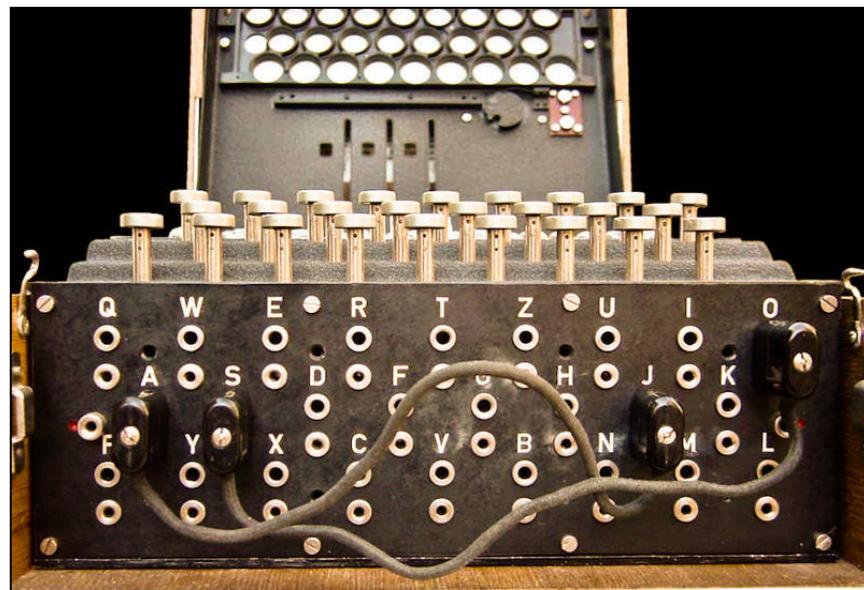
Per exemple, un dia, el llibre de codis pot enumerar la configuració descrita a la clau del dia següent:

- a. **Configuració del connector:** A/L – P/R – T/D – B/N – K/F – O/Y
- b. **Disposició del rotor (o scrambler):** 2 — 3 — 1
- c. **Orientacions del rotor:** D – K – P

- a. **Configuració del connector:** A/L – P/R – T/D – B/N – K/F – O/Y

Una placa de connexió és similar a una placa de commutació de telèfon antiga amb deu cables, cada cable té dos extrems que es poden connectar a una ranura. Cada cable d'endoll pot connectar dues lletres per formar un parell (connectant un extrem del cable a la ranura d'una lletra i l'altre extrem a una altra lletra). Les dues lletres d'un parell s'intercanviaran, de manera que si "A" està connectada amb "Z", "A" es converteix en "Z" i "Z" es converteix en "A". Això proporciona un nivell addicional de seguretat per a l'encriptació.

Per implementar aquesta clau, primer, s'han d'intercanviar les lletres A i L connectant-les al tauler, canviar P i R connectant-les al tauler, i després el mateix amb els altres parells de lletres esmentats anteriorment. Essencialment, un extrem d'un cable es connectaria a la ranura "A" i l'altre extrem es connectaria a la ranura L. Abans que es produeixi cap configuració dels rotors, això afegeix una primera capa de codificació on les lletres connectades pel cable es codifiquen entre si. Per exemple, si s'intercanvien la T i la P, platan s'encriptaria com tlapan, i això després s'encripta mitjançant els rotors.

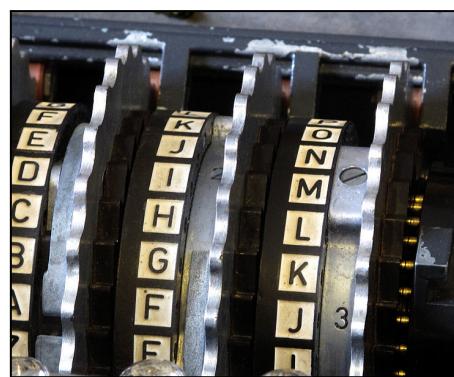


Il·lustració 7: Connector intercanvi de lletres. Màquina Enigma.

b. Disposició del rotor (o scrambler): 2 — 3 — 1

Les màquines Enigma venien amb diversos rotors diferents, cada rotor proporcionant un esquema de codificació diferent. Per codificar un missatge, les màquines Enigma agafaven tres rotors alhora, un en cadascuna de les tres ranures. Cada combinació diferent de rotors produiria un esquema de codificació diferent. Nota: la majoria de màquines Enigma militars tenien tres ranures de rotor, tot i que algunes en tenien més.

Per aconseguir la configuració anterior, col·loqueu el rotor #2 a la primera ranura de l'enigma, el rotor #3 a la segona ranura i el rotor #1 a la tercera ranura.



Il·lustració 8: Rotors de la màquina enigma.

c. Orientacions del rotor: D – K –P

A cada rotor, hi ha un alfabet al llarg de la vora, de manera que l'operador pot establir una orientació determinada. Per a aquest exemple, l'operador giraria el rotor a la ranura 1 perquè es mostri D, giraria la segona ranura perquè es mostri K i giraria la tercera ranura perquè es mostri P.



Il·lustració 9: Orientació dels rotors d'una màquina enigma.

2.7. El boom dels 80 i 90

Amb el boom de les computadores, la criptografia va aconseguir nivells de progrés molt majors que en l'era analògica. L'encriptació matemàtica de 128-bits, molt més fort que qualsevol xifratge antic o medieval, és ara l'estàndard per a molts dispositius sensibles i sistemes informàtics. En 1990, es posaria en marxa tota una nova forma de criptografia, sobrenomada criptografia quàntica, per part de científics computacionals que esperaven elevar una vegada més el nivell de protecció ofert per l'encriptació moderna.

2.8. En l'actualitat

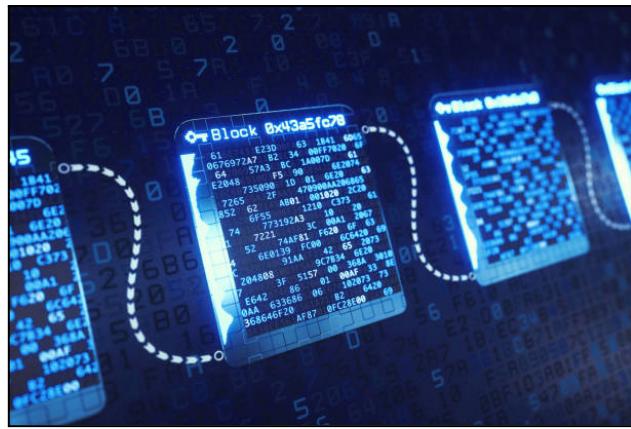
Més recentment, tècniques criptogràfiques han estat també utilitzades per a fer possibles les criptomonedes. Aquestes aprofiten diverses tècniques criptogràfiques avançades, com les funcions *hash*, la criptografia de clau pública i les signatures digitals. Aquestes tècniques es fan servir principalment per a garantir la seguretat de les dades emmagatzemades en *blockchains* i per a autenticar les transaccions. Una forma especialitzada de criptografia, anomenada

Elliptic Corbi Digital Signature Algorithm (ECDSA), serveix de puntal a Bitcoin i a altres sistemes de criptomonedes, en proporcionar una seguretat complementària i garantir que els fons només poden ser usats pels seus legítims propietaris.



Il·lustració 10: Bitcoin.

La criptografia ha recorregut un llarg camí en els últims quatre mil anys, i no sembla que vagi a aturar-se aviat. En la mesura en què informació crítica o delicada continuï requerint protecció, la criptografia continuarà avançant. A pesar que els sistemes criptogràfics emprats actualment en les *blockchains* de les criptomonedes representen algunes de les formes més avançades d'aquesta ciència, són també peces d'una tradició que abasta bona part de la història humana.



Il·lustració 11: Blockchain.

3. Criptografia Clàssica

3.1. Xifratge Monoalfabètic

En un xifratge monoalfabètic, es xifren totes les lletres utilitzant una única clau, per tant, cada vegada que aparegui una mateixa lletra en el text, es xifra de la mateixa forma i tindrà la mateixa lletra resultant. A causa d'això és possible desencriptar el text xifrat mitjançant una simple anàlisi de freqüència.

Alguns exemples de xifratge monoalfabètic són el xifratge Cèsar on cada lletra es desplaça en funció d'una clau numèrica, i el xifratge atbash, on cada lletra s'assigna a la lletra simètrica respecte al centre de l'alfabet.

3.2. Xifratge Polialfabètic

Un xifrat polialfabètic és qualsevol xifrat basat en la substitució, utilitzant diversos alfabets de substitució. En els xifratges de substitució polialfabètica, les lletres de text pla es xifren de manera diferent segons la seva posició en el text. En lloc de ser una correspondència un a un, hi ha una relació d'un a molts entre cada lletra i els seus substituts.

Per exemple, "a" es pot xifrar com "d" a l'inici del text, però com "n" al mig. Els xifratges polialfabètics tenen l'avantatge d'amagar la freqüència en què apareixen les lletres al text. Per tant, l'atacant no pot utilitzar la freqüència estàtica de lletres individuals per desencriptar el text xifrat.

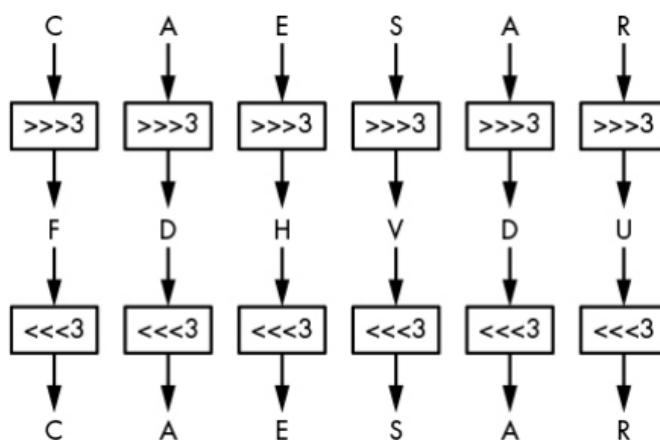
El primer xifrat polialfabètic va ser el xifrat Alberti que va ser introduït per Leon Battista Alberti l'any 1467 com solució a l'anàlisi de freqüència.

3.3. Xifrat Cèsar

El Xifrat Cèsar o “Caesar cipher” xifra un missatge desplaçant cadascuna de les lletres un nombre determinat de posicions endavant de l’alfabet, tornant a girar cap a A si el canvi arriba a Z. Aquest número de desplaçaments és la clau.

Per a desxifrar un missatge xifrat, s’han de tornar a moure les lletres aquest mateix nombre de posicions enrere.

Per exemple, fent servir la clau 3, ZOO xifra a CRR, FHVDU desxifra a CESAR.



Il·lustració 12: Exemple de xifratge i desxifratge de la paraula Caesar. Xifratge Cèsar.

3.4. Xifrat Vigenère

El xifrat Vigenère va ser creat al segle XVI per un italià anomenat Giovan Battista Bellaso.

El xifrat de Vigenère és similar al xifrat de Cèsar, excepte que les lletres no es desplacen totes el mateix nombre de posicions, sinó per valors definits per una clau formada per una col·lecció de lletres que representen números en funció de la seva posició a l’alfabet. Per exemple, si la clau és DUH, les lletres del text es desplacen utilitzant els valors 3, 20, 7 perquè D està tres lletres després de A, U està 20 lletres després de A i H està set lletres després de A. El patró 3, 20, 7 es repeteix fins que s’ha xifrat tot el text sense format. Per exemple, la paraula CRYPTO xifraria a FLFSNV utilitzant DUH com a clau: C es desplaça tres posicions a F, R es desplaça 20 posicions a L, i així successivament.

En aquest exemple es pot veure la frase “Torre del Palau” sent encriptada mitjançant el xifrat vigènere utilitzant la clau DUH.

T	O	R	R	E	D	E	L	P	A	L	A	U
D-3	U-20	H-7	D-3									
W	I	Y	U	Y	K	H	F	W	D	F	H	X

II·Il·lustració 13: Exemple de Xifrat Vigènere.

3.5. Xifrat Alberti

El xifrat Alberti és un tipus de xifrat polialfabètic. Un xifrat polialfabètic és similar a un xifrat de substitució. En alguns casos, els alfabets múltiples són només rotacions de l'alfabet existent.

Fer a xifrar i desxifrar es fa servir un disc d'Alberti. El disc d'Alberti clàssic està format per dos discs, un exterior fix i un interior mòbil. El disc exterior està dividit en 24 parts, i en cada part s'escriu una lletra de l'A a la Z excepte les lletres H, J, K, U, W, Y, i al final s'escriuen els nombres de l'1 al 4.

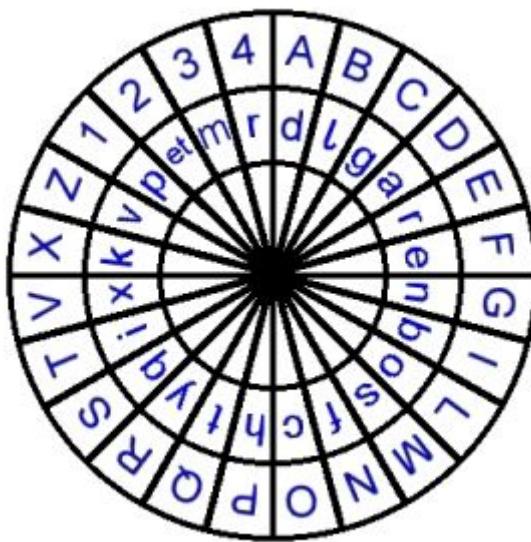
En el disc interior es tornen a escriure les lletres, però aquest cop no s'escriuen en ordre, sinó que s'escriuen de forma aleatòria. S'escriuen les lletres de l'A a la Z excepte la J, U i W, i en l'últim espai s'escriu et.

En un disc d'Alberti actual, els discs es divideixen en 26 lletres i s'escriu l'alfabet sencer en ells.

Per a encriptar, s'agafa una lletra clau del disc interior, per exemple k i s'alinea amb una lletra qualsevol del disc exterior, per exemple B i s'informa d'aquesta lletra al receptor del missatge. A partir d'aquí, per a escriure el missatge, cada lletra del disc exterior es representarà amb la lletra del disc interior amb la que coincideixi. Després d'escriure tres o quatre lletres, es pot tornar a canviar la

posició de la lletra clau i, per exemple, alinear la k amb la D i en el missatge s'escriu una D majúscula. A partir d'aquest punt k ja no significarà B sinó D. Aquest procés es continua repetint fins a completar el missatge.

Com les lletres encriptades no tenen el mateix resultat al llarg de tot el missatge, el missatge xifrat no es pot desxifrar mitjançant una simple anàlisi de freqüència.



Il·lustració 14: Disc d'Alberti

3.6. Anàlisi de freqüència

Els xifratges monoalfabètics, com el xifrat Cèsar, es caracteritzen per utilitzar una mateixa clau durant tot el text, és a dir, si encriptem A com a H, durant tot el text, totes les H seran una A.

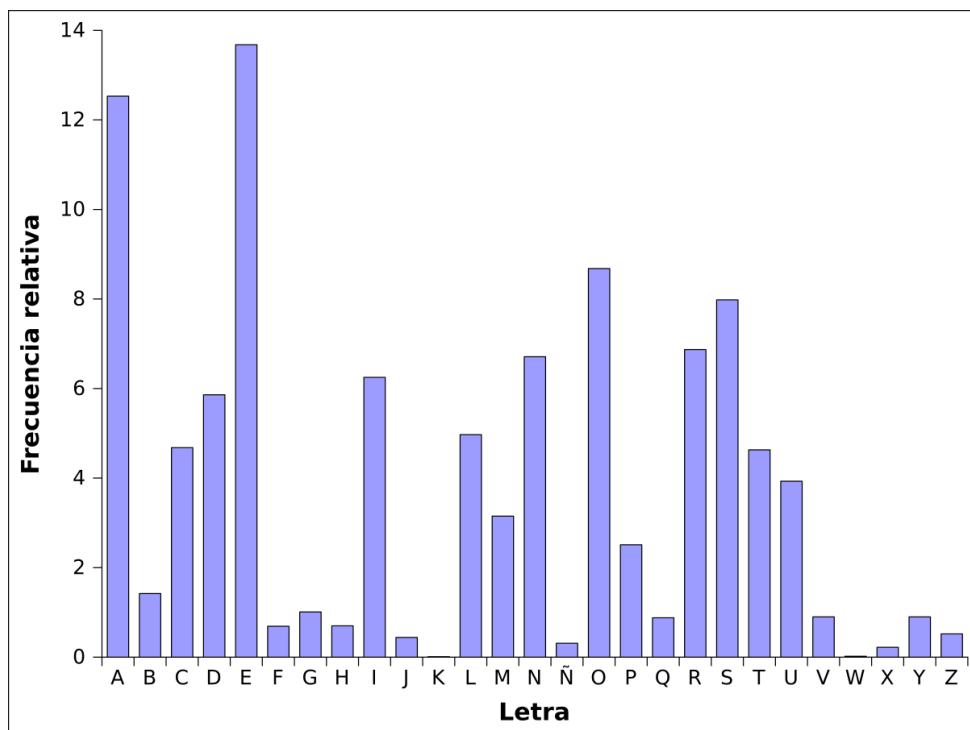
Fins a l'edat mitjana, aquests tipus de xifrat eren considerats segurs, però en el segle IX, el matemàtic àrab Al-Kindi va escriure el llibre "Manuscrit sobre el desxiframent de missatges criptogràfics" en el que va descriure el mètode d'anàlisi de freqüència per a desxifrar aquest tipus de xifrat.

Per a poder dur a terme aquest mètode de desxifrar, necessitem un text, de mínim una pàgina de llarg i sense xifrar, que estigui escrit en el mateix idioma que el text

xifrat, llavors hem de comptar quantes vegades apareix cada lletra en aquest text i ordenar les lletres en ordre de les que més apareixen a les que menys. Una vegada tenim aquesta llista hem de fer el mateix amb el text xifrat.

Les lletres més utilitzades en el text sense xifrar segurament també seran les més utilitzades en el text xifrat, per tant, podem veure com estan xifrades per a saber quina és la clau i desxifrar tot el text fent servir aquesta clau. Si el resultat és un text amb coherència, hem aconseguit desxifrar-lo, però si el text no té sentit, podem repetir el procés amb una altra de les lletres més usades.

A més d'això, podem tenir en compte les característiques pròpies de cada lletra: per exemple, la q en castellà va sempre seguida d'una u, la lletra x sol anar precedida de la lletra e... Les lletres amb característiques tan evidents són una perdició per a qualsevol que vulgui utilitzar un xifratge monoalfabètic. Per això, molts criptògrafs no fan ús de la q, sinó que empren la k i totes les x les substitueixen per s. Encara que el missatge sense xifrar tingui faltes d'ortografia, el seu sentit es recupera fàcilment.



Il·lustració 15: Taula de freqüència de lletres en castellà.

4. Criptografia Moderna

Una de les aportacions del darrer quart del s. XX són els sistemes de xifratge asimètric o de clau pública (com RSA), en contraposició amb tots els anteriors, que són criptosistemes simètrics o de clau privada, que feien servir la mateixa clau per al xifratge i el desxifrat del missatge. L'avantatge d'aquests sistemes és que permeten solucionar un dels problemes de criptografia clàssica, la distribució de les claus secretes als participants en la comunicació. A la criptografia de clau pública, una de les claus es pot fer pública sense que per això la seguretat de la clau secreta es vegi afectada. El xifrat amb la clau secreta es pot desxifrar amb la pública i viceversa. Aquesta propietat dels criptosistemes asimètrics permet també altres aplicacions d'aquests criptosistemes, com la signatura digital que és tan important a les xarxes de telecomunicacions avui.

4.1. Clau simètrica

La criptografia de clau simètrica, o encriptació simètrica, utilitza una clau secreta per al xifratge i el desxifrat. Aquest enfocament és l'invers del xifratge asimètric, que fa servir una clau per a xifrar i una altra per a desxifrar. Les dades es tradueixen a un format que no pot ser interpretat o inspeccionat per algú que no tingui la clau secreta feta servir per a xifrar-los durant aquesta fase.

El rendiment del generador de nombres aleatoris emprat per a generar la clau secreta determina l'eficàcia d'aquest mètode. La criptografia de clau simètrica, la més utilitzada a Internet avui dia, comprèn dos tipus d'algorismes: El de bloc i el de flux.

4.1.1. Data Encryption Standard - DES

L'algoritme DES (Data Encryption Standard) va ser el primer algoritme de xifrat per blocs. DES utilitza blocs de 64 bits i la clau és de 64 bits però l'algoritme utilitza només 56 bits i els altres 8 s'utilitzen per a comprovar la paritat i després es descarten.

- **Estructura Bàsica**

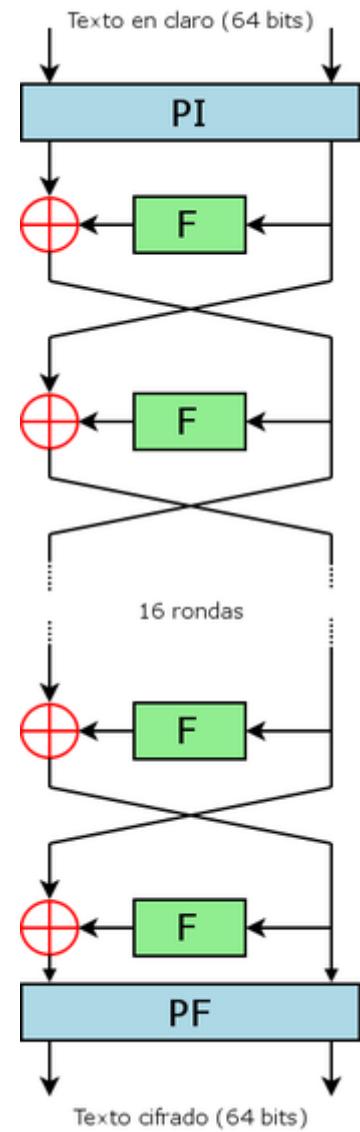
L'estructura bàsica de l'algoritme apareix representada a la Figura 1:

Hi ha 16 fases idèntiques de procés, anomenades rondes. També hi ha una permutació inicial i final anomenades PI i PF, que són funcions inverses entre si (PI "desfà" l'acció de PF, i viceversa).

PI i PF no són criptogràficament significatives, però es van incloure presumptament per facilitar la càrrega i descàrrega de blocs sobre el maquinari de mitjans dels setanta.

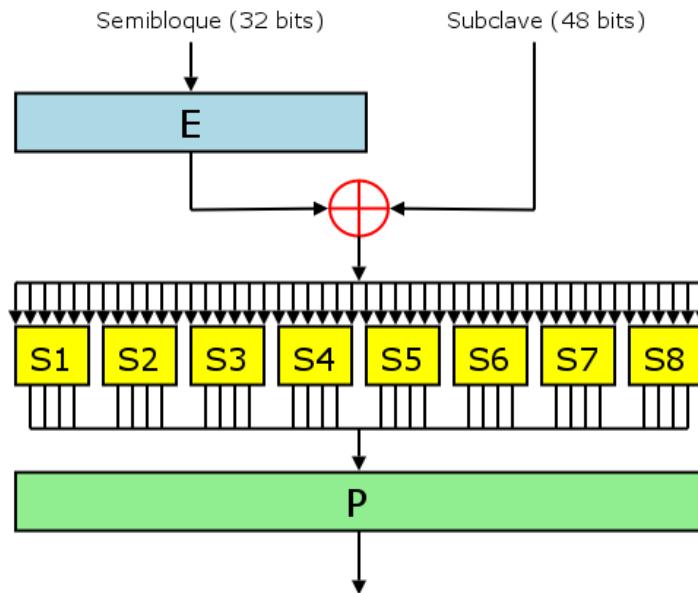
Abans de les rondes, el bloc és dividit en dues meitats de 32 bits i processades alternativament. Aquest entrecreuament es coneix com a esquema Feistel.

El símbol vermell " \oplus " representa l'operació XOR.



Il·lustració 16: Estructura DES.

- **La funció (F) de Feistel**



Il·lustració 17: Funció de Feistel.

La funció (F) de Feistel opera sobre mig bloc (32 bits) i consta de quatre passos:

1. **Expansió:** La meitat del bloc de 32 bits s'expandeix a 48 bits mitjançant la permutació d'expansió (E) duplicant alguns bits. Això passa perquè el bloc de 32 bits es divideix en 8 blocs, amb cada bloc format per 4 bits i llavors s'afegeixen 2 bits més a cada bloc.
2. **Barreja:** El resultat de l'expansió es combina amb la subclau utilitzant l'operació XOR. Es deriven 16 subclaus de la clau inicial, una per a cada ronda, mitjançant la generació de subclaus descrita més a sota.
3. **Substitució:** Després de ser barrejat amb la subclau, el bloc es divideix en 8 fragments de 6 bits abans de ser processats per les S-caixes, o caixes de substitució. Cadascuna de les vuit S-caixes reemplaça els seus sis bits d'entrada amb quatre bits de sortida, d'acord amb una transformació no lineal, especificada per una taula de cerca. Les S-caixes constitueixen el nucli de la seguretat de DES — sense, el xifratge seria lineal, i fàcil de trencar.
4. **Permutació:** finalment, les 32 sortides de les S-caixes es reordenen d'acord amb una permutació fixa; la P-caixa.

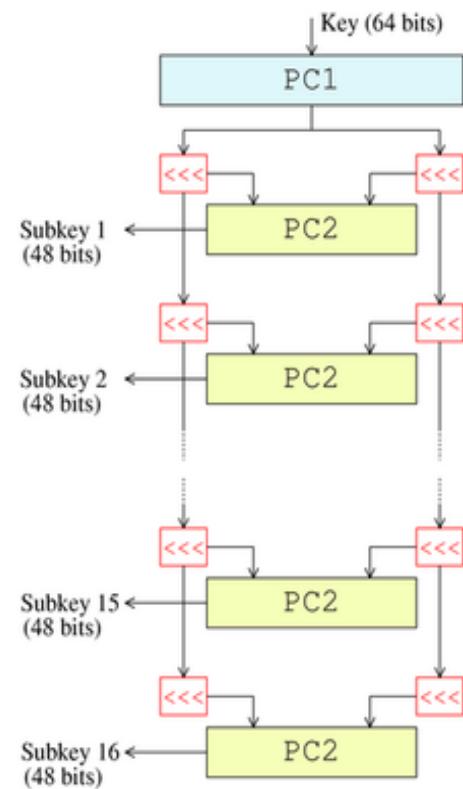
- **Generació de subclaus**

Primer, se seleccionen 56 bits de la clau dels 64 inicials mitjançant l'Elecció Permutada 1 (PC-1)

Els 56 bits es divideixen llavors en dues meitats de 28 bits; tot seguit cada meitat es tracta independentment.

En rondes successives, ambdues meitats es desplacen cap a l'esquerra un o dos bits (dependent de cada ronda), i aleshores se seleccionen 48 bits de subclau mitjançant l'Elecció Permutada 2 (PC-2) — 24 bits de la meitat esquerra i 24 de la dreta.

La generació de claus per a desxiframent és similar, però s'han de generar les claus en ordre invers.



Il·lustració 18: Generació de claus.

4.1.2. Xifratge de flux

Un xifratge de flux és un mètode de xifratge on es combina un flux de xifres pseudo-aleatòries (pretetenen donar al text una aparença normal tot i ocultar un missatge xifrat) amb díigits de text normal.

Aquest flux de xifres s'aplica a cada dígit binari, un bit a la vegada. Aquest mètode de xifratge utilitzava un nombre infinit de díigits de xifratge pseudo-aleatoris per clau. El xifratge de flux també es coneix com xifrat d'estat.

Un xifratge de flux xifra una longitud arbitrària de text normal, amb un algorisme que fa servir una clau. Perquè aquesta forma de xifratge es mantingui segura, els seus díigits de xifratge pseudo-aleatori han de ser imprevisibles i la clau mai no s'ha d'emprar més d'una vegada.

Els díigits de xifratge es generen a través de diversos valors aleatoris que fan servir registres de desplaçament digitals. El xifratge de cada dígit depèn de l'estat actual de la xifrada, justificant això per a l'estat del nom. RC4 és un popular xifrat de flux que s'usa àmpliament en programació.

4.1.3. Xifratge en bloc

Un xifratge de bloc consisteix en dos algoritmes; un algoritme d'encriptació i un de desencriptació.

L'algoritme d'encriptació (E) utilitza una clau K , i un bloc de text sense xifar P i produeix el text xifrat C . L'operació d'encriptació s'escriu com a: $C = E(K, P)$.

L'algoritme de desencriptació (D) funciona de forma inversa l'algoritme d'encriptació. Mitjançant la clau K desencripta el text encriptat C per arribar al text original P .

Com els algoritmes d'encriptació i desencriptació són inversos, impliquen operacions similars.

4.1.3.1. Seguretat

Perquè un bloc encriptat sigui segur hauria de ser una permutació pseudoaleatoria, en anglès *Pseudorandom Permutation* (PRP). Això vol dir que mentre la clau sigui secreta, un atacant no hauria de poder calcular un resultat del xifratge de bloc a partir de cap entrada. És a dir, sempre que K sigui secreta i aleatòria des de la perspectiva d'un atacant, no haurien de tenir ni idea sobre com és $E(K, P)$ per a qualsevol P donat.

A més a més, un atacant no hauria de poder trobar cap patró en el text encriptat, en altres paraules, hauria de ser impossible diferenciar el text encriptat d'un conjunt de lletres realment aleatòries.

4.1.3.2. Tamany dels Blocs

El xifratge de blocs té dos valors d'importància: la mida del bloc i la grandària de la clau.

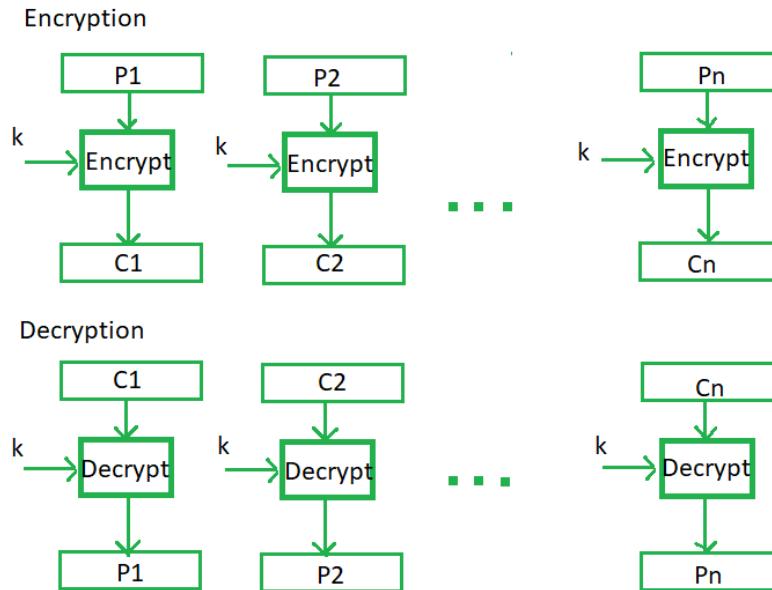
La majoria de blocs són de 64 bits o bé de 128 bits. Els blocs del xifratge DES són de 64 bits i les claus de 64 bits, dels quals només 56 són utilitzats per l'algoritme. DES va ser desenvolupat per IBM el 1975, però ja no es considera segur, com que la clau és molt curta. El 1998, DES va ser reemplaçat per AES, que té blocs de 128 bits i la clau és de 128, 192 or 256 bits.

Quan la longitud o la petjada de memòria dels textos xifrats són crítiques, es poden emprar blocs de 64 bits, perquè aquests produeixen textos xifrats més curts i consumeixen menys memòria. Si aquest no és el cas, és millor fer servir blocs de 128 bits, principalment perquè es poden processar blocs de 128 bits de forma més eficient que els de 64 bits a les CPU modernes i també són més segurs. A més a més, les CPU poden aprofitar instruccions especials de la CPU per processar de manera eficient un o més blocs de 128 bits en paral·lel.

4.1.3.3. Tipus d'encriptació de Blocs

- **Electronic Code Book (ECB)**

L'ECB és el mode de funcionament de xifrat de blocs més fàcil. És el més fàcil per què encripta cada bloc de forma directa i independent i dóna com a resultat blocs de text xifrat.



Il·lustració 19: Representació del funcionament d'encriptació ECB.

- **Avantatges d'utilitzar ECB:**

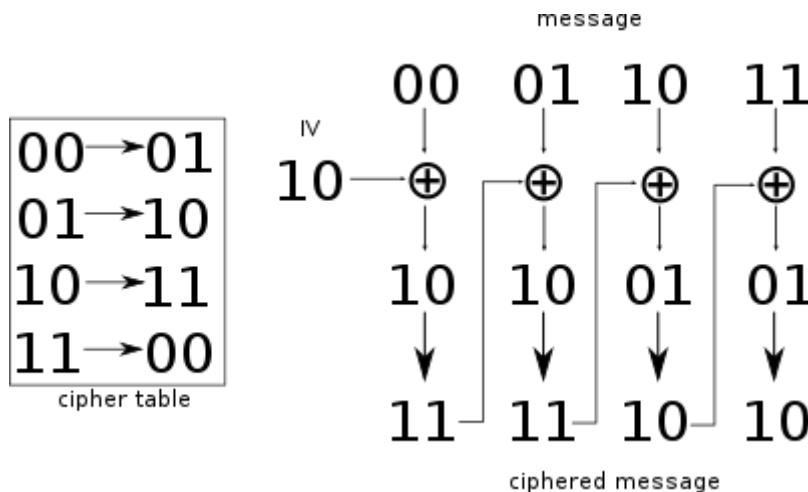
- És possible xifrar els diferents blocs de forma paral·lela, per tant, és una forma més ràpida de xifratge.
- Mètode senzill de xifratge de blocs.

- **Desavantatges d'utilitzar ECB:**

- Propens a criptoanàlisi, ja que hi ha una relació directa entre el text pla i text xifrat.

- **Cipher Block Chaining (CBC)**

El xifrat CBC és un avenç respecte al xifrat ECB, ja que el ECB compromet alguns requisits de seguretat. A CBC, el bloc de xifratge anterior es dóna com a entrada al següent algorisme de xifratge després de XOR amb el bloc de text original. En poques paraules, un bloc de xifratge es produeix xifrant una sortida XOR del bloc de xifrat anterior i el bloc de text pla present.



II·lustració 20: Exemple d'encriptació CBC

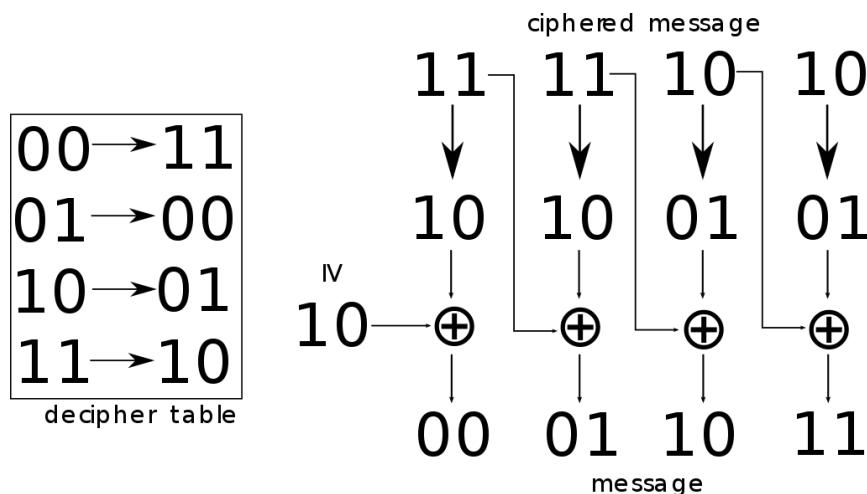
En aquesta imatge podem veure com funciona l'encriptació CBC. El primer bloc s'encripta fent un XOR del text pla inicial (00) i un vector d'inicialització (10).

Entrada 1	Entrada 2	Resultat XOR
0	0	0
0	1	1
1	0	1
1	1	0

II·lustració 21: Taula XOR

El resultat del XOR de 00 i 10 és 10, ja que 0 i 1 es queda com a 1, i 0 i 0 queda com a 0.

Llavors s'encripta el bloc resultant del XOR i ja tenim el primer bloc xifrat. A continuació es fa el XOR d'aquest bloc xifrat i el següent bloc sense xifrar, seguint la mateixa taula i s'encripta el resultat. Aquest procés es repeteix fins que està tot xifrat.



Il·lustració 22: Exemple desencriptació CBC

Per a desencriptar, primer es desencripta el primer bloc i després es fa un XOR amb el bloc desencriptat i el vector d'inicialització, això ens deixa el primer bloc completament desxifrat. A continuació es desencripta el segon bloc i es fa el XOR amb el segon bloc desencriptat i el primer encriptat això ens deixa el segon bloc completament desxifrat. Aquest procés es repeteix amb la resta dels blocs fins que aconseguim tenir tots els blocs desxifrats.

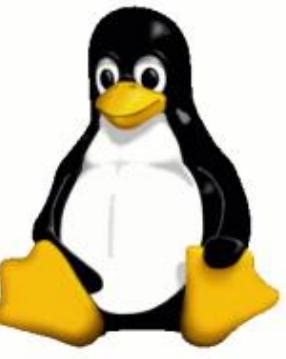
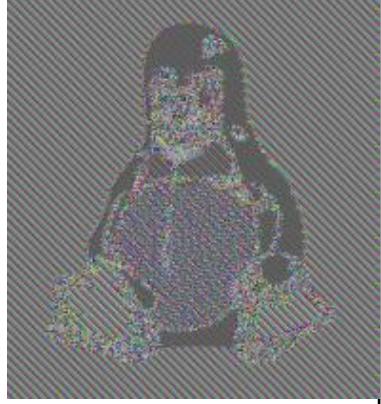
- **Avantatges de CBC**

- CBC és un bon mecanisme d'autenticació.
- Millor naturalesa resistent a la criptoanàlisi que l'ECB.

- **Desavantatges de CBC**

- El xifratge paral·lel no és possible, ja que cada xifratge requereix un xifrat previ.

- Comparació ECB i CBC

		
Imatge Original.	Encriptada amb ECB. (Es pot veure un patró)	Encriptada mitjançant CBC (Pseudoaleatori)

Il·lustració 23: Comparació encriptació ECB i altres mètodes pseudo-aleatoris

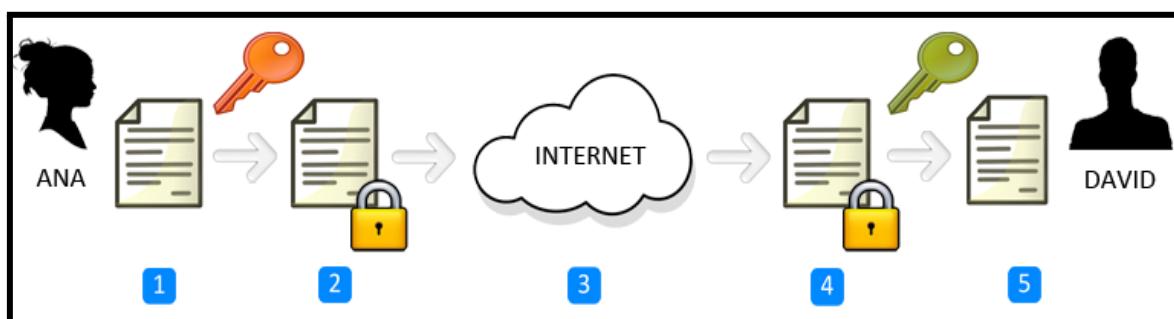
En aquest exemple podem veure la diferència d'encriptar una imatge amb ECB o amb algun altre mètode. Al encriptar mitjançant ECB, com és un mètode senzill i que encripta els blocs individualment podem veure un patró resultant de la imatge original. Aquest tipus de patrons fan que l'encriptació sigui susceptible a criptoanàlisi.

Per l'altra banda, si la imatge és encriptada utilitzant qualsevol altre mètode, com per exemple el CBC, al no encriptar els blocs d'un en un i fer que s'alteren uns als altres, queda un resultat pseudoaleatori, és a dir, que sembla un conjunt de punts completament aleatori, sense que contingui cap informació.

4.2. Clau asimètrica

La criptografia asimètrica o criptografia de clau pública xifra i desxifra un missatge utilitzant dues claus similars. A la criptografia de clau asimètrica, s'usen dues claus: una clau privada, i una pública. La persona que vol rebre el missatge ha de crear les claus de forma aleatòria i fa pública una clau que es fa servir per a encriptar, mentre que la clau utilitzada per a desencriptar els missatges és privada i només la pot fer servir ell mateix. Qualsevol pot fer servir una clau pública per xifrar un document, de manera que només el receptor previst pugui desxifrar-lo amb la clau privada. La clau privada o secreta només la coneix el generador de la clau.

El principal avantatge de la criptografia asimètrica és la major seguretat de les dades. Com que no s'espera que els usuaris revelin o intercanvien les seves claus privades, es redueixen els riscos de l'activitat cibernètica sobre la clau privada d'un usuari durant la transmissió.



Il·lustració 24: Exemple d'encriptació asimètrica.

- 1.David crea una clau pública i una privada.
- 2.David comparteix la clau pública amb qui li vol enviar un missatge
- 3.Anna redacta un missatge i encripta fent servir la clau pública.
- 4.David rep el missatge encriptat i el desencripta amb la seva clau privada.

D'aquesta forma, qualsevol persona amb la clau pública li pot enviar un missatge a David, però només ell els pot llegir, ja que es necessita la clau privada que no ha compartit amb ningú.

4.2.1. Principals algorismes

4.2.1.1. RSA

La encriptació mitjançant l'algorisme RSA es basa en el fet de que és difícil factoritzar un nombre enter molt gran.

Per a crear les claus utilitzades en el RSA, s'han de generar dos nombres primers, i a partir d'aquests es crea una clau privada i una pública. La clau pública està formada per dos nombres, un que es el resultat de la multiplicació dels dos nombres primers i un altre que també es treu a partir d'aquests nombres.

Per tant, si algú pot factoritzar el producte dels nombres primers, la clau privada es veu compromesa. La seguretat del xifratge depèn totalment de la mida de la clau i si doblem o tripliquem la mida de la clau, la força del xifratge augmenta de manera exponencial. Les claus RSA solen tenir una longitud de 1024 o 2048 bits, però els experts creuen que les claus de 1024 bits es podrien trencar en un futur pròxim. Però fins ara sembla ser una tasca inviable.

Creació de les claus

1. Es generen dos nombres primers de forma aleatòria, anomenats **p** i **q**
 - Ex: **p = 61 // q = 53**
2. Multipliquem **p** i **q** per a trobar **n**. **n** s'utilitza per al mòdul, tant en l'encriptació com en la desencriptació. La seva llargada s'expressa en bits i es la llargada de la clau. **n** es dóna a conèixer com a part de la clau pública.
 - Ex: $61 * 53 = 3233$. **n = 3233**
3. Calculem $f(n)$. $f(n) = (p-1)*(q-1)$
 - Ex: $f(n) = (61-1)*(53-1) = 3120$
4. Busquem un nombre **e** tal que: $1 < e < \lambda(n)$ i $\text{mcd}(e, \lambda(n)) = 1$
 El valor triat més habitualment per a **e** és $2^{16} + 1 = 65537$. El valor més petit (i més ràpid) possible per a **e** és 3
 - Ex: **e = 3**

5. Trobar d tal que $(e * d) \bmod f(n) = 1$

- EX: $(3 * d) \bmod 3120 = 1$

Un cop generades les claus, n i e formen la clau pública i es fan públiques per a que qualsevol pugui utilitzar-les per a encriptar un text que només el creador de les claus podrà desencriptar, utilitzant les claus privades, que son n i d .

a. Per a encriptar:

- i. $c(m) = m^e \bmod n$
- ii. Per a encriptar el missatge m hem d'elevar-lo a la clau e i fer el mòdul de la divisió d'aquesta potència entre n .
- iii. $c(m)$ és el missatge encriptat
- iv. m és el missatge que volem encriptar.

b. Per desencriptar:

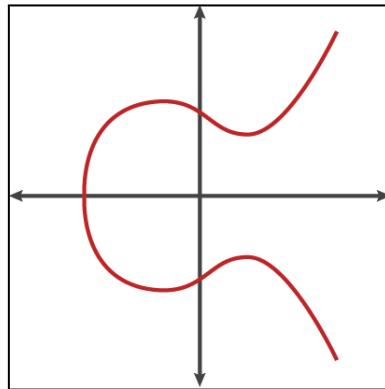
- i. $m(c) = m^d \bmod n$
- ii. Per a desencriptar el missatge c hem d'elevar-lo a la clau d i fer el mòdul de la divisió d'aquesta potència entre n .
- iii. $m(c)$ és el missatge desencriptat
- iv. c és el missatge que volem desencriptar.

4.2.1.2. ECDSA

L'algoritme de signatura digital de corba el·líptica (ECDSA) és un algoritme de signatura digital (DSA) que utilitza claus derivades de la criptografia de corba el·líptica. És una equació particularment eficient basada en criptografia de clau pública.

Una característica principal de l'ECDSA confront d'un altre algorisme popular, RSA, és que ECDSA proporciona un major grau de seguretat amb longituds de clau més curtes. Això augmenta encara més el seu ROI, ja que ECDSA fa servir menys potència informàtica que RSA, que menys segura.

ECDSA s'usa en molts sistemes de seguretat, és popular per al seu ús en aplicacions de missatgeria segura i és la base de la seguretat de Bitcoin (amb "direccions" de Bitcoin que serveixen com a claus públiques).

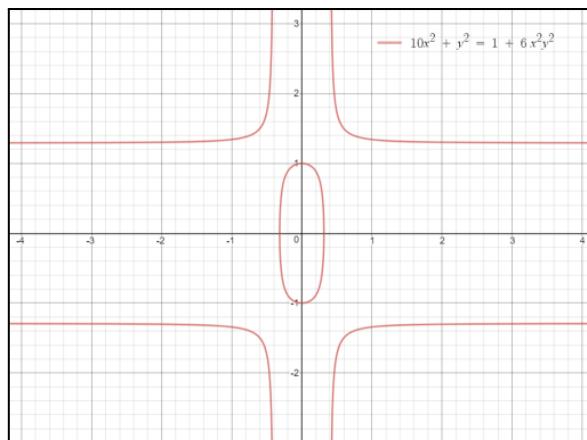


Il·lustració 25: Corba el·líptica ECDSA.

4.2.1.3. EdDSA

EdDSA és un esquema de signatura digital basat en Schnorr que funciona sobre corbes el·líptiques. Tot i que ECDSA és probablement l'esquema de signatura digital de corba el·líptica més estès, EdDSA té una sèrie de propietats que el converteixen en una alternativa atractiva a ECDSA.

La principal diferència és el tipus de corba amb què es treballa (ambdues el·líptiques, però EdDSA utilitza la corba Edward, que permet un càlcul més ràpid i segur de la multiplicació de punts).



Il·lustració 26: Corba el·líptica EdDSA.

4.2.1.4. ElGamal

El xifratge ElGamal és un criptosistema de clau pública. Utilitza xifratge de clau asimètrica per a comunicar-se entre dues parts i xifrar el missatge.

Aquest criptosistema es basa en la dificultat de trobar logaritmes discrets en un grup cíclic, és a dir, encara que sapiguem g^a i g^k , és extremadament difícil calcular g^{ak} .

Exemple

Suposem que Alícia vol comunicar-se amb Bob.

1. *Bob genera claus públiques i privades:*
 - Bob escull un nombre molt llarg q i un grup cíclic F_q
 - Del grup cíclic F_q , escull qualsevol element g i un element a , quedaria $\gcd(a, q) = 1$
 - Després computa $h = g^a$
 - Bob pública F , $h = g^a$, q i g com a clau pública i manté a com a privada.
2. *Alícia encripta informació utilitzant la clau pública d'en Bob*
 - Alícia selecciona un element k del grup cíclic F , quedant $\text{mcd}(k, q) = 1$
 - Més tard computa $p = g^k$ i $s = h^k = g^{ak}$
 - Multiplica s amb M
 - Posteriorment envia $(p, M*s) = (g^k, M*g^ak)$
3. *Bob desencripta el missatge*
 - Bob calcula $s' = p^a = g^{ak}$
 - Divideix $M*s$ amb s' per a obtenir M com a $s = s'$

4.2.2. Firma digital

La signatura digital d'un document s'obté després d'una operació en tres passes:

1. S'aplica al document un algorisme matemàtic que crea una empremta digital anomenada hash.
2. Aquest hash és un número que identifica de manera inequívoca el document.
3. El hash s'cripta usant la clau privada del signant.
4. El hash encriptat i la clau pública del signant es combinen en una signatura digital que s'afegeix al document.

Per verificar l'autenticitat del document el receptor ha de tenir un programa que suporti signatures digitals. El programa utilitza la clau pública per desencriptar la clau hash. Després calcula un nou hash per al document. D'aquesta manera podem comparar el hash calculat amb el hash desencriptat; si coincideixen, el document no ha estat modificat. Així mateix, el programa valida que la clau pública utilitzada a la signatura pertany al nom que l'ha signat.

5. Aplicacions

5.1. Aplicacions de xifratge

L'aplicació principal dels algorismes de xifrat és garantir la confidencialitat dels documents encara que aquests resultessin accessibles a persones no autoritzades. La situació pràctica en la qual més s'utilitza és la transferència d'informació per canals de comunicació no segurs, com és Internet.

Les tècniques de xifratge, a més de garantir la confidencialitat, garanteixen col·lateralment la integritat, ja que si el document no és accessible per a usuaris no autoritzats, tampoc és modificable.

Els algoritmes de xifrat juguen un paper decisiu en la transferència d'arxius, per exemple per correu electrònic, i en la transferència d'informació mitjançant navegadors, per exemple durant l'accés a la pàgina web d'un banc.

També s'empren les tècniques de xifratge per a protegir documents importants dins del disc dur o en qualsevol mitjà d'emmagatzematge digital, per si es produeix un accés il·legal. Un accés no autoritzat a informació confidencial pot tenir lloc tant per un accés il·lícit al sistema com pel robatori dels medis físics d'emmagatzematge (especialment en ordinadors portàtils o flash Drive).

Per a garantir la confidencialitat, es poden utilitzar tant les tècniques de xifratge de tipus simètric on existeix una clau secreta que s'utilitza per a xifrar i desxifrar, com les tècniques de xifratge de tipus asimètric on existeix una clau pública i una altra privada.

El gran avantatge dels mètodes asimètrics és que permeten iniciar les comunicacions sense haver d'haver acordat prèviament, i de manera segura, una clau secreta. No obstant això, aquests algorismes són molt costosos des del punt de vista computacional, per la qual cosa es tendeix a utilitzar mètodes simètrics.

En la majoria de les aplicacions pràctiques, es fan servir els mètodes asimètrics per a iniciar una comunicació segura durant la qual s'estableix una clau secreta generada aleatoriament i s'acorda un algorisme de xifratge simètric a partir de les preferències dels dos interlocutors. A partir d'aquest moment la comunicació té lloc per mitjà d'algorismes simètrics que són més eficients.

Una de les aplicacions més esteses de les tècniques de xifratge en la comunicació en manera segura del navegador d'Internet, és a dir quan l'usuari posa https en lloc d'http. En establir una connexió https el navegador sol·licita la clau pública del servidor i després s'estableix la comunicació mitjançant algoritmes simètrics. En el cas del correu electrònic xifrat, se segueix l'estàndard S/ACARONI que es basa en les normes PKCS#7. Anàlogament, s'apliquen xifratges simètrics amb claus autogenerades i més tard aquestes claus es xifren amb algoritmes asimètrics.

5.2. Aplicacions firma electrònica

La signatura electrònica s'utilitza per a aconseguir dues de les característiques de seguretat: integritat i autenticació. És important destacar que un document electrònic signat pot ser públic, perquè integritat i autenticació no impliquen necessàriament confidencialitat. Per tant, els mètodes de xifratge de clau secreta no són apropiats per a realitzar signatura electrònica, i, no obstant això, els mètodes asimètrics sí que poden aplicar-se.

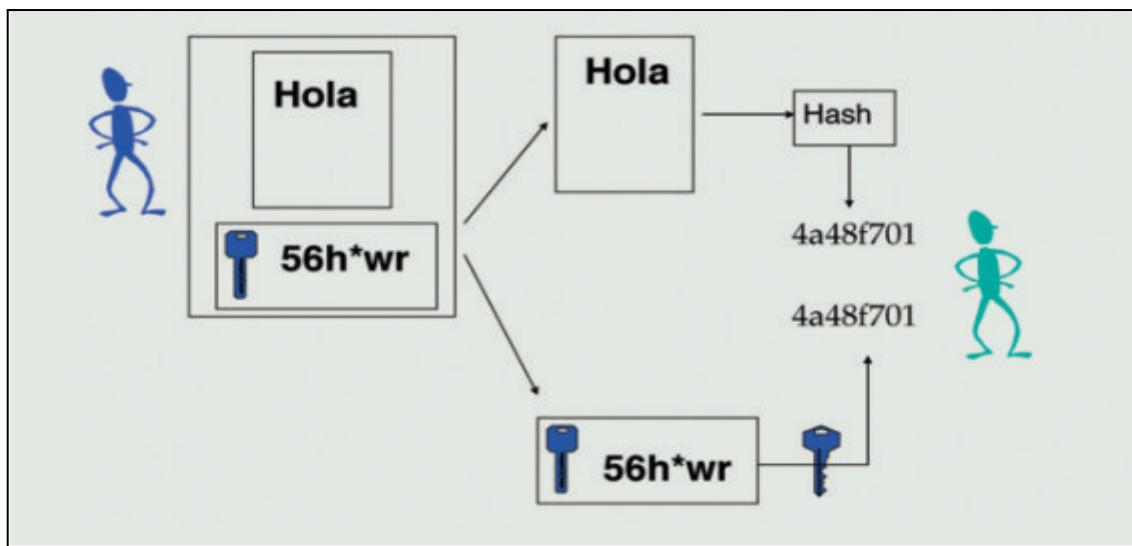
Si un usuari xifra una certa informació usant la seva clau privada, llavors qualsevol altra persona tindrà accés a aquesta informació mitjançant la clau pública de l'usuari. Atès que és impossible obtenir la clau privada a partir de la clau pública i que la clau privada es manté segura, es garanteix l'autoria (autenticació) i que ningú pot haver modificat el document (integritat).

Un document electrònic signat és equivalent a un document en paper i signat a mà que es publica en un tauler d'anuncis. La signatura electrònica és un dels aspectes més importants de la criptografia perquè permet dur a terme moltes

transaccions per Internet, evitant desplaçaments i pèrdues de temps. De fet, a Espanya, a la signatura electrònica aplicada sobre dades consignades en forma electrònica, se li atorga l'equivalència funcional amb la signatura manuscrita en virtut de la llei 59/2003, de 19 de desembre, de Signatura Electrònica.

Potser l'aplicació més coneguda a Espanya és la possibilitat de lliurar la declaració de la renda en format electrònic per Internet. Per a això l'usuari s'identifica davant el servidor web de l'Agència Tributària mitjançant el seu certificat digital, i després lliura el document electrònic de la declaració signat amb la seva clau privada. Un altre exemple són unes certes peticions que es fan mitjançant formularis en la web, que si se signen electrònicament té la mateixa validesa que una petició presencial.

El correu electrònic també pot beneficiar-se dels algorismes de signatura electrònica per a poder enviar missatges signats que té la mateixa validesa que una carta signada, i evitant els problemes de falsificació del remitent del correu. Finalment, un aspecte en el qual es va més a poc a poc són les transaccions bancàries, les quals guanyaria en nivell de seguretat si es fessin signades digitalment.



Il·lustració 27: Verificació de la firma d'un document públic.

En la pràctica no se sol xifrar tota la informació del document amb la clau privada, ja que resulta molt pesat computacionalment, sinó que resulta molt més eficient (tant per a l'emissor com per als receptors) obtenir un resum del document mitjançant algorismes HASH i després xifrar exclusivament el codi obtingut.

Finalment, s'envia el document, en principi sense xifrar, juntament amb uns codis de seguretat que representen la signatura. El receptor pot accedir al document perquè no està xifrat, i per a verificar integritat i autenticació fer les operacions sintetitzades en la il·lustració 27. El document enviat per l'usuari blau està format pel document original i per la signatura electrònica, que el HASH del document xifrat amb la seva clau privada.

Llavors el receptor separa el document de la signatura i calcula el HASH del document, d'una banda, i aconsegueix el HASH original utilitzant la clau pública sobre el codi de la signatura. Si el resultat que assoleix pels dos camins és el mateix significa que el document no ha estat alterat. Per a signar un document es poden aplicar qualsevol dels algorismes HASH i qualsevol dels algorismes de xifratge asimètric.

El més usat és aplicar MD5 o SHA1 com a funció de resum i després RSA com a algorisme asimètric per a xifrar aquest resum. L'alternativa és fer servir l'estàndard de signatura electrònica del NIST que es denomina DSA.

5.3. Certificats digitals

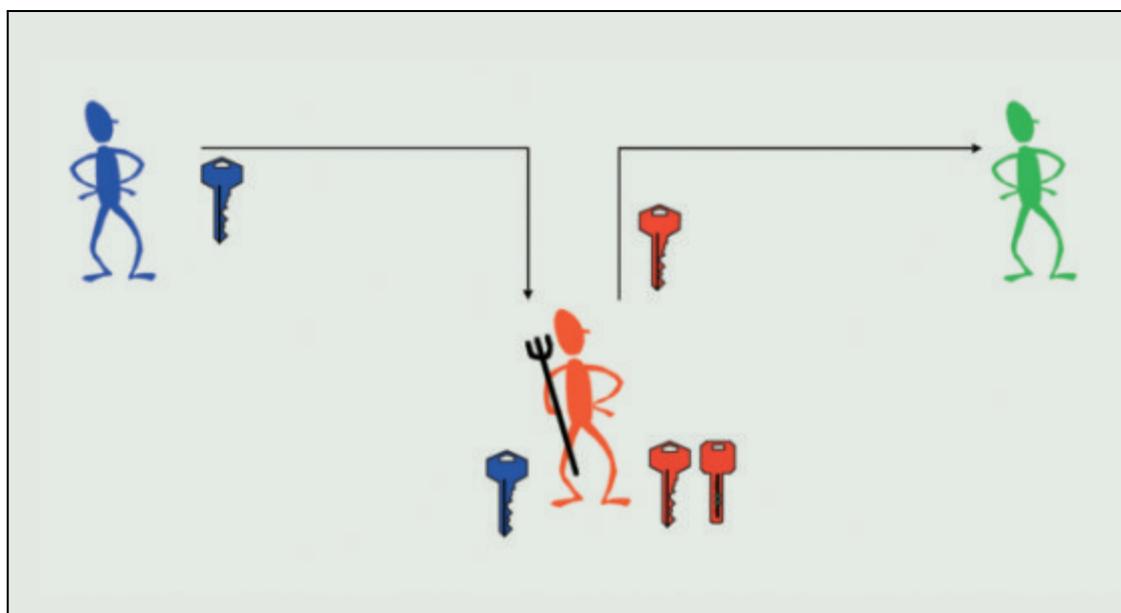
Els algorismes asimètrics, és a dir els que utilitzen una clau pública i una clau privada, van ser dissenyats per a poder intercanviar informació de manera segura sense necessitat d'haver acordat prèviament una clau secreta de xifratge. En efecte tot missatge xifrat amb la clau pública d'un usuari només pot ser desxifrat amb la clau privada que l'usuari manté segura.

No obstant això, existeix una vulnerabilitat en aquest procés que consisteix a suplantar la identitat del receptor en un mecanisme d'atac conegut amb el nom de

"man-in-the-middle attack" o bé MITM (vegeu II·lustració 28). Suposant que un atacant (representat en vermell) té capacitat d'interceptar la comunicació entre dos usuaris, l'atacant pot fer una substitució de claus que li donen capacitat per a veure i modificar els missatges sense que els usuaris siguin conscients de la intrusió.

Segons l'exemple de la il·lustració, l'usuari verd vol enviar un missatge confidencial a l'usuari blau, per la qual cosa reclama la seva clau pública. No obstant això, la clau pública de l'usuari blau és interceptada per l'atacant, que l'emmagatzema i la substitueix per una clau pública falsa (en vermell) que és enviada a l'usuari verd. Llavors els missatges xifrats per l'usuari verd amb la clau pública falsa són interceptats i desxifrats per l'atacant, que els pot veure i modificar abans de reexpedir-los a l'usuari blau xifratges amb la seva clau pública autèntica.

En aquest esquema l'usuari verd pensa que està utilitzant la clau pública del seu destinatari quan realment està utilitzant una clau pública falsa, i l'usuari blau rep un missatge perfectament xifrat amb la seva clau pública autèntica i no detecta que el missatge ha estat interceptat.



II·lustració 28: Intercepció de missatges.

Per a evitar aquest tipus d'atac, que faria perdre tota la utilitat dels algorismes asimètrics, és imprescindible que cada usuari tingui un mecanisme per a verificar si les claus públiques dels altres usuaris són reals o falses. Un mecanisme per a garantir l'autenticitat de les claus públiques és prendre-les de servidors de claus públiques en lloc de fer-ho directament del destinatari del missatge.

Aquest mecanisme dificulta considerablement l'atac "man in the middle", ja que a més d'interceptar la comunicació entre emissor i receptor caldria interceptar la comunicació entre l'emissor i el repositori de claus. Aquest últim canal de comunicació pot ser molt difícil de trencar, especialment si existeixen diversos servidors de claus que a més poden estar inclosos en la Intranet de l'emissor.

Un altre mecanisme per a reforçar la seguretat de les claus públiques és secundar-se en una tercera figura, independent de l'emissor i del receptor, que s'encarrega de signar electrònicament les claus. A aquesta nova figura se la denomina Entitat de Certificació i la seva funció principal és certificar que una clau és vàlida i pertany a una determinada persona.

La implementació pràctica d'aquest mecanisme es realitza guardant la clau pública juntament amb una certa informació addicional (nom de l'usuari, dates de validesa, número de sèrie...) dins d'un certificat digital segons l'estàndard X.509, i sol·licitar la signatura electrònica d'aquest certificat per part de l'entitat de certificació. L'entitat de certificació verifica les dades de la persona i llavors signa el certificat, és a dir, que calcula un resum del certificat mitjançant SHA1 o MD5 i el xifra amb la seva pròpia clau privada (que és l'actiu millor guardat de la companyia).

A més, l'entitat de certificació manté una llista de certificats que han estat revocats, és a dir certificats que han estat anul·lats a petició dels seus propietaris, per exemple en cas de detectar el robatori del certificat.

Un certificat digital que estigui signat per una entitat de certificació es pot verificar en qualsevol moment utilitzant la clau pública de l'entitat de certificació i consultant

les llistes de revocació. La majoria dels navegadors d'Internet i dels programes de correu incorporen de manera segura les claus públiques d'entitats de certificació de reconegut prestigi com VeriSign, Thawte, Entrust i FNMT (Fàbrica Nacional de Moneda i Timbre d'Espanya). En definitiva, la distribució de claus públiques per mitjà de certificats digitals signats és un mecanisme segur i que permet fer les següents operacions pràctiques:

- **Xifrar comunicacions**

L'aplicació més estesa és la comunicació segura amb servidors web (per exemple el banc) quan s'usa https. En aquest cas el navegador sol·licita un certificat digital al banc, i aquest ha de presentar un certificat signat per una de les entitats reconegudes pel navegador. Llavors s'estableix una comunicació xifrada asimètrica per a acordar una clau secreta i un algorisme de xifratge i a partir d'aquest moment s'estableix una comunicació xifrada simètrica.

- **Signar missatges i documents**

Si un usuari disposa d'un certificat digital instal·lat en el seu programa de correu, pot enviar missatges de correu signats electrònicament. Això permet als destinataris verificar l'autenticitat del correu i, per tant, s'obté la garantia que ningú ha suplantat el remitent de l'emissor. Degut a que suplantar el remitent del correu electrònic resulta especialment senzill a causa de la baixa seguretat dels protocols utilitzats, aquesta aplicació resulta especialment interessant.

- **Identificació davant un sistema o autenticació d'usuaris**

Encara se sol fer servir un nom d'usuari i una clau per a identificar-se davant molts sistemes informàtics, com per exemple el banc per Internet. Aquests noms d'usuari i contrasenyes en manera text són bastant vulnerables i la suplantació de l'usuari seria immediata. Alternativament, els servidors web poden exigir un certificat digital a l'usuari per a comprovar la seva identitat, la qual cosa resulta molt més segur. En combinar la capacitat de signar informació amb la possibilitat d'identificar usuaris de manera segura, apareix la possibilitat de realitzar

transaccions per Internet que tradicionalment es realitzaven de manera presencial.

Els certificats digitals podrien evitar molts dels problemes que es produeixen actualment de suplantació d'identitat per Internet. Hi ha dues tècniques, cada vegada més esteses, per a aconseguir obtenir el nom d'usuari i la clau d'accés d'una persona: enregistradors de teclat i phishing. Els primers són programes que s'autoinstal·len en l'ordinador (com si fossin un virus) i registren les pulsacions del teclat de l'usuari.

Atès que el nom d'usuari i la clau són cadenes de text, és senzill per a l'enregistrador de teclat detectar-les i enviar-les per Internet a l'atacant. La tècnica de phishing consisteix a enganyar l'usuari, normalment mitjançant un correu electrònic, perquè introduceixi les seves dades en una pàgina web. Aquesta pàgina ha estat creada per l'atacant imitant la pàgina principal d'un banc.

PART PRÀCTICA

Introducció

El nostre objectiu de la part pràctica del TDR era poder posar a prova el que hem après al llarg del procés de búsqueta d'informació per a la part teòrica i poder crear diferents programes d'encriptació posant en pràctica els diferents mètodes explicats al llarg d'aquest document. Tota la programació està guardada en un repositori de GitHub al qual pot accedir tothom.

https://github.com/rca諾esbri/tdr_cryptografia

Per a programar la part pràctica vam decidir utilitzar html per a la pàgina web, javascript per la encriptació i desencriptació i css per a l'aspecte de la pàgina web, a més a més, per a la programació amb javascript vam utilitzar la llibreria jquery i BigInt per a poder fer càlculs amb números grans, malauradament, tot i utilitzar aquesta llibreria, el nostre ordinador no ha estat capaç de fer els càlculs necessaris per a poder encriptar amb claus tan grans com les que s'utilitzen avui en dia per a encriptar i hem hagut d'utilitzar números més petits per a donar un exemple de com funciona la criptografia moderna.

La part pràctica està dividida en tres pràctiques de programació: El Xifratge Cèsar, el Xifratge Vigenère i l'encriptació Asimètrica amb RSA. I totes aquestes parts estan pujades a la pàgina web juntament amb la teoria.

1. Programació d'una pàgina web

1.1. Xifratge Cèsar

El primer mètode de encriptació que vam decidir posar en pràctica va ser el xifratge cèsar, ja que és un xifrat de substitució monoalfabètic i senzill d'entendre. La pàgina web té dos inputs de text, un per a la clau i un per al text que es vulgui encriptar o desencriptar. Al introduir el text, el programa el primer que fa és netejar la cadena de text per a assegurar-se que no hi han símbols estranys que no permetrien l'encriptació, com per exemple accents o símbols d'exclamació e interrogació.

```
function sanitizetext(str){
    const from = 'ÀÀÄÄÈÈÊÊÍÍÒÒÛÛÑÑ';
    const to = 'AAAAEEEIIIIOOOOUUUUNC';
    for (let i=0, l=from.length; i<l; i++) {
        str = str.replace(new RegExp(from.charAt(i), 'g'), to.charAt(i));
    }
    str.replace(/[^a-zA-Z0-9]/g, '');
    return str;
}
```

Il·lustració 29: Programació, funció sanitizetext.

Aquesta és la funció que s'encarrega de netejar el text. En el primer bucle “**for**” va lletra per lletra comprovant si alguna d'aquestes té accent (com a la constant **from**) i les reemplaça per les lletres sense accent (de la constant **to**). A l'acabar aquest bucle, el que fa és buscar a la cadena de text qualsevol símbol que no sigui una lletra entre la a i la z i els elimina.

Per a encriptar el text, creem un bucle que s'encarrega d'anar lletra per lletra encriptant el missatge. El missatge s'encripta passant tant el text sense encriptar com la clau a números i sumant-los. El missatge encriptat primer el guarda en una cadena de números (**ciphernum**) i després passa aquests números a una cadena de text (**ciphertext**) mitjançant la funció **getLetter**.

```
for (let i = 0; i < lengthplaintext; i++) {
    if (textsanitized.charCodeAt(i) >= 65 && textsanitized.charCodeAt(i) <= 90){
        ciphernum[i] = textsanitized.charCodeAt(i) - ciphersanitized.charCodeAt();
        ciphertext[i] = getLetter(ciphernum[i]);
        count++;
    } else {
        ciphertext[i] = ' ';
    }
}
```

Il·lustració 30: Encriptació Cèsar.

```
function getLetter(num){
    if (num>=26){
        num = num-26;
    } else if (num<0){
        num = num+26;
    }
    return String.fromCharCode(num + 65);
}
```

Il·lustració 31: Funció `getLetter` (per a passar de números a lletres).

La encriptació cèsar clàssica utilitza sempre la lletra C com a clau, és a dir que li sumaria tres a cada lletra, però nosaltres hem decidit que el nostre programa, en compte de tenir la clau ja fixa, escull l'usuari quina clau vol utilitzar.

Il·lustració 32: Pàgina Web Encriptació Cèsar.

1.2. Xifratge Vigenère

Una vegada vam tenir el xifratge Cèsar acabat, vam decidir fer el programa de xifratge Vigenère, que és molt semblant al Cèsar però que la clau, en compte de ser de una sola lletra, pot ser de més d'una lletra i llavors no tot el text s'encripta utilitzant el mateix número. Això fa que sigui un xifratge polialfabètic.

Com que el xifratge Vigenère és molt semblant al cèsar, el codi és gairebé idèntic, només vam haver de canviar la funció d'encriptació i desencriptació perquè, depenent de la posició del text en què es troba, utilitzés una lletra de la clau o una altra.

A l'hora d'agafar el número de la clau de la posició necessària, s'utilitza l'operació `count % lengthcipher` que agafa el mòdul de la divisió del número de posició de la lletra que volem encriptar dins de la paraula, entre el número de lletres que té la clau.

Per exemple: si volem encriptar la lletra que hi ha en 8a posició i tenim una clau de 3 lletres. El mòdul de la divisió ($8 \% 3$) donarà 2, per tant haurem d'utilitzar la segona lletra de la clau per a l'encriptació.

Amb aquesta operació ens assegurem que, sense importar el tamany del text a xifrar o al tamany de la clau, sempre s'agafa la lletra adequada per a l'encriptació.

```
for (let i = 0; i < lengthplaintext; i++) {
    if (textsanitized.charCodeAt(i) >= 65 && textsanitized.charCodeAt(i) <= 90){
        ciphernum[i] = textsanitized.charCodeAt(i) - ciphersanitized.charCodeAt(count % lengthcipher);
        ciphertext[i] = getLetter(ciphernum[i]);
        count++;
    } else {
        ciphertext[i] = ' ';
    }
}
```

Il·lustració 33: Programa encriptació vigenère.

A part d'aquesta operació, no hi ha cap altra diferencia notable entre el programa de xifratge cèsar i vigenère, ja que els dos encripten sumant el número de la lletra de la clau al del text xifrat.

1.3. Encriptació Asimètrica: RSA

Per a crear les claus del RSA primer hem de crear dos números primers aleatoris (p i q) mitjançant les funcions `randomnumber` i `isPrime`. La funció `randomnumber` s'encarrega de crear un nombre aleatori entre 1500 i 21500 i la funció `isPrime` s'encarrega de comprovar si aquest és un nombre primer, si no ho és, es busca un altre nombre fins que es troba un.

```

function calcularclaus(){
    let p = randomnumber();
    console.log(p);
    document.getElementById('p').innerText = p;
    let q = randomnumber();
    console.log(q);
    document.getElementById('q').innerText = q;
    let n = p*q;
    console.log(n);
    document.getElementById('n').innerText = n;
    let z = (p-1)*(q-1);
    console.log(z);
    document.getElementById('z').innerText = z;
    let k = findCoprime(z);
    console.log(k);
    document.getElementById('k').innerText = k;
    let j = findj(z, k);
    console.log(j);
    document.getElementById('j').innerText = j;
    claus = [BigInt(p), BigInt(q), BigInt(n), BigInt(z), BigInt(k), BigInt(j)];
    console.log('works');
}

```

Il·lustració 34: Creació de claus RSA.

```

function randomnumber(){
    let prime = false;
    while (prime === false){
        let num = Math.floor(Math.random() * 20000 + 1500);
        prime = isPrime(num);
        if (prime === true){
            console.log('works');
            return num;
            break;
        }
    }
}

```

Il·lustració 35: Creació d'un nombre aleatori.

```
function isPrime(num) {
    if (num <= 3) {
        return num > 1;
    }
    if ((num % 2 === 0) || (num % 3 === 0)) {
        return false;
    }
    let count = 5;
    while (Math.pow(count, 2) <= num) {
        if (num % count === 0 || num % (count + 2) === 0) {
            return false;
        }
        count += 6;
    }
    return true;
}
```

Il·lustració 36: Comprovar si un nombre és primer o no.

A continuació, per a trobar la clau n multipliquem $p*q$ i per a trobar z multipliquem $(p-1)*(q-1)$. Llavors hem de trobar k tal que k i z siguin nombres coprimeros, és a dir, que no tenen cap factor en comú. Per a trobar aquest nombre utilitzem la funció `factorizar` per a trobar els factors de k i després busquem un nombre que no sigui divisible per cap d'aquests factors.

```
function findCoprime(num) {
    let coprime=false;
    let factors = factorizar(num);
    let count = 1;
    let coPrime;
    while (coprime === false){
        let num2 = 16 + count;
        let length = factors.length;
        let strposition = 1;
        while (strposition <= length+1) {
            if (num2 % factors[strposition] === 0) {
                break;
            } else if (strposition === length){
                coprime = true;
                coPrime = num2;
            }
            strposition++;
        }
        count++;
    }
    return coPrime;
}
```

II·lustració 37: Trobar un nombre coprimer.

```
function factorizar(num) {
    let factors = [];
    if (num % 2 === 0){
        factors[1] = 2;
    }
    if (num % 3 === 0) {
        factors[2] = 3;
    }
    let count = 5;
    let strposition = 3;
    while (count <= num) {
        if (num % count === 0) {
            if (isPrime(count) === true){
                factors[strposition] = count;
                strposition++;
            }
        }
        if (num % (count + 2) === 0) {
            if (isPrime(count+2) === true){
                factors[strposition] = count + 2;
                strposition++;
            }
        }
        count += 6;
    }
    console.log(factors);
    return factors;
}
```

II·lustració 38: Factoritzar un nombre.

L'últim pas per a aconseguir totes les claus del RSA és trobar un nombre **j** enter tal que $j = (1+(x^z))/k$. Per a trobar-lo utilitzem la funció **findj**.

```
function findj(num1, num2) {
    let found = false;
    let z = num1;
    console.log(z);
    let k = num2;
    console.log(k);
    let x=1;
    while (found === false){
        let j = ((1+(x*z))/k);
        console.log(j % 1);
        console.log(j % 1 === 0);
        x++;
        if (j % 1 == 0){
            console.log(j);
            found = true;
            return j;
        }
    }
    return j;
}
```

Il·lustració 39: Funció per a trobar J.

Totes aquestes claus queden guardades en una **array** en la forma:
`claus = [p, q, n, z, k, j]` i per a poder accedir a elles quan les necessitem a l'hora d'encriptar i desencriptar.

Per a encriptar el text la funció **encriptar** eleva el text que volem encriptar a **k** (`claus[4] = k`) i després fa el mòdul de la divisió d'aquesta potència entre **n** (`claus[2] = n`). El resultat del mòdul és el text encriptat. Com que no funciona amb nombres molt grans, per a fer aquest exemple, el text s'encripta lletra per lletra, cosa que no és gens segura en la realitat, però funciona bé per a l'exemple.

```
function encriptar(){
    console.log(claus);
    cleartext = BigInt(document.getElementById('cleartext').value);
    ciphertext = BigInt(((cleartext**claus[4]) % claus[2]));
    console.log(ciphertext);
    document.getElementById('resultat').innerText = ciphertext;
}
```

Il·lustració 40: Funció encriptar RSA.

Per a desencriptar s'eleva el text que volem desencriptar a **j** (`claus[5] = j`) i després fa el mòdul de la divisió d'aquesta potència entre **n** (`claus[2] = n`). El resultat del mòdul és el text desencriptat.

```
function desencriptar(){
    console.log('botonfunciona');
    console.log(claus);
    console.log(ciphertext);
    cleartext = BigInt((ciphertext ** claus[5]) % claus[2]);
    let resultat = Number(cleartext);
    console.log(resultat);
    console.log('resultat');
    document.getElementById('resultat').innerText = resultat;
}
```

Il·lustració 41: Funció desencriptar RSA.

1.4. Pàgina Web

Un cop vam tenir fetes les pràctiques de programació vam decidir crear una pàgina web per a pujar totes aquestes pràctiques, juntament amb la teoria.

Teoria	Xifratge Cesar	Xifratge Vigénere	Encriptació Simètrica	Encriptació Asimètrica	Missatges
--------	----------------	-------------------	-----------------------	------------------------	-----------

TDR CRIPTOGRAFIA

1. Introducció

2. Història de la criptografia

3. Criptografia Clàssica

4. Criptografia Moderna

5. Aplicacions

1. Introducció

2. Història de la criptografia

Moltes vegades, sigui o bé per ignorància, o bé per desconeixement, tendim a pensar que la paraula criptografia va ligada a les missions d'espionatge internacional, les guerres, així com a organismes o entitats com la NASA o la Interpol. Però la realitat està molt més lluny d'això, ja que només cal pensar que quan ens connectem a serveis com Gmail o WhatsApp estem establint una comunicació segura, i, per tant, xifrada amb el nostre ordinador o dispositiu mòbil i els serveis informàtics. Per tal d'assegurar la nostra privacitat, els nostres missatges estan codificats mitjançant algorismes complexos que busquen precisament evitar la intrusió d'un agent extern a la xarxa, el qual podria posar en risc la seguretat de les nostres dades.

La necessitat de mantenir la informació a resguard d'ulls curiosos no és nova, és molt més antiga del que, potser, ens podem arribar a imaginar. Com a primer pas per a entendre com funciona la criptografia, fem un breu repàs als seus gairebé quatre mil anys d'història. La criptografia s'encarrega, precisament, de xifrar o codificar missatges per a evitar que el seu contingut pugui ser llegit per un tercer no autoritzat: és a dir, la generació de

[Descarregar PDF](#)

Il·lustració 42: Pàgina web.

Aquest és l'enllaç a la pàgina web: http://raul.cano-esbri.com/tdr_criptografia/

La pàgina està bloquejada per la WiFi de l'institut, per tant per a poder entrar-hi s'han de fer servir dades o una WiFi diferent.

2. Encriptació amb notes musicals

Inspirant-nos en el quadrat de Polibi, hem creat un mètode per a encriptar text passant-lo a música. Per a fer-ho hem creat aquesta taula, mitjançant la qual es pot encriptar un text a notes musicals en un pentagrama. Cada lletra s'encripta amb la seves coordenades dins del quadrat, per exemple: L = Sol \downarrow .

	DO	RE	MI	FA	SOL	LA	SI	DO
\circ	A	B	C	\textcircumflex	D	E	F	G
\downarrow	H	I	J	K	L	M	N	\tilde{N}
$\downarrow\downarrow$	O	P	Q	R	S	T	U	V
\uparrow	W	X	Y	Z	0	1	2	3
$\uparrow\uparrow$	4	5	6	7	8	9		

Il·lustració 43 Taula encriptació musical.

Si s'utilitza sempre la mateixa taula per a encriptar, cada lletra s'encriptarà sempre amb la mateixa nota resultant i el mètode no seria gens segur. Per a afegir alguna capa més de seguretat, hem pensat que per a encriptar es pot utilitzar una clau, formada per una lletra, un número ($4, 2, 1, \frac{1}{2}, \frac{1}{4}$) i una nota (do re mi fa sol la si).

Amb aquesta clau, qui encripta el missatge crea una taula per a xifrar-lo i el receptor pot crear una altra taula per a desencriptar el missatge.

- Primer: Les lletres es col·loquen en la taula, començant per la que s'ha donat amb la clau. ex: clau M = mnñopqrstuvwxyz0123456789abcçdefghijkl
- Segon: Segons el número que es dóna les notes de l'esquerra es posen en ordre, començant per la nota a la qual el seu temps equival al número de la clau. ex: clau $\frac{1}{2}$: $\uparrow\downarrow\circ\downarrow\downarrow$
- Tercer: Les notes de la part superior de la taula s'ordenen començant per la nota donada en la clau. ex: clau Sol = Sol La Si Do Do Re Mi Fa

ex: Clau = M ½ Sol

Taula Resultant:

	SOL	LA	SI	DO	DO	RE	MI	FA
♪	M	N	Ñ	O	P	Q	R	S
♩	T	U	V	W	X	Y	Z	0
○	1	2	3	4	5	6	7	8
♩	9	A	B	C	Ç	D	E	F
♪	G	H	I	J	K	L		

Il·lustració 44: Exemple taula encriptació musical. Clau: M ½ Sol.

Per a afegir encara un altra capa de seguretat, podem utilitzar claus de més d'una lletra que, com al Vigenère, es va repetint fins encriptar el text sencer. En aquest cas, si volem utilitzar una clau de tres lletres, tant l'emisor com el receptor haurien de crear tres taules, una amb cada lletra per a poder encriptar i desencriptar el missatge, però a canvi el resultat seria un xifrat força més segur, ja que no serviria amb un simple anàlisi de freqüència per a desencriptar el missatge si algú l'intercepta.

Per exemple, si la clau es DUH ½ Sol, s'hauran de crear tres taules, una en que les lletres comencen per la D, una en la qual les lletres comencen per la U i una última en la qual les lletres comencen per la H.

La primera lletra s'encripta utilitzant la primera taula, la segona lletra utilitzant la segona taula, la tercera lletra s'encripta utilitzant la tercera taula i llavors es van repetint successivament.

Ex: Clau = DUH ½ Sol

	SOL	LA	SI	DO	DO	RE	MI	FA
♪	D	E	F	G	H	I	J	K
♩	L	M	N	Ñ	O	P	Q	R
◦	S	T	U	V	W	X	Y	Z
♩	0	1	2	3	4	5	6	7
♪	8	9	A	B	C	Ç		

	SOL	LA	SI	DO3	DO4	RE	MI	FA
♪	U	V	W	X	Y	Z	0	1
♩	2	3	4	5	6	7	8	9
◦	A	B	C	Ç	D	E	F	G
♩	H	I	J	K	L	M	N	Ñ
♪	O	P	Q	R	S	T		

	SOL	LA	SI	DO	DO	RE	MI	FA
♪	H	I	J	K	L	M	N	Ñ
♩	O	P	Q	R	S	T	U	V
◦	W	X	Y	Z	0	1	2	3
♩	4	5	6	7	8	9	A	B
♪	C	Ç	D	E	F	G		

Il·lustració 45: Taules encriptació clau DUH ½ Sol.

Utilitzant aquesta clau (DUH ½ Sol) i les tres taules que hem creat a partir de la clau, si volem encriptar la paraula hauríem de utilitzar les taules en l'ordre següent:

	C	R	I	P	T	O	G	R	A	F	I	A
Taula:	1	2	3	1	2	3	1	2	3	1	2	3

Il·lustració 46: Correspondència lletra-taula encriptació musica.

A l'encriptar la paraula CRIPTOGRAFIA amb aquesta clau la partitura resultant seria:



Il·lustració 47: Resultat encriptació paraula CRIPTOGRAFIA

En aquesta partitura, cada nota representa una lletra de la paraula encriptada, mentre que els silencis no signifiquen absolutament res i cal ignorar-los a l'hora de desencriptar la paraula.

Per a desencriptar la paraula hem d'agafar cada nota i buscar-la a la taula que li correspongui per a trobar quina és la lletra que encripta.

<https://flat.io/score/>

Aquest és l'enllaç a la pàgina que hem utilitzat per a crear la partitura amb la paraula encriptada.

CONCLUSIONS

El nostre objectiu al llarg d'aquest treball era poder aprendre sobre criptografia i posar en pràctica aquests coneixements. Després de gairebé un any investigant sobre el tema hem ampliat enormement els nostres coneixements previs, que en un inici eren gairebé inexistentes.

Hem estudiat la història de la criptografia, des dels seus inicis fins a la segona guerra mundial i hem après com funcionen els mètodes de xifratge actuals que s'encarreguen que qualsevol búsqueta que fem a internet, qualsevol transacció bancària o qualsevol missatge que enviem estigui protegit.

A part d'aquesta recerca sobre el present i passat de la criptografia voliem investigar sobre com serà el futur de la criptografia, ja que erem conscients que amb l'aparició de nous tipus d'ordinadors, com l'ordinador quàntic, la criptografia, tal i com l'entenem avui en dia cambiaria totalment. Hem descobert que en efecte, l'aparició de ordinadors quàntics farà que quedin obsolets molts dels sistemes de encriptació que s'utilitzen avui en dia, però també servirà per a crear nous sistemes encara més segurs i sense les vulnerabilitats dels sistemes actuals.

Finalment, per a la part pràctica tenim dos objectius, el primer era crear un programa per a encriptar textos utilitzant algun dels mètodes que hem après al llarg de la investigació per a la part teòrica, i podem dir que ho hem aconseguit i hem sigut capaços de fer coses que a l'inici del treball no creiem possibles. Hem creat programes d'encriptació mitjançant el xifrat Vigenère, el Cèsar i fins i tot el RSA. Tot i que al crear el programa del RSA ens vam veure limitats pel fet de que per a crear un xifratge com el que s'acostuma a utilitzar s'han de fer servir nombres molt elevats, cosa el el nostre ordinador no ens permetia fer, per tant hem decidit crear un programa que funciona utilitzant els mateixos passos que fa servir el xifratge RSA però fent servir nombres més petits. El resultat ha sigut un programa que tot i que no es igual de segur és un exemple perfectament vàlid del seu funcionament.

Ademés vam tenir la idea d'agrupar tots aquests programes en una pàgina web juntament amb la teoria. Aquesta pàgina ara està disponible per a qualsevol. Aquesta és l'adreça: http://raul.cano-esbri.com/tdr_criptografia/

Per a innovar sobre el tema, vam decidir que volíem crear un mètode de xifratge propi, inspirant-nos en els conceptes aprenguts al llarg del treball, però que fos diferent a tots els altres tipus d'encriptació més populars. El resultat ha sigut un xifrat polialfabètic per a convertir text a notes musicals representades sobre un pentagrama. Malauradament, no vam poder crear un programa per a automatitzar el procés, però sí que hem pogut explicar el funcionament pas per pas i crear un exemple.

BIBLIOGRAFIA

Aumasson, J.P. (2017). *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press. [Enllaç Llibre](#)

Aumasson, J.P. (2021). *Crypto Dictionary: 500 Tasty Tidbits for the Curious Cryptographer*. No Starch Press. [Enllaç Llibre](#)

Menezes, A. J., van Oorschot, P. C. i Vanstone, S. A. (1996). *HANDBOOK OF APPLIED CRYPTOGRAPHY*. [Enllaç Llibre](#)

Barakat, M., Eder, C. i Hanke, T. (2018). *An Introduction to Cryptography*. [Enllaç Llibre](#)

WEBGRAFIA

Història de la Criptografia:

<https://www.neoteo.com/criptografia-la-maquina-enigma-y-la-segunda-guerra/>

https://en.wikipedia.org/wiki/Enigma_machine#Military_Enigma

<https://www.areatecnologia.com/maquina-enigma-alemana.htm>

<https://hipertextual.com/2011/07/la-maquina-enigma-el-sistema-de-cifrado-que-puso-en-jaque-a-europa>

<https://www.tutorialspoint.com/what-is-polyalphabetic-substitution-cipher-in-information-security>

<https://www.mub.eps.manchester.ac.uk/science-engineering/2018/11/28/cracking-stuff-how-turing-beat-the-enigma/>

Criptografia Clàssica:

<https://www.geeksforgeeks.org/difference-between-monoalphabetic-cipher-and-polyalphabetic-cipher/?ref=lbp>

Criptografia

<https://www.geeksforgeeks.org/vigenere-cipher/?ref=lbp>

<https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/?ref=lbp>

<https://www.geeksforgeeks.org/substitution-cipher/?ref=lbp>

<https://www.gaussianos.com/critpografia-cifrado-por-sustitucion/>

Criptografia Moderna:

<https://www.geeksforgeeks.org/difference-between-block-cipher-and-stream-cipher/?ref=lbp>

<https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/?ref=lbp>

<https://www.geeksforgeeks.org/rsa-algorithm-cryptography/?ref=lbp>

<https://www.hypr.com/security-encyclopedia/block-cipher>

<https://www.techtarget.com/searchsecurity/definition/block-cipher>

https://www.tutorialspoint.com/cryptography/block_cipher.htm

<http://clasespersonales.com/sisdis/paratranscribir1.pdf>

Futur de la Criptografia:

<https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm>

<https://quantum-computing.ibm.com/>

<https://www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art>

<https://www.hindawi.com/journals/scn/2018/8214619/>

<https://www.taylorfrancis.com/chapters/edit/10.4324/9780203360408-21/future-cryptography-dorothy-denning>

<https://newsroom.ibm.com/IBM-Explores-the-Future-of-Cryptography>

<https://www.techtarget.com/searchsecurity/definition/quantum-cryptography>

<https://quantumxc.com/blog/quantum-cryptography-explained/>

<https://www.zdnet.com/article/singapore-germany-to-mutually-recognise-iot-cyber-security-labels/>

<https://qiskit.org/textbook/ch-algorithms/shor.html>

<https://quantum-computing.ibm.com/composer/docs/iqx/manage/systems/cite>

<https://quantum-computing.ibm.com/docs/>

<https://arxiv.org/ftp/arxiv/papers/1501/1501.02365.pdf>

https://en.wikipedia.org/wiki/RSA_problem

https://en.wikipedia.org/wiki/RSA_Factoring_Challenge

<https://www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art>

<https://crypto.stackexchange.com/questions/42628/for-rsa-cryptography-how-long-does-it-take-to-factor-out-p-1024-if-given-q-1>

<https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm>

<https://www.quantiki.org/wiki/shors-factoring-algorithm>

<https://www.youtube.com/watch?v=lvTqbM5Dq4Q>

<https://www.youtube.com/watch?v=FA21Dj2I3Ac>

<https://medium.com/swlh/quantum-computing-for-dummies-part-1-2686b9ba3c51>

https://en.wikipedia.org/wiki/Quantum_computing

<https://www.freecodecamp.org/news/what-is-a-quantum-computer-explained-with-a-simple-example-b8f602035365/>

BANC D'IMATGES

II·lustració 1:

<https://elretohistorico.com/encryptacion-mensajes-secretos-espías-antiguedad-criptología/>

II·lustració 2: <https://es.wikipedia.org/wiki/Esc%C3%ADtala#/media/Archivo:Skytale.png>

II·lustració 3: <https://joseluistabaracarabajo.gitbooks.io/criptografia-clasica/content/Cripto07.html>

II·lustració 4: https://upload.wikimedia.org/wikipedia/commons/4/40/Jefferson%27s_disk_cipher.jpg

II·lustració 5:

[https://en.wikipedia.org/wiki/Enigma_machine#/media/File:Enigma_\(crittografia\) - Museo_scienza_e_tecnologia_Milano.jpg](https://en.wikipedia.org/wiki/Enigma_machine#/media/File:Enigma_(crittografia) - Museo_scienza_e_tecnologia_Milano.jpg)

II·lustració 6: <http://www.portierramaryaire.com/imagenes/enigma.html>

II·lustració 7: <https://hipertextual.com/2016/09/alan-turing-cancion-ordenador>

II·lustració 8:

<https://www.bbvaopenmind.com/tecnologia/innovacion/los-fallos-humanos-que-derrotaron-a-enigma/>

II·lustració 9: <https://culturizando.com/la-historia-del-genio-logro-descifrar-codigo-enigma/>

II·lustració 10:

<https://www.rfi.fr/es/economia/20220107-la-criptomoneda-bitcoin-sacudida-por-la-crisis-en-kazajist%C3%A1n>

II·lustració 11: <https://folou.co/internet/blockchain-que-es/>

II·lustració 12: <https://blockchainlanzarote.org/2020/10/12/criptación-nociones-avanzadas-1/>

II·lustració 13: Taula creada per nosaltres.

II·lustració 14: <https://joseluistabaracarabajo.gitbooks.io/criptografia-clasica/content/Cripto11.html>

II·lustració 15:

https://es.wikipedia.org/wiki/Archivo:Frecuencia_de_uso_de_letras_en_espa%C3%B1ol.svg

II·lustració 16: <http://clasespersonales.com/sisdis/paratranscribir1.pdf>

II·lustració 17: <http://clasespersonales.com/sisdis/paratranscribir1.pdf>

II·lustració 18: <http://clasespersonales.com/sisdis/paratranscribir1.pdf>

II·lustració 19: <https://es.acervolima.com/modos-de-operacion-de-block-cipher/>

II·lustració 20: <https://blog.isecauditors.com/2020/04/seguridad-ssl-tls-lucky13.html>

II·lustració 21: Taula creada per nosaltres.

II·lustració 22: <https://blog.isecauditors.com/2020/04/seguridad-ssl-tls-lucky13.html>

II·lustració 23: <https://www.economiasolidaria.org/noticias/gnulinux-el-triunfo-silencioso/>

II·lustració 24:

<https://www.edu.xunta.gal/centros/iesblancoamorculleredo/aulavirtual/mod/assign/view.php?id=25079>

II·lustració 25:

<https://www.ssl.com/es/preguntas-frecuentes/%C2%BFQu%C3%A9-es-la-criptograf%C3%A9tica-%C2%BFde-curva-el%C3%ADptica%3F/>

II·lustració 26:

<https://i2.wp.com/sefiks.com/wp-content/uploads/2018/12/twisted-edwards-curve.png?resize=515%2C379&ssl=1>

II·lustració 27:

https://www.iit.comillas.edu/documentacion/IIT-06-105R/Aplicaciones_pr%C3%A1cticas_de_la_cripograf%C3%A3DA.pdf

II·lustració 28:

https://www.iit.comillas.edu/documentacion/IIT-06-105R/Aplicaciones_pr%C3%A1cticas_de_la_cripograf%C3%A3DA.pdf

II·lustració 29: Programació

II·lustració 30: Programació

II·lustració 31: Programació

II·lustració 32: Pàgina Web creada per nosaltres

II·lustració 33: Programació

II·lustració 34: Programació

II·lustració 35: Programació

II·lustració 36: Programació

II·lustració 37: Programació

II·lustració 38: Programació

II·lustració 39: Programació

II·lustració 40: Programació

II·lustració 41: Programació

II·lustració 42: Pàgina Web creada per nosaltres

II·lustració 43: Taula creada per nosaltres (Encriptació Musical)

II·lustració 44: Taula creada per nosaltres (Encriptació Musical)

II·lustració 45: Taula creada per nosaltres (Encriptació Musical)

II·lustració 46: Taula creada per nosaltres (Encriptació Musical)

II·lustració 47: <https://flat.io/score/>

II·lustració 48: Taula creada per nosaltres

II·lustració 49: <https://www.gaussianos.com/critpografia-cifrado-por-sustitucion/>

II·lustració 50: Taula creada per nosaltres

II·lustració 51: https://blogg.sintef.no/wp-content/uploads/2019/03/shutterstock_1259122546.jpg

II·lustració 52: https://cdn-images-1.medium.com/max/800/1*kArIThgfMe-21Vr0j8wl7A.png

II·lustració 53:

https://assets.geoexpro.com/uploads/c9b22aa0-617c-46b8-b787-25b1e078fb7f/graph_800.jpg

ANNEXOS

ANNEX 1: TERMINOLOGIA

Per a poder seguir correctament el treball, us facilitem la lectura amb l'explicació de diferents termes que estan presents en aquest.

Xifrat per substitució: El xifrat de substitució és un mètode de xifrat pel qual unitats de text pla són substituïdes amb text xifrat seguint un sistema regular.

Criptoanàlisi: La criptoanàlisi és la branca de la criptografia que s'encarrega d'estudiar com vulnerar certs criptosistemes.

Anàlisi de freqüència: L'anàlisi de freqüència és un mètode de la criptoanàlisi el qual tracta d'estudiar la freqüència de lletres o números per a tal de poder desxifrar un missatge encriptat.

Funció Hash criptogràfic: Una funció criptogràfica hash- usualment coneguda com “hash”- és un algorisme matemàtic que transforma qualsevol bloc arbitrari de dades en una nova sèrie de caràcters amb una longitud fixa. Independentment de la longitud de les dades d'entrada el valor hash de sortida tindrà sempre la mateixa longitud.

Blockchain: La blockchain, o cadena de blocs, és una estructura formada per blocs on es recopila informació variada. Per a accedir a la informació d'un bloc de forma no autoritzada s'haurien de desencriptar tots els blocs anteriors a aquest.

Criptosistema: Un criptosistema és una estructura formada per un conjunt d'algoritmes amb la finalitat de poder encriptar o desencriptar elements.

Paritat (claus): Els bits de paritat d'un nombre en binari indiquen si aquest nombre és parell o imparell i serveix per a comprovar si el receptor del missatge l'ha rebut correctament o no.

Permutació: Variar o disposar l'ordre d'una seqüència d'elements.

Operació XOR: XOR és una porta lògica amb dos entrades en binari i una sortida. En la funció XOR, la sortida sera 1 si una i només una de les entrades és un 1, si son les dos 1 o les dos 0 la sortida sera 0.

Entrada 1	Entrada 2	Resultat XOR
0	0	0
0	1	1
1	0	1
1	1	0

Il·lustració 48: Porta Lògica XOR.

Xifratge en paral·lel: Amb xifratges senzills, els ordinadors realitzen les diferents operacions necessàries per a encriptar diferents parts del text simultàniament, però amb alguns tipus de xifratge concrets, com l'encriptació de blocs CBC no és possible, ja que per a encriptar un bloc necessita primer el resultat de encriptar el bloc anterior.

Grup Cíclic: Un grup cíclic és aquell que pot ésser generat per un sol element; és a dir, hi ha un element a del grup G (anomenat "generador" de G), tal que tot element de G pot ser expressat com una potència de a .

ANNEX 2: QUADRAT DE POLIBI

El xifrat amb quadrat de Polibi es va crear a l'antiga Grècia i originalment utilitzava l'alfabet grec, però s'ha adaptat per a utilitzar-lo amb la resta d'alfabets.

El quadrat de Polibi està format per una matriu, que acostuma a ser de 5x5 en la que es coloquen totes les lletres, i en les capçaleres de les columnes i files es col·loquen nombres o lletres preestablerts per l'emissor, aquests acostumen a ser els nombres de l'1 al 5.

1	2	3	4	5	
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Il·lustració 49: Quadrat de Polibi.

Per a encriptar, es substitueix cada lletra per els nombres que li corresponen per les coordenades. ex: M = 32

Hola quedaria encriptat com a 23343111

També es pot utilitzar una clau per a que sigui més segur. Aquesta clau el que fa és reordenar les lletres del quadrat, amb les lletres de la clau posades en les primeres posicions, i a continuació es posen la resta de lletres, sense repetir les que ja s'ha posat, en l'ordre de l'abecedari.

Per exemple, si utilitzem la paraula “Cipher” com a clau, el quadrat quedaria així:

	1	2	3	4	5
1	C	I (I,J)	P	H	E
2	R	A	B	D	F
3	G	K	L	M	N
4	O	Q	S	T	U
5	V	W	X	Y	Z

Il·lustració 50: Quadrat de Polibi clau “Cipher”.

ANNEX 3: FUTUR DE LA CRIPTOGRAFIA

1. Criptografia Quàntica

A l' hora de parlar de criptografia relacionada amb ordinadors quàntics, parlarem de dos tipus de criptografia:

- Criptografia Quàntica (Quantum Cryptography): Es refereix a una sèrie de mètodes d'encriptació que utilitzen tecnologia quàntica per a encriptar.
- Criptografia Post-Quàntica (Post-Quantum Cryptography): Fa referència als algoritmes (clau pública) d'encriptació que suposadament haurien de ser segurs davant d'un atac amb un ordinador quàntic.

La seguretat dels sistemes de criptografia de clau pública, com el RSA, es basa en la dificultat de factoritzar un nombre molt elevat. Com ja hem vist prèviament, les claus del xifratge RSA es generen a partir de dos nombres primer aleatoris molt elevats, **p** i **q**, els quals multipliquem per a trobar **n**, que forma part de clau pública, per tant és accessible a tothom. A partir de **p** i **q** es generen la resta de claus utilitzades a l' hora d'encriptar i desencriptar, per tant, si s'aconsegueix factoritzar **n** es trencaria per complert el xifrat. Per això, per a crear les claus s'utilitzen nombres molt elevats i factoritzar un nombre tan elevat trigaria moltíssim temps.

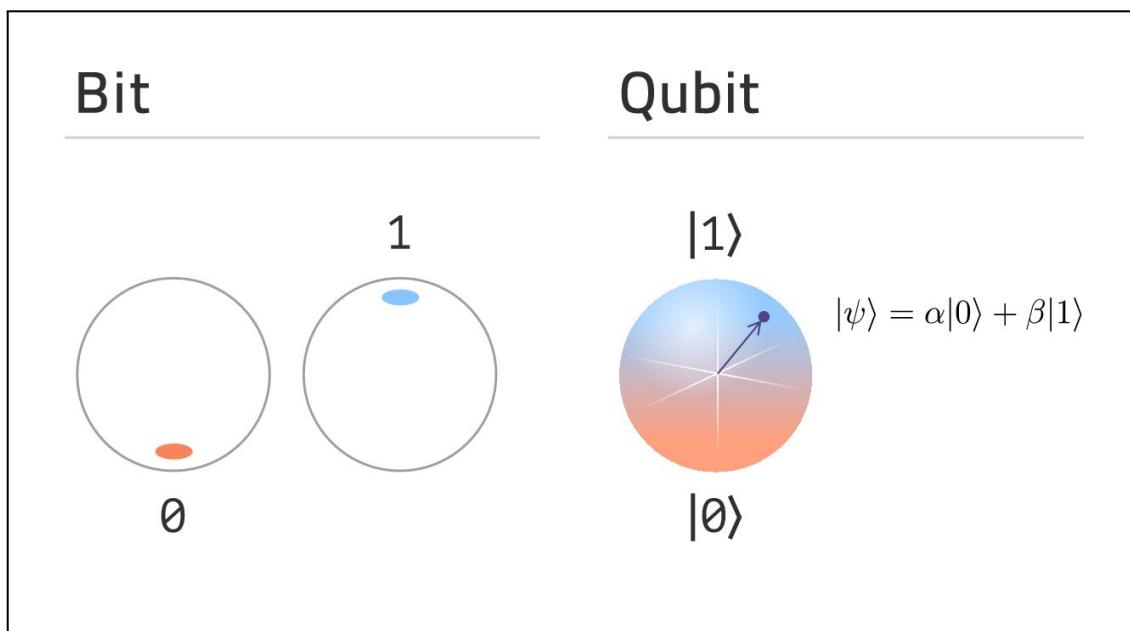
El tamany d'una clau de RSA acostuma a ser de 2048 o fins i tot 4096 bits és a dir que en decimal el nombre màxim que pot arribar a tenir és de $2^{2048}-1$ o $2^{4096}-1$. Amb nombres tan elevats, un ordinador normal triga moltíssim a trobar els factors i trencar el xifratge.

Actualment estem presenciant l'aparició d'un nou tipus d'ordinador, els ordinadors quàntics, que seran capaços de factoritzar aquests nombres enormes en un temps molt mes reduït i això vol dir que els xifratges de clau pública deixaran de ser segurs.

Un ordinador clàssic trigaria una quantitat enorme de temps a trencar una clau RSA de 2048 bits, és per això que ens podem sentir segurs amb aquest mètode d'encriptació, però un ordinador quàntic amb 4099 Qbits perfectament estables seria capaç de reduir aquest temps a tan sols 10 segons, fent que aquest mètode de xifratge sigui gairebé inútil.

El problema és que un ordinador com aquest encara no existeix, l'ordinador quàntic més potent que existeix a dia d'avui és l'ordinador Bristlecone de Google, el qual té només 72 Qbits i ademés tenen taxa d'error del 0,6%.

Dir que els ordinadors quàntics són un nou tipus d'ordinador no és del tot apropiat, és més aviat com dir que la bombeta és un nou tipus d'espelma. Al cap i a la fi, totes dues tenen la mateixa funció, la de iluminar, però la forma en que funcionen és completament diferent. Aquest és el cas amb els ordinadors i els ordinadors quàntics. Un ordinador clàssic funciona amb bits, els quals poden tenir un valor de 0 o de 1, mentre que un ordinador quàntic funciona amb Qbits, o quantum bits, els quals no tenen un valor definit, però tenen tots dos valors simultàneament.



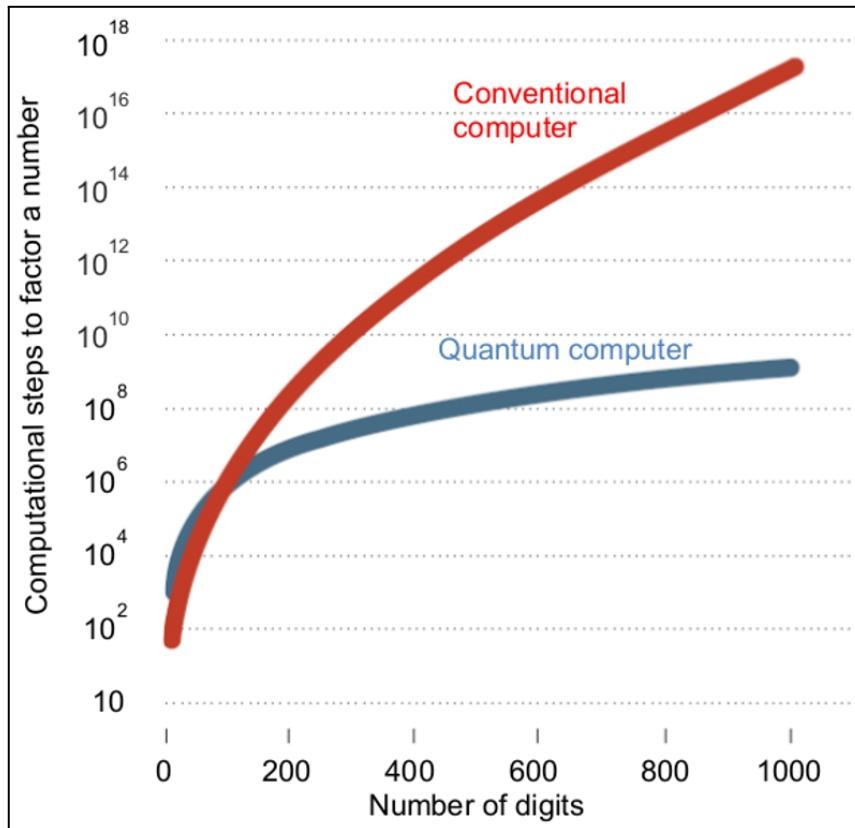
Il·lustració 51: Representació Qbit.

Això vol dir que a l'hora de fer una sèrie d'operacions complicades, un ordinador clàssic les ha de fer una per una fins a trobar la solució correcta, mentre que un ordinador quàntic pot fer múltiples operacions alhora i una vegada trobi el resultat es definirà el valor dels Qbits.

Això fa que els ordinadors quàntics siguin molt més ràpids alhora de fer operacions complexes en les que un ordinador normal trigaria molt, com per exemple a l'hora de calcular els factors d'un nombre enorme.

QUANTUM BITS (QUBITS)	EQUIVELANT CLASSICAL BITS
3	8
10	1024
20	1,048,576
...	...
300	2.037035976...E90

Il·lustració 52: Equivalent Qbits a bits.



Il·lustració 53: Operacions necessàries per factoritzar un nombre en funció del tamany.

Per a solucionar aquest problema s'estan creant nous mètodes d'encriptació, la seguretat dels quals no depengui en la factorització d'un nombre.

a. Criptografia Quàntica:

La criptografia quàntica és la ciència que es dedica a explotar les propietats mecàniques quàntiques per realitzar tasques criptogràfiques.

L'exemple més conegut de criptografia quàntica és la distribució quàntica de claus (Quantum Key Distribution) que ofereix una solució teòricament segura de la informació al problema de l'intercanvi de claus. És impossible copiar dades codificades en estat quàntic. Si s'intenta llegir les dades codificades, l'estat quàntic es canviarà a causa del col·lapse de la funció d'ona. Això es podria utilitzar per detectar espies en la distribució quàntica de claus (QKD).

Una propietat important i única de la distribució quàntica de claus és la capacitat dels dos usuaris que es comuniquen per detectar la presència de qualsevol tercer

que intenti conèixer la clau. Això resulta d'un aspecte fonamental de la mecànica quàntica: el procés de mesura d'un sistema quàntic en general pertorba el sistema. Un tercer que intenti conèixer la clau l'ha de mesurar d'alguna manera, introduint així anomalies detectables. Mitjançant l'ús de superposicions quàntiques o l'entrellat quàntic i la transmissió d'informació en estats quàntics, es pot implementar un sistema de comunicació que detecti l'escola i llavors es pot assegurar que la clau és segura i no ha sigut descoberta per ningú.

b. Criptografia Post-Quàntica:

La criptografia post-quàntica (**Post-quantum Cryptography**) (de vegades anomenada com **quantum-proof, quantum-safe or quantum-resistant**) es refereix a algorismes criptogràfics (normalment algorismes de clau pública) que es creu que són segurs contra un atac criptoanalític per part d'un ordinador quàntic. El problema amb els algorismes populars actualment és que la seva seguretat es basa en un dels tres problemes matemàtics difícils: el problema de **factorització de nombres enters**, el problema de **logaritmes discrets** o el problema de **logaritmes discrets de corba el·lítica**. Tots aquests problemes es podrien resoldre fàcilment en un ordinador quàntic prou potent amb l'algoritme de Shor.

Tot i que els ordinadors quàntics actuals no tenen la capacitat de processament per trencar qualsevol algorisme criptogràfic real, molts criptògrafs estan dissenyant nous algorismes per preparar-se per a un moment en què la computació quàntica es converteixi en una amenaça.