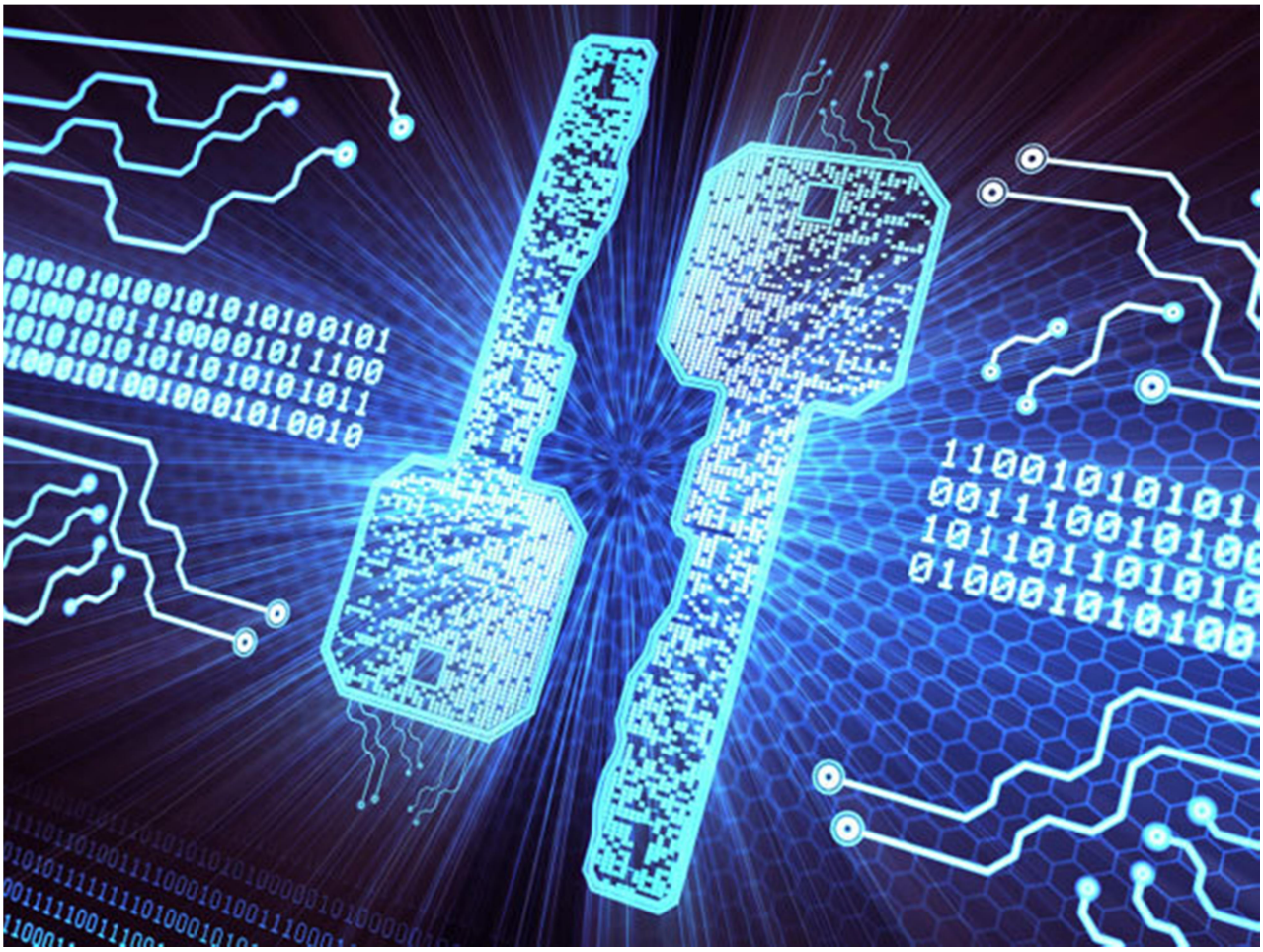
CR1PT0GRAF1A:

LA CLAU AL MÓN DELS SECRETS



Pseudònim: Euler

Índex:

Introducció

Des del primer moment en què vaig llegir la criptografia com a proposta per elaborar un treball de recerca que em va cridar l'atenció. Sempre havia tingut certa inquietud per saber com pot un missatge ser xifrat de tal manera que ningú en pugui esbrinar el contingut. A més a més, com que havia estat considerant estudiar matemàtiques a la universitat, vaig pensar que era la oportunitat perfecte per veure si realment m'interessaven. Principalment aquest fou el motiu pel qual vaig decidir que la criptografia seria el tema del meu treball de recerca.

L'objectiu que em proposo per aquest treball de recerca és el d'estudiar els mètodes criptogràfics més rellevants a nivell històric i comprendre el seu funcionament amb profunditat.

Per aconseguir-ho el mètode que he escollit per aquest treball és el següent: en primer lloc és necessària la comprensió dels conceptes que utilitza cada mètode criptogràfic. A continuació cal analitzar l'aplicació d'aquests conceptes al xifrat. Aleshores, utilitzaré els coneixements adquirits per a poder entendre els motius que els fan mètodes segurs o insegurs. Finalment comprovaré que sé aplicar tot allò après a través d'un exemple. I els tractaré en ordre cronològic, de més antics a més recents.

En primer lloc s'introduirà l'exemple estàndard de la criptografia. Aleshores seran tractats els mètodes criptogràfics següents: el xifrat cèsar, el polialfabètic, el bloc/quadern d'un sol ús, la màquina Enigma. A continuació s'aprofundirà en l'aritmètica modular per a facilitar la comprensió dels mètodes Diffie-Hellman, RSA i ElGamal. Llavors es farà un repàs de codificació per, finalment, analitzar les propostes criptogràfiques dels diners quàntics i el BB84.

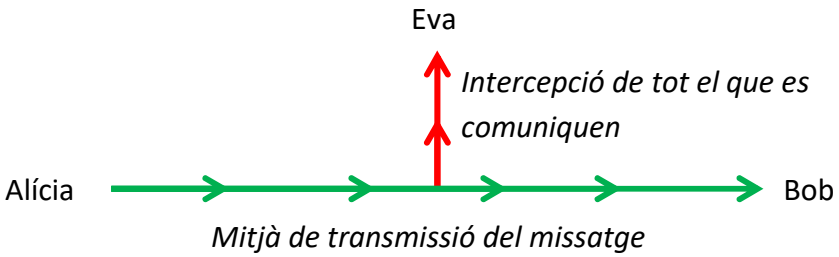
A més a més, per decidir què explicar i com explicar-ho, he seguit tres criteris: claredat, eficiència i rigor. És a dir, no es justificarà quelcom si la justificació és excessivament llarga pel paper que juga en el treball o si és quelcom que dificultaria en gran mesura la comprensió d'un concepte. També s'intentarà que tot allò mencionat en el treball sigui informació acceptada per la comunitat científica. Per aquests mateixos motius, tots aquells conceptes matemàtics que formen part de l'educació obligatòria, és a dir fins a 4t d'ESO, es donen per suposats. La resta seran definits i explicats en el mateix treball.

Abstract

First of all, this research project briefly introduces the standard example of Alice, Bob and Eve. In its second part the following ciphers are explained: Caesar, polyalphabetic and one-time pad. It goes on to describe the functioning of the Enigma Machine. In the following part, modular congruence is defined and its properties are demonstrated. Following this, the Diffie-Hellman algorithm, RSA and ElGamal are described and their security and viability are proved. Finally, after briefly describing the ASCII code and mentioning some of the properties of quantum physics, it illustrates the concept of quantic money and the BB84 cipher.

1. L'Alícia, en Bob i l'Eva

Des dels orígens de la criptografia, en general, quan es volia posar un exemple hipotètic és deia que la persona A enviava un missatge a la persona B. Això va canviar el 1978, quan l'equip conegut com RSA va publicar l'article "A method for obtaining digital signatures and public-key cryptosystems"[1] on posaven un exemple on la persona A es deia Alícia i la persona B, Bob. Deu anys després, Charles Bennet, Gilles Brassard i Jean-Marc Robert van publicar un article on van introduir el personatge de l'Eva, que sempre estava escoltant les comunicacions entre l'Alícia i en Bob. Li van posar aquest nom ja que la pronunciació de Eve en anglès és idèntica a la primera síl·laba de la paraula en anglès "eavesdropper". Que s'usa per referir-se a una persona que escolta converses alienes de forma secreta. Aquests tres noms s'han convertit en un estàndard i, per tant, seran els que usaré en tots els exemples d'aquest treball. L'esquema que es mostra a continuació pot ajudar a visualitzar els exemples que seran plantejats en els diversos xifrats:

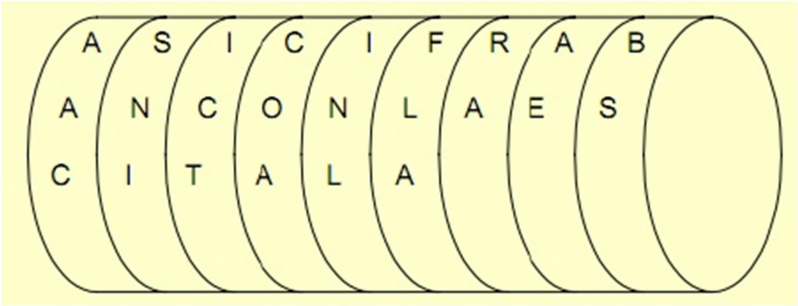


[1] En català seria "Un mètode per obtenir signatures digitals i criptosistemes de clau pública" i en aquest article els criptògrafs Rivest, Shamir i Adleman van proposar la criptografia de clau pública.

2. Història

2.1. Orígens

La criptografia va sorgir amb la guerra, a partir de la necessitat de comunicar-se entre aliats de forma que, encara que el missatge fos interceptat, l'enemic no en pogués obtenir informació. Els primers mètodes coneguts estaven basats en l'enginy. Per exemple, l'Escitala dels espartans (500 a.C) consistia en una tira de paper escrit que s'enrotllava a un cilindre d'un radi determinat per poder escriure el missatge. Per tant només amb un tub del mateix radi es podia desxifrar el missatge.



1. Dibuix d'una Escitala

Més endavant, concretament l'any 58 a.C, trobem el xifrat de Cèsar. Aquest mètode consisteix en substituir cada lletra del missatge per una que estigui *x* posicions més endavant. Això comporta que prèviament l'emissor i el receptor del missatge han de compartir un nombre que serà el que usaran per substituir les lletres. Usaré l'exemple següent per aclarir la idea:

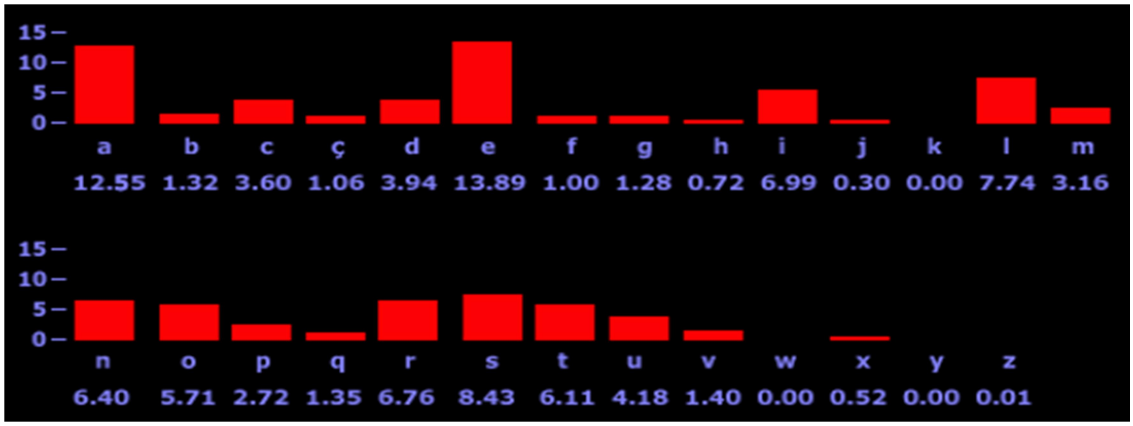
Posem que l'Alícia i en Bob han acordat que el nombre de posicions que desplaçaran l'abecedari és 3. Ergo substituiran les lletres que es mostren a dalt per les que són a sota en el següent esquema:

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

2. Esquema de l'exemple del xifrat Cèsar

Per tant, si l'Alícia vol enviar el missatge *Quedem el dissabte a les tres al bosc* a en Bob, el que realment enviarà serà *Txhghp ho glvvewh d ohv wuhv*. Quan en Bob rebi el missatge farà 3 passes enrere en l'abecedari per obtenir el vertader missatge.

Un punt feble d'aquest xifrat era que el podies resoldre amb el que s'anomena atac de força bruta. És a dir, simplement provar totes les possibilitats, en aquest cas només 26, i trobar la opció que té sentit. I tot i que aquest mètode va perdurar durant segles, aproximadament vuit-cents anys després el matemàtic àrab Al-kindi va trobar la forma de trencar aquest codi a través d'una propietat que té qualsevol llengua, que és la freqüència amb què cada lletra apareix en un text. Per exemple en el gràfic que apareix a continuació es pot veure la freqüència amb la que cada lletra apareix en qualsevol text en català.



3. Gràfic de freqüència de les lletres del català

Si tens aquesta informació, per desxifrar el missatge tot el que has de fer és elaborar un gràfic com el de la imatge però amb el missatge encriptat. D'aquesta forma pots simplement comparar les freqüències que cada lletra hauria de tenir amb les que té per desencriptar el missatge. Per exemple: si en el missatge encriptat un 12,6% de les lletres són *x*, pots deduir que les lletres *a* han estat substituïdes per lletres x i per tant, pots deduir el desplaçament aplicat a cada lletra. Si cada lletra de cada llengua té una freqüència amb la que sol aparèixer, podríem dir que tots els idiomes tenen com una mena d'empremta dactilar que els identifica.

2.2. Pas al xifrat polialfabètic

Un cop les febleses del xifrat Cèsar van quedar exposades, al segle XVI, Giovan Battista Belaso de Brescia va idear un nou xifrat, el xifrat vigènere. Aquest xifrat és polialfabètic i es basa en el mateix principi de substitució del xifrat Cèsar però, amb l'avantatge que difuminava l'empremta dels idiomes. Aquest mètode és com un xifrat Cèsar però, a més de partir d'un sol nombre, s'acorda prèviament una paraula i s'associa cada lletra de la paraula amb un nombre (la seva posició a l'alfabet per exemple). Després anem repetint la seqüència en el nostre missatge associant cada lletra del missatge amb un

nombre de la seqüència, que serà el nombre de desplaçaments que aplicarem a la lletra que té associada. Per aclarir la idea posaré un exemple:

Imaginem-nos que l'Alícia i en Bob han acordat prèviament que la paraula serà *objectiu*, ergo, la seqüència de nombres serà 15, 2, 10, 5, 3, 20, 9, 21. Si l'Alícia vol enviar el missatge *Quedem el dissabte a les tres al bosc* a en Bob, haurà d'aplicar el desplaçament d'acord amb la seqüència tal i com es mostra en el següent esquema:

Q	u	e	d	e	m	e	l	d	i	s	s	a	b	t	e	a	l	e	s
15	2	10	5	3	20	9	21	15	2	10	5	3	20	9	21	15	2	10	5

t	R	e	s	a	l	b	o	s	c
3	20	9	21	15	2	10	5	3	20

4. Esquema de l'exemple del xifrat polialfabètic

Per tant el missatge que enviarà serà *fwoihg ng skcxdvcz p nox wlnn pn ltvw*. Quan en Bob rebi el missatge, aplicarà la seqüència i farà el pas invers per desvelar el missatge.
D'aquesta forma es dissimulava l'empremta lingüística. Tot i això no l'esborrava del tot ja que hi seguia havent diferències en la freqüència amb la que apareixien les lletres. D'aquesta forma si l'Eva volia desencriptar el missatge, tot el que havia de fer era determinar la llargada de la seqüència. Un cop descoberta, ja simplement es tracta de desencriptar tants xifrats Cèsar com lletres té la paraula clau. Per tant la seguretat afegida del xifrat Cèsar és que s'ha de determinar la llargada de la seqüència a més a més de resoldre més d'un xifrat Cèsar.

2.3. Pas al bloc/quadern d'un sol ús

Durant uns 400 anys encara no s'havia trobat la solució al problema de l'empremta. Tot i això, cap a finals del segle XIX, Frank Miller va proposar un nou concepte per encriptar els missatges que era perfectament segura, coneguda com el bloc/quadern d'un sol ús.

Aquest mètode consisteix en compartir prèviament una seqüència de nombres generats de forma aleatòria que sigui tant llarga com el missatge en si i que només podran utilitzar un cop. A continuació s'aplica el desplaçament que li correspon a cada lletra i haurem obtingut el missatge xifrat. Per veure com funciona posaré un exemple: suposem que l'Alícia i en Bob comparteixen prèviament la següent seqüència de nombres que ha generat l'Alícia de forma aleatòria: 6, 13, 14, 9, 8, 22, 19, 8, 16, 16, 4, 23, 8, 1, 8, 4, 3, 13, 14, 13, 17, 4, 14, 11, 9, 6, 25, 25, 21, 4. Si l'Alícia vol enviar el missatge *Quedem el dissabte a les tres al bosc* a en Bob, haurà d'aplicar el desplaçament segons la seqüència tal i com es mostra en el següent esquema:

Q	u	e	d	e	m	e	l	d	i	s	s	a	b	t	e	a	l	e	s
6	13	14	9	8	22	19	8	16	16	4	23	8	1	8	4	3	13	14	13

t	r	E	s	a	l	b	o	s	c
17	4	14	11	9	6	25	25	21	4

5. Esquema de l'exemple del bloc/quadern d'un sol ús

Per tant, el que realment haurà d'enviar serà *wosmmi xt tywpicbi d ysf kvsd jr anng*. Quan en Bob rebi el missatge, aplicarà la seqüència fent el pas invers per desvelar el missatge.

2.4. Seguretat del bloc/quadern d'un sol ús

La seguretat del quadern d'un sol ús es coneix com el secret perfecte ja que cada una de les lletres de l'abecedari té la mateixa probabilitat d'aparèixer en el missatge. Això comporta que sigui impossible desencriptar el missatge.

Per fer-nos una idea d'aquest concepte, el compararé amb allò que és pràcticament segur, per exemple: quan posem una contrasenya al cadenat de la nostra bici, com pot ser 4296. Creiem que és segur perquè un lladre tardaria massa temps a provar totes les combinacions fins a donar amb la correcta. Per tant que tot i que el considerem segur no ho acaba de ser del tot. D'altra banda el quadern d'un sol ús és vertaderament segur ja que el xifrat no segueix cap mena de patró. És a dir, el missatge *ouñbeg lsrb* desencriptat té la mateixa probabilitat de ser *atacar nord* que *atacar oest*. I no hi ha cap criteri que puguem fer servir per discernir quin és el vertader missatge. Això comporta que qualsevol persona que vulgui desencriptar un missatge encriptat amb el quadern d'un sol ús, fins i tot si disposés de capacitat de computació infinita, el millor que pot fer és intentar endevinar-ho.

2.5. Inconvenients del bloc/quadern d'un sol ús

Tot i ser un xifrat indesxifrable, el bloc d'un sol ús no s'utilitza gairebé mai, ja que és molt poc pràctic. És a dir, tot i tenir una seguretat impenetrable a nivell teòric, a la pràctica presenta molts inconvenients.

En primer lloc generar nombres vertaderament aleatoris és una tasca veritablement complexa degut a què ni els humans ni els ordinadors som capaços de produir nombres aleatoris. Per una banda els ordinadors només són capaços de rebre una informació, realitzar un procés i emetre un resultat, per tant, si li proporciones la mateixa informació a un ordinador dues vegades, obtindràs el mateix resultat els dos cops. D'altra banda, els humans tendim a pensar que nombres com el 8263 són més aleatoris que, per exemple, el 3333.

En realitat, hi ha molts pocs fenòmens que es considerin vertaderament aleatoris, ja que, fins i tot tirar un dau no és vertaderament aleatori degut a què si tiréssim un dau exactament igual dues vegades, obtindríem el mateix resultat.

A això també li has d'afegir que la clau només pot ser usada un cop. Per tant, per cada missatge que l'Alícia volgués enviar a en Bob, no només hauria de generar aleatòriament una seqüència de nombres tant llarga com el missatge, sinó que també hauria de compartir-la prèviament amb en Bob abans de poder-se comunicar amb ell. Factor molt important en el món actual on l'Alícia i en Bob podrien estar a continents diferents.

Per tots aquests motius, el quadern d'un sol ús rarament s'utilitza. De fet, tal i com veurem més endavant en el treball, hi ha mètodes que, tot i no ser indesxifrables, són molt segurs i molt més pràctics.

3. Enigma

3.1. Introducció

Enigma va ser una màquina dissenyada per Arthur Schrebius, un enginyer alemany, que podia transcriure missatges xifrats. En un principi Arthur volia vendre aquestes màquines a empreses que estiguessin interessades en una comunicació segura. El 1923 va muntar Chiffriermaschinen Aktiengesellschaft (Corporació de màquines de xifrar) a Berlín per manufacturar el seu producte. Tres anys després la marina ja estava produint la seva pròpia versió i més tard també s'hi van afegir l'exèrcit (1928) i les forces aèries (1933). Va ser molt utilitzada durant la segona guerra mundial per l'Alemanya nazi. És més, es podria dir que Enigma era al centre de la seva estratègia bèl·lica ja que aquesta es basava en poder atacar per sorpresa a l'enemic. Per això era de vital importància que els Aliats no poguessin obtenir cap mena d'informació per anticipar-se als seus moviments.

3.2. La màquina Enigma

La màquina constava de les següents parts (el funcionament de les quals és explicat amb més precisió al següent apartat):

- **Teclat:** per escriure.

- **Taula d'endolls:** cada lletra tenia associat un endoll. Si connectaves els endolls de dues lletres mitjançant un cable podies fer que els camins de dues lletres s'intercanviessin. És a dir, feia que, per exemple, una *W* sortís de la taula com una *D* i que una *D* sortís de la taula com una *W*. Com que hi havia deu cables, podies intercanviar els camins de deu parells de lletres.

- **Rotors:** peça giratòria que funciona com un comptador mecànic. Cada una de les 26 lletres de l'abecedari tenien un contacte d'entrada i un de sortida a cada rotor. A dins de cada rotor hi havia cables que conduïen la senyal elèctrica des del contacte d'entrada d'una lletra, per exemple la *M*, fins a el de sortida d'una altra, per exemple la *G*. D'aquesta manera, la senyal elèctrica que entrava en un rotor com una *M* sortia del mateix rotor com una *G*. Hi havia un total de 5 rotors diferents dels quals se n'havien d'escollir 3 per col·locar a la màquina.

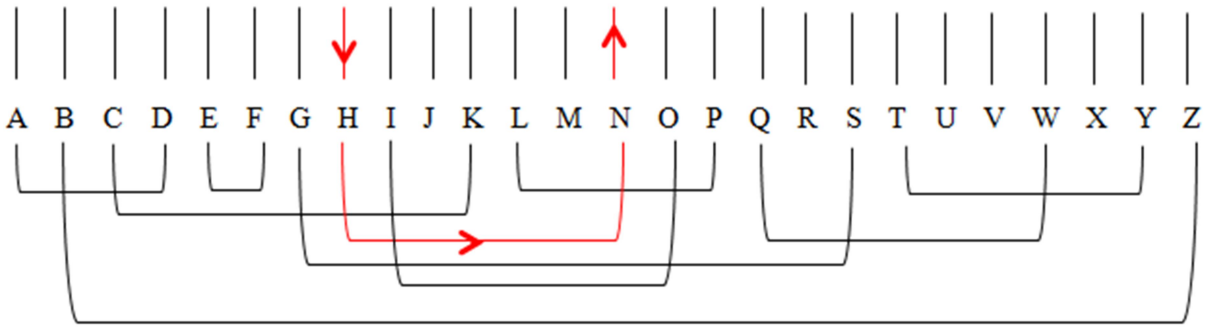
- **Reflector:** peça que es col·locava al costat de l'últim rotor. La seva funció era la de redirigir les senyals que sortien de l'últim rotor a aquest mateix.

- **Llums:** cada llum tenia una lletra associada que s'il·luminava quan en premies una al teclat. La lletra que s'il·luminava era o bé la encriptada o la desencriptada.

6. Fotografia d'una màquina Enigma

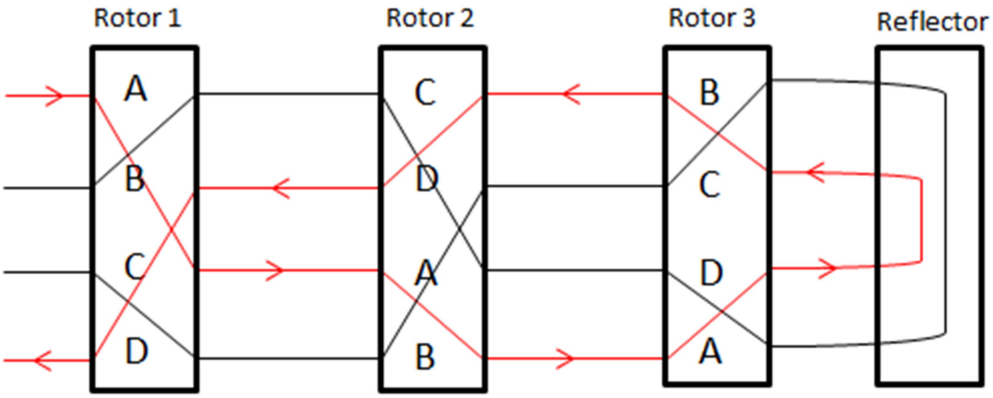
3.3. Funcionament

Quan premies una tecla, una senyal elèctrica sortia d'aquesta i passava per la taula d'endolls, després pels rotors, aleshores, com que el reflector connectava l'últim rotor amb ell mateix, tornava a passar pels rotors, tornava a la taula d'endolls i s'acabava il·luminant la lletra encriptada o desencriptada. A continuació explicaré detalladament quin era el procés per encriptar, que és el mateix que el procés per desencriptar.
En primer i últim lloc la senyal elèctrica passava per la de la taula dels endolls on podies fer que deu parells de lletres intercanviessin els camins. A l'esquema que apareix a sota, el camí vermell seria el recorregut que realitzaria una senyal elèctrica si preméssim la *H*.



7. Esquema de la taula d'endolls d'Enigma

La segona fase és el conjunt dels rotors. Per explicar-ho de forma més aclaridora ho faré a partir d'un esquema simplificat, és a dir, l'esquema representaria els rotors d'Enigma si només hi haguessin 4 lletres.



8. Esquema dels rotors d'Enigma

El camí vermell representa el recorregut que realitzaria una senyal elèctrica. Com podem veure quan premem una lletra qualsevol, aquesta seguirà un recorregut únic que l'encriptarà a una altra lletra. També hem de tenir en compte que quan premies una tecla, el rotor de més a l'esquerre en aquest esquema girava una posició. Per tant que, en aquest esquema, la seqüència que es representaria pel rotor esquerre després de prémer una tecla seria, de dalt a baix, *DABC*. Com podem observar, encara que teclegéssim la mateixa lletra dues vegades seguides, la senyal elèctrica recorreria un camí completament nou que la portaria a encriptar la mateixa lletra a una altra completament diferent.

Per aclarir la idea posaré un exemple hipotètic de com s'encriptaria una lletra qualsevol: imaginem-nos que premem la lletra *A.* Hi ha una senyal elèctrica que surt d'aquesta tecla i es dirigeix a la taula d'endolls on hi ha un cable que redirigeix aquesta senyal cap al camí de la *W*, ergo la lletra encriptada en aquest moment és una *W*. A continuació aquesta senyal entra pel port de la *W* al primer rotor, però surt del mateix rotor com una *K* perquè els cables dins dels rotors estan barrejats. Després entra al segon rotor com a una *Y* perquè els rotors estan girats en diferents posicions. Llavors surt del segon rotor com una *D* i així successivament amb els quatre rotors que tenia el model final. Finalment la senyal surt del quart rotor com una *P*, i torna a entrar a aquest per l'acció del reflector com una *F*. I torna a passar per tots els rotors i la taula d'endolls, d'on surt com una *M*, per tant el cable que surt de la *M* farà que s'il·lumini el llum d'aquesta per indicar que és la lletra encriptada. Per desxifrar aquesta *M* tot el que necessites és prémer la *M* en una màquina enigma idèntica que parteixi de la mateixa posició inicial. El disseny d'Enigma (la forma en la que els cables estan barrejats) fa que, a través del mateix procés descrit abans, t'il·lumini la *A*, que és la lletra original.

3.4. Seguretat

3.4.1. Concepte

Enigma va ser dissenyat per intentar aproximar-se a la aleatorietat creant una seqüència molt basta. És a dir, tot i que els nombres seguien un patró, aquest tardava tant a repetir-se que era com si fossin aleatoris. Seria com xifrar un missatge de 50 lletres utilitzant el mètode polialfabètic amb una paraula de vint-mil lletres; a l'Eva li seria impossible determinar el patró perquè és tan llarg que no es repeteix. D'aquesta forma sembla una seqüència vertaderament aleatòria. Si això era automatitzat, com que qualsevol màquina comença en una posició inicial realitza una operació i emet un resultat, tant l'emissor del missatge com el receptor havien de tenir una màquina idèntica i col·locar-la en la mateixa posició per poder-se comunicar. Si tenim això en compte es podria pensar que amb una màquina idèntica, qualsevol podria desencriptar els missatges. Però això no era possible degut a l'enorme quantitat de posicions inicials que hi havia.

3.4.2. Càlcul del nombre de posicions inicials possibles

La màquina Enigma tenia dues parts que havien de configurar els operadors: els rotors i la taula d'endolls. Ergo per calcular la quantitat de posicions inicials hem de calcular les formes possibles que hi havia de col·locar cada part i multiplicar-les.

En primer lloc hi havia 5 rotors diferents d'entre els quals n'havies de triar 3. Per tant, primerament en triaves un d'entre 5, després un d'entre 4 i finalment un d'entre 3. Aleshores hi havia $5 \cdot 4 \cdot 3 = 60$ formes d'ordenar-los. Però també hem de tenir en compte que cada rotor es pot col·locar en una posició d'entre 26, això fa que les posicions possibles dels tres rotors fossin $26 \cdot 26 \cdot 26 = 17.576$. A més a més hi havia la taula d'endolls que per calcular les formes que hi havia d'ordenar-la necessitem fer diversos passos: en primer lloc hem de tenir en compte que hi ha $26!$ (factorial de 26) formes d'ordenar l'abecedari. Però com que només hi ha 10 cables que connecten 10 parelles de lletres, hi ha 6 lletres que no reordenarem, dividim entre 6!. Com que els cables són idèntics l'ordre en què estiguin és irrellevant dividim entre 10!. *A* continuació, com que estem intercanviant lletres, no és important si connectem la A amb la *B* o la *B* amb la *A*, dividim entre 2 deu vegades, que seria 2^{10}. Finalment realitzem la operació que s'acaba de descriure per saber les formes possibles d'ordenar els cables $\frac{26!}{6! \cdot 10! \cdot 2^{10}} = 150.738.274.937.250.$

Un cop ja tenim els tres resultats, simplement els multipliquem per obtenir el nombre de possibles posicions inicials de la màquina enigma, que seria $60 \cdot 17.576 \cdot 150.738.274.937.250 = 158.962.555.217.826.360.000$. Com podem observar, la quantitat de posicions inicials possibles era tant immensa, que feia pràcticament impossible determinar en quina d'aquestes s'havien començat a comunicar els alemanys.

3.4.2. Punt feble

El gran punt feble que va portar al desxiframent d'Enigma és el reflector ja que impedia que una lletra s'encriptés a ella mateixa. Per explicar el perquè posaré un exemple: suposem que premem la lletra *A* i arriba als rotors com una *W*. Perquè la lletra encriptada sigui l'*A*, la senyal hauria de sortir dels rotors com una *W*. Això era impossible ja que quan arribés a l'últim rotor, hauria de fer el mateix recorregut per tornar a sortir com una *W*. I com que el reflector intercanviava camins entre lletres, això era impossible. Això va ajudar a The bombe (el super-ordinador aliat) a descartar moltes possibilitats i, en definitiva, a trencar el codi d'Enigma.

4. Aritmètica modular

4.1. Conceptes previs

4.1.1. Divisió euclidiana

La divisió Euclidiana és aquella divisió d'enters en què, com a resultat, obtenim un quocient q i un residu r menor al divisor, on q i r també són enters. Si tenim això en compte, una divisió euclidiana, on a sigui el dividend i n el divisor, la podem escriure com una igualtat amb el següent format: $a, n, q, r \in \mathbb{Z}, a = n \cdot q + r$.

Quan dividim un nombre negatiu entre un nombre natural, podem seguir dues convencions: en una el residu ens dona negatiu i en l'altra positiu. Per tant, per exemple la divisió euclidiana de $-8 \div 5$, la podem escriure com dues igualtats diferents que són les següents:

$$-8 = 5 \cdot (-1) - 3$$
$$-8 = 5 \cdot (-2) + 2$$

4.1.2. Relació d'equivalència

En aquest apartat usaré la notació $a \sim b$ per dir que un element a està relacionat amb un element b.

Definició: Diem que una relació és d'equivalència quan en un conjunt A compleix les següents propietats:

- Reflexivitat: per qualsevol $a \in A$, $a \sim a$.
- Simètrica: per qualsevol $a, b \in A$, si $a \sim b$, llavors $b \sim a$.
- Transitiva: per qualsevol $a, b, c \in A$, si $a \sim b$ i $b \sim c$, llavors $a \sim c$.

4.1.3. Classe d'equivalència

Un cop hem definit relació d'equivalència i esmentat les propietats que ha de satisfer, podem definir què és una classe d'equivalència.

Definició: Sigui A un conjunt en el que hi ha definida una relació d'equivalència $\sim$. Una classe d'equivalència és el conjunt d'elements de A que estan relacionats entre sí.

Exemple: Suposem que treballem amb una relació d'igualtat entre totes les edats de tots els humans del món. A serà el conjunt format per tots els humans, així doncs, podríem dir que totes les persones que tenen 20 anys pertanyen a la mateixa classe d'equivalència.

Freqüentment, quan treballem amb classes d'equivalència escollim un dels elements de cada classe i l'anomenem representant. Aquest representant pot ser escollit segons diferents criteris, per exemple: el més petit, el més gran...

4.2. Congruència modular

La notació que utilitzaré en aquest apartat serà la següent: $a| b$ voldrà dir que a és un divisor de b.

Un exemple d'una relació d'equivalència és la congruència modular. Aquesta es defineix a continuació.

Definició: Siguin a, b i n nombres enters, direm que a és congruent amb b mòdul n ($a \equiv b\ (mod\ n)$) si i només si $n \mid (a - b)$.

Tenint això en compte, si $a \equiv b\ (mod\ n)$ aleshores existeix un nombre enter k tal que $a - b = n \cdot k$, que podríem reordenar com $a = b + n \cdot k$. Aquesta segona versió serà molt útil en vàries demostracions que segueixen.

Exemple:
$34 \equiv 7\ (mod\ 3)$ perquè $(34 - 7) \div 3 = 9$, que és un nombre enter, per tant podem dir que $(34 - 7)$ és divisible entre 3. Com que aquesta és la definició de congruència modular podem dir que $34 \equiv 7\ (mod\ 3)$.

Propietat: La relació de congruència modular és una relació d'equivalència.

Demostració:
Per tal de demostrar que en efecte és una relació d'equivalència, hem de demostrar que satisfà les tres propietats mencionades en l'apartat *4.1.4. Relació d'equivalència.*

- Reflexiva ($a \equiv a\ (mod\ n)$)
  Per definició, $a \equiv a\ (mod\ n)$ si i només si $n \mid a - a$. Tenint en compte que si a un nombre li restes el mateix nombre donarà zero, obtindrem $n \mid 0$ que ens confirma la propietat de la reflexivitat ja que zero és divisible entre qualsevol enter.
- Simètrica (Si $a \equiv b\ (mod\ n)$, aleshores $b \equiv a\ (mod\ n)$)
  Partim de $n \mid a - b$. Com que sabem que en multiplicar un nombre divisible per n per qualsevol enter, el producte seguirà sent divisible entre n, podem dir que $n\ |(-1) \cdot (a - b)$. Si realitzem la operació ens queda $n \mid b - a$, que per definició és $b \equiv a\ (mod\ n)$.

- Transitiva (Si $a \equiv b\ (mod\ n)$ i $b \equiv c\ (mod\ n)$, aleshores $a \equiv c\ (mod\ n)$
  Partim de $n \mid a - b$ i $n \mid b - c$. Com que sabem que la suma de dos nombres divisibles entre n, és divisible entre n, podem dir que $n \mid (a - b) + (b - c)$. Si realitzem la operació ens queda que $n \mid a - c$. Que per definició vol dir $a \equiv c\ (mod\ n)$.

4.3. Propietats de la congruència modular

Tots els nombres esmentats en aquest apartat són enters. Amb l'excepció del nombre k esmentat a la propietat 5, que és natural.

A més a més de ser una relació d'equivalència, la congruència modular té una sèrie de propietats, que són les següents:

Addició
Si $a \equiv b\ (mod\ n)$ i $c \equiv d\ (mod\ n)$, aleshores $a + c \equiv b + d\ (mod\ n)$

Demostració:
$a \equiv b\ (mod\ n)$ i $c \equiv d(mod\ n)$ per definició vol dir que $a = b + n \cdot k_1$ i $c = d \cdot n \cdot k_2$. Després ho plantegem com un sistema d'equacions i sumem les dues equacions.

$$\left.\begin{array}{l} a = b + n \cdot k_1 \\ c = d + n \cdot k_2 \end{array}\right\} \quad a + c = b + n \cdot k_1 + d + n \cdot k_2$$

Si ara extraiem factor comú n i ho reordenem ens queda:

$$a + c = b + d + n \cdot (k_1 + k_2)$$

Que per definició és $a + c \equiv b + d\ (mod\ n)$.

Multiplicitat
Si $a \equiv b\ (mod\ n)$ i $c \equiv d\ (mod\ n)$, aleshores $ac \equiv bd\ (mod\ n)$.

Demostració:
$a \equiv b\ (mod\ n)$ i $c \equiv d\ (mod\ n)$ per definició és $a = b + n \times k_1$ i $c = d + n \times k_2$. Un cop tenim les dues equacions plantejades, multipliquem a per c substituint la a i la c pel que són equivalents respectivament.

$$\left.\begin{array}{l} a = b + n \cdot k_1 \\ c = d + n \cdot k_2 \end{array}\right\} \quad ac = (b + n \cdot k_1) \times (d + n \cdot k_2)$$

Ara fem la multiplicació i extraiem factor comú n:

$$ac = bd + b \cdot n \cdot k_1 + n \cdot k_1 \cdot n \cdot k_2 + d \cdot n \cdot k_2$$
$$ac = bd + n \cdot (b \cdot k_2 + k_1 \cdot n \cdot k_2 + d \cdot k_1)$$

Si observem l'expressió final veurem que per definició vol dir que $ac \equiv bd \ (mod\ n)$.

Propietat 3

Si $a \equiv b \ (mod\ n)$, aleshores $ak \equiv bk \ (mod\ n)$. I, sempre i quan n i k siguin coprimers, si $ak \equiv bk \ (mod\ n)$, aleshores $a \equiv b \ (mod\ n)$.

Demostració:

$a \equiv b \ (mod\ n)$ per definició és $n \mid (a - b)$. Si tenim en compte que si multipliquem un nombre divisible entre n per qualsevol enter el resultat seguirà sent divisible entre n, podem fer $n \mid (a - b)k$. Si apliquem la propietat distributiva ens queda $n \mid ak - bk$, que per definició és $ak \equiv bk \ (mod\ n)$. En el cas que n i k comparteixin algun factor, no podem aplicar la propietat en el sentit contrari ja que això implica que la següent expressió $n \mid (a - b)k$ pot ser certa indiferentment dels valors de a i b degut a què k pot aportar els factors necessaris per tal que el conjunt $(a - b)k$ sigui divisible entre n.

Propietat 4

Si $a \equiv b \ (mod\ n)$, aleshores $a + k \equiv b + k \ (mod\ n)$. I, si $a + k \equiv b + k \ (mod\ n)$, aleshores $a \equiv b \ (mod\ n)$.

Demostració:

$a \equiv b \ (mod\ n)$ per definició és $n \mid (a - b)$. Com que un nombre menys ell mateix dona 0, podem sumar i restar un nombre k i ens queda $n \mid (a - b) + k - k$, si ho agrupem seria $n \mid (a + k) - (b + k)$ que per definició és $a + k \equiv b + k \ (mod\ n)$.

Propietat 5

Si $a \equiv b \ (mod\ n)$, aleshores $a^k \equiv b^k \ (mod\ n)$.

Demostració:

Aquesta propietat es demostrarà amb una prova per inducció, és a dir, primerament es demostrarà un cas base on $k = 1$ i a continuació es demostrarà que si és vàlid per un valor $k - 1$, això implica que és vàlid per un valor k.

Si $k = 1$, aleshores $a \equiv b \ (mod\ n) \Rightarrow a^1 \equiv b^1 \ (mod\ n)$. Un cop demostrat el cas base fem el pas inductiu. Partim de que el resultat és cert per $k - 1$: si $a \equiv b \ (mod\ n)$, aleshores $a^{k-1} \equiv b^{k-1} \ (mod\ n)$ i podem aplicar la propietat multiplicativa:

$$\left.\begin{array}{l} a^{k-1} \equiv b^{k-1} \ (mod\ n) \\ a \equiv b \ (mod\ n) \end{array}\right\} \qquad a \cdot a^{k-1} \equiv b \cdot b^{k-1} \ (mod\ n)$$

Finalment amb propietats de les potències podem concloure que $a^k \equiv b^k \ mod\ n$, ergo $k-1 \Rightarrow k$ i per tant ens queda demostrada la propietat per qualsevol valor natural de k.

4.4. Classes d'equivalència en la congruència modular

Quan estem treballant amb congruència modular, les classes d'equivalència estan formades per tots aquells nombres que, quan els dividim entre el valor del mòdul n, obtenim el mateix residu. Per aquest motiu el nombre que es sol usar com a representant és el residu, és a dir el nombre de la classe d'equivalència que estigui entre 0 i $n-1$.

4.5. Operació mòdul

4.5.1. Definició

Quan en computació parlem de l'operació mòdul, ens referim a un concepte lleugerament diferent al de la congruència modular.

Mòdul (en computació): és l'operació que ens dona com a resultat el residu de la divisió euclidiana entre dos nombres enters. Per representar aquesta operació utilitzem l'abreviatura de mòdul mod. Per tant, siguin a, n, $r_{a/n}$ enters, podem dir que $a\ mod\ n = r_{a/n}$ on a seria el dividend, n el divisor i $r_{a/n}$ el residu obtingut.

El valor del residu està relacionat amb la congruència modular ja que un nombre sempre serà congruent amb el residu que obtenim al dividir-lo entre n. Això ho podem demostrar simplement comparant l'expressió de la divisió euclidiana amb aquella expressió deduïda a partir de la definició de congruència modular:

$$a = b + n \cdot k_1$$
$$a = n \cdot q + r$$

Com que el valor del residu, per definició, estarà entre 0 i $n-1$, podem dir que calcular-lo seria com trobar el representant de la classe d'equivalència d'a explicada a l'apartat *4.4. Classes d'equivalència en la congruència modular.*

4.5.2. Cas particular

Si volem computar el següent: $a \; mod \; n$, on $a < 0$. Sempre buscarem el valor que estigui entre 0 i $n - 1$. Això vol dir que sempre utilitzarem la convenció que ens dona el residu positiu descrita a l'apartat *4.1.1. Divisió Euclidiana.*

5.Mètodes criptogràfics actuals: criptografia asimètrica

5.1. Introducció

Des dels orígens de la criptografia, un dels grans problemes que havien d'afrontar els criptògrafs era la distribució de les claus. Com podien l'Alícia i en Bob acordar les claus de forma segura? Al llarg dels anys hi van haver moltes propostes per solucionar el problema. Per exemple, hi havia bancs que, per comunicar-se de forma segura amb els seus clients, distribuïen les claus per encriptar als seus clients mitjançant missatgers. Aquests havien de complir certs requisits com no tenir cap antecedent penal. D'aquesta manera el banc intentava assegurar-se la fiabilitat dels missatgers contractats. Tot i això, aquest procés era extremadament costós i no massa fiable ja que hi seguia havent el risc que les claus es perdessin o que algú les hi robés al missatger.

5.2. Orígens

Aquest problema va perdurar fins a finals de la dècada dels setanta, que va ser quan l'equip de criptògrafs format per Whitfield Diffie i Martin Hellman va publicar un article revolucionari. En aquest article van introduir el concepte de la criptografia asimètrica, que és aquell tipus de criptografia on la clau per encriptar i per desencriptar són diferents. Aquesta proposta va canviar completament la criptografia ja que, fins el moment, la idea que la clau per encriptar i per desencriptar havien de ser la mateixa s'havia considerat pràcticament un fet axiomàtic. Per aclarir millor la idea que ells van proposar posaré un exemple conceptual.

Suposem que l'Alícia vol enviar un missatge personal a en Bob. L'Alícia primer li envia el missatge tancat en una caixa amb un cadenat del qual només ella té la clau. Quan en Bob rep la caixa, hi afegeix el seu propi cadenat del qual només ell té la clau, i li torna a enviar a l'Alícia. Quan l'Alícia el rep, treu el seu cadenat i li torna a enviar. Quan en Bob rep la caixa, ja només queda el seu propi cadenat que treu amb la seva clau. D'aquesta forma han compartit un missatge de forma segura sense la necessitat d'intercanviar claus.

5.3. Algorisme Diffie-Hellman

Diffie i Hellman van proposar un algorisme basat en la idea explicada a l'apartat anterior, anomenat Diffie-Hellman. Tot i que era imperfecte, va ser molt innovador ja que era una solució viable al problema de la distribució de claus. Aquest algorisme es basava en una funció d'una via, és a dir, una funció fàcil de fer, però extremadament difícil de desfer. En aquest cas la funció era:

$$f(x) = p^x \ \ (mod\ q)$$

Observem que donats p, q i x és molt senzill trobar $f(x)$, en canvi donats $f(x)$, p i q és extremadament complicat trobar el valor de x.

En primer lloc s'explicarà l'algorisme amb un exemple i a continuació es justificarà matemàticament la seva validesa. Així doncs, imaginem que l'Alícia i en Bob volen acordar una clau secreta. En primer lloc comparteixen de forma pública els nombres $p = 7$ i $q = 10$ de la funció d'una via mencionada anteriorment, que en aquest cas seria:

$$f(x) = 7^x \ \ (mod\ 10)$$

Després l'Alícia avalua la funció en un nombre que ella escull i manté en secret, per exemple 3. És a dir, fa el següent càlcul:

$$f(3) = 7^3 = 343 = 3 \ \ (mod\ 10)$$

i envia el resultat a en Bob. Aquest també tria un nombre que manté en secret, per exemple 5, i realitza la mateixa operació que l'Alícia:

$$f(5) = 7^5 = \ 16807 = 7 \ \ (mod\ 10)$$

Tot seguit li envia el resultat a l'Alícia. Aquesta torna a fer la mateixa operació, però substitueix el valor de p pel nombre que en Bob li ha enviat, i en Bob fa el mateix però amb el nombre de l'Alícia. Si ara realitzem les dues operacions veurem que tenen el mateix resultat, que serà la clau secreta.

$$7^3 = 343 = 3\ (mod\ 10)$$
$$3^5 = 243 = 3\ (mod\ 10)$$

A continuació es demostrarà matemàticament que tant l'Alícia com en Bob sempre recuperen la mateixa clau. La nomenclatura que s'usarà es descriu a continuació:

p, q $\}$ $nombres\ públics\ en\ els\ quals\ es\ basen\ les\ operacions$

a: $nombre\ secret\ de\ l'Alícia$

b: $nombre\ secret\ d'en\ Bob$

α: $primer\ resultat\ de\ l'operació\ realitzada\ per\ l'Alícia$

β: $primer\ resultat\ de\ l'operació\ realitzada\ per\ en\ Bob$

k: $clau\ secreta\ compartida$

Si expressem les operacions que hem vist abans però amb aquestes lletres ens quedaria això:

$$p^a \equiv \alpha\ (mod\ q) \qquad \beta^a \equiv k\ \ (mod\ q)$$
$$p^b \equiv \beta\ \ (mod\ q) \qquad \alpha^b \equiv k\ \ (mod\ q)$$

Aplicant la propietat 5 enunciada i demostrada a l'apartat *4.3. Propietats de la congruència modular* podem deduir que si $p^a \equiv \alpha\ (mod\ q)$, $(p^a)^b \equiv \alpha^b\ (mod\ q)$. I podem dir el mateix de p^b i β. Si ara apliquem la propietat transitiva enunciada a l'apartat *4.2. Congruència modular*, podem deduir que si $\alpha^b \equiv k\ (mod\ q)$ i $(p^a)^b \equiv \alpha^b\ (mod\ q)$, aleshores $(p^a)^b \equiv k\ \ (mod\ q)$. Ara fem el mateix per p^b i β i comparem les dues expressions resultants.

$$(p^b)^a mod\ q = k$$
$$(p^a)^b mod\ q = k$$

Com podem veure, tant l'Alícia com en Bob obtenen el mateix nombre perquè han fet les mateixes operacions.

Tot i que va ser un gran pas endavant per la criptografia, aquest algorisme era imperfecte perquè només servia per compartir claus i era necessari que l'Alícia i en Bob es comuniquessin diverses vegades abans d'iniciar la comunicació. I si tenim en compte que aquest algorisme estava pensat per ser usat per persones que no es poguessin trobar físicament, no és massa pràctic que dues persones separades molts kilòmetres s'hagin d'enviar varis correus abans de començar a parlar. La seva imperfecció va provocar que comencés una carrera per a veure qui era el primer en descobrir un algorisme matemàtic bo i fiable que apliqués aquesta idea de claus asimètriques.

5.4. RSA

5.4.1. Introducció

Inspirats per la idea revolucionària de la criptografia asimètrica proposada per l'equip Diffie-Hellman, un equip format pels informàtics Ronald Rivest i Adi Shamir i el matemàtic Leonard Adleman van treballar la idea del xifratge asimètric i van acabar desenvolupant el que avui en dia coneixem com al xifratge RSA (Rivest, Shamir i Adleman). Aquest és un xifratge de clau pública, és a dir, que consta d'una clau pública i una privada. La idea d'aquest tipus de criptografia es pot exemplificar de la següent manera. Posem que l'Alícia vol enviar un missatge a en Bob, que té una empresa de finances. L'Alícia va a correus, demana un cadenat Bob i tanca el seu missatge amb aquest. Després en Bob obre el missatge amb la seva clau. D'aquesta forma, en Bob pot rebre tots els missatges que vulgui de forma segura, tenint exclusivament una clau privada.

Per veure com funciona el procés de xifrar i desxifrar també posaré un exemple. Imaginem-nos que en Bob té un banc i l'Alícia li vol enviar la contrasenya que es vol posar al seu compte bancari, per exemple 2. Així doncs, l'Alícia consulta els nombres públics que necessita per encriptar la seva contrasenya i veu la següent funció $f_e(x) = x^7 \ (mod\ 10)$. Després substitueix la x pel seu missatge i expressa el resultat que està entre 0 i 10:

$$f_e(2) = 2^7 = 128 = 8 \ \ (mod\ 10)$$

L'Alícia envia 8 a en Bob. Quan aquest rep el missatge tot el que ha de fer és, amb el mateix mòdul 10, elevar el missatge a la seva clau per desencriptar, que en aquest cas és 6.

$$8^3 = 512 = 2 \ \ \ (mod\ 10)$$

Com podem observar ha obtingut el missatge original i, en els pròxims apartats veurem perquè aquesta comunicació ha estat segura.

5.4.2. Conceptes matemàtics necessaris

5.4.2.1. Funció φ d'Euler

Definició: La funció φ d'Euler es defineix de forma que, per a qualsevol n natural, $\varphi(n)$ és la quantitat de nombres naturals menors que n que hi són coprimers.

Per aclarir la idea posaré un exemple: $\varphi(14) = 6$, ja que si escrivim tots els nombres naturals menors a 14 (a excepció del 0) i descartem tots aquells amb els que té un factor en comú ens queda: 1,~~ 2~~, 3,~~ 4~~, 5,~~ 6~~,~~ 7~~,~~ 8~~, 9,~~ 10~~, 11,~~ 12~~, 13,~~ 14~~. Si ara contem els nombres que no estan ratllats veurem que són 6.

Moltes de les propietats interessants d'aquesta funció requereixen un nivell de matemàtiques molt avançat per demostrar-les. Per tant, a continuació s'enuncia una propietat de vital importància per la seguretat del sistema RSA sense la demostració pertinent.

Multiplicitat: Donats p i q nombres naturals, $\varphi(p \cdot q) = \varphi(p) \cdot \varphi(q)$.

Un cas particular molt interessant són els nombres primers que, al només ser divisibles entre 1 i ells mateixos, són coprimers amb tots els nombres enters que estan entre 0 i ells a excepció d'ells mateixos. En conclusió:

Propietat: Sigui p un nombre primer, aleshores $\varphi(p) = p - 1$.

5.4.2.2. Algorisme d'Euclides ampliat

En aquest apartat utilitzaré la notació $mcd(a, b)$ per simbolitzar el màxim comú divisor de dos nombres enters a i b.

Aquest algorisme s'utilitza per, donats dos nombres enters a i b, trobar els nombres enters x, y i $mcd(a, b)$ perquè es compleixi que $ax + by = mcd(a, b)$. El procediment seria el següent:

1. Fem la divisió euclidiana de a entre b i ho expressem en forma d'igualtat $a = (b) \cdot q_1 + r_1$.
2. Realitzem el mateix procés, però dividint el divisor de l'expressió anterior entre el residu de l'expressió anterior, és a dir $b = (r_1) \cdot q_2 + r_2$.
3. Anem repetint el pas 2 fins que el residu ens doni 0. El residu de l'expressió anterior serà el $mcd(a, b)$.

(A partir de la tercera igualtat, totes les expressions tindran el següent format: $r_{n-2} = (r_{n-1}) \cdot q_n + r_n$ *)*

4. Suposem que ens han calgut N passos per arribar al final del pas 3. Aïllem el residu d'aquella expressió i ens quedarà quelcom així:
   $r_{N-2} - (r_{N-1}) \cdot q_N = mcd(a, b)$

5. Com que el nombre que està multiplicant al quocient és el residu de l'expressió anterior, podem aïllar-lo de l'expressió anterior i substituir-lo a la del $mcd(a,b)$.
   $r_{N-2} - (r_{N-3} - (r_{N-2}) \cdot q_{N-1}) \cdot q_N = mcd(a,b)$
6. Extraient factor comú i sumant el que està multiplicant a aquest ens queda:
   $r_{N-2} \cdot (q_{N-1} \cdot q_N + 1) - r_{N-3} \cdot q_N = mcd(a,b)$
   Si continuem substituint les expressions dels residus corresponents, en el pas k obtindrem expressions de la forma:
   $r_{N-(k+1)} \cdot t_k + r_{N-k} \cdot g_k = mcd(a,b)$ on $t_k, g_k \in \mathbb{Z}$.
   Fixem-nos que en el pas $k = N - 2$ obtenim la següent expressió:
   $r_1 \cdot t_{N-2} + r_2 \cdot g_{N-2} = mcd(a,b)$
   Substituïm r_2 i reordenem els termes:
   $r_1 \cdot t_{N-2} + (b - r_1 \cdot q_2) \cdot g_{N-2} = mcd(a,b)$
   $r_1 \cdot (t_{N-2} - q_2 \cdot g_{N-2}) + g_{N-2} \cdot b = mcd(a,b)$
7. Després de N passos, la igualtat resultant tindrà la forma $ax + by = mcd(a,b)$, amb x, y son els nombres enters que estàvem buscant. Quan fem aquest pas final obtindrem l'expressió següent:
   $$(a - b \cdot q_1) \cdot (t_{N-2} - q_2 \cdot g_{N-2}) + g_{N-2} \cdot b = mcd(a,b)$$
   $$a \cdot (t_{N-2} - q_2 \cdot g_{N-2}) + b \cdot (-q_1 \cdot (t_{N-2} - q_2 \cdot g_{N-2}) + g_{N-2}) = mcd(a,b)$$

Per aclarir aquest procés posaré un exemple senzill. Suposem que $a = 47$ i $b = 11$.

1. $47 = (11) \cdot 4 + 3$
2. $11 = (3) \cdot 3 + 2$
3. $3 = (2) \cdot 1 + 1$
   $2 = (1) \cdot 2 + 0$
4. $3 - (2) \cdot 1 = 1$
5. $11 - (3) \cdot 3 = 2$
   $3 - (11 - (3) \cdot 3) \times 1 = 1$
6. $-11 + (3) \cdot 4 = 1$
7. $47 - (11) \cdot 4 = 3$
   $-11 + (47 - (11) \cdot 4) \cdot 4 = 1$
   $-17 \cdot (11) + 4 \cdot (47) = 1$

5.4.3. Nombres de l'RSA

En aquest apartat s'introdueixen els diversos nombres amb els que treballem amb l'RSA i s'esmenten les condicions que han de complir. Aquestes seran justificades més endavant.

L'RSA treballa principalment amb tres nombres que anomenem N, e i d. N i e els triem de forma aleatòria tenint en compte certes condicions. En canvi, calculem d a partir dels altres dos.

El nombre N és el valor del mòdul que utilitzem per encriptar i desencriptar el missatge. Aquest ha de complir dues condicions: ha de ser públic i ha de ser producte de dos nombres primers, que anomenarem p i q i que s'han de mantenir en secret, d'entre 150 i 300 xifres els dos. D'altra banda, solem referir-nos al nombre e com la clau de xifrar, i és l'exponent al que elevem el missatge per encriptar-lo. Ha de complir tres condicions: ha de ser públic, coprimer amb $\varphi(N)$ i estar entre 1 i $\varphi(N)$. Finalment tenim el nombre d, al que solem referir-nos com la clau de desxifrar, que és l'exponent al que elevem un missatge per desxifrar-lo. Aquest ha de complir una condició, s'ha de mantenir en secret.

5.4.4. Relació entre e i d

Encara que la clau de xifrar i la desxifrar siguin diferents, com que una desfà l'efecte de l'altra, per força han d'estar relacionades matemàticament. Per tant, el repte que va afrontar l'equip RSA fou el de trobar una forma de fer que una pugui ser derivada de l'altre per, exclusivament, el receptor del missatge. Per aconseguir-ho van partir de dues equacions.

La primera sortia arran d'aquell concepte proposat per Diffie i Hellman, és a dir, la potenciació modular. La idea amb la que volien encriptar i desencriptar era que quan féssim mòdul N del nostre missatge a elevat a la clau d'encriptar ens donés el missatge encriptat b, i que quan elevéssim b a la clau de desencriptar ens tornés a.

$$a^e \equiv b\ (mod\ N)$$
$$b^d \equiv a\ (mod\ N)$$

Com que $a^e \equiv b\ (mod\ N)$, podem substituir[2] la b de l'expressió de sota per a^e i ens quedaria $(a^e)^d \equiv a\ (mod\ N)$. Si fem la operació d'elevar-ho a d ens quedaria $a^{e\cdot d} \equiv a\ (mod\ N)$. Aquesta és la primera expressió de la que van partir.

La segona expressió la van obtenir a partir del següent teorema:

Teorema d'Euler-Fermat: $a^{\varphi(N)} \equiv 1\ (mod\ N)$, sempre i quan a i N siguin nombres coprimers, és a dir, que no tinguin cap divisor en comú.

[2] El motiu pel qual es pot fer aquesta substitució és el mateix que s'ha utilitzat per la demostració de l'algorisme Diffie-Hellman a l'apartat *5.3. Algorisme Diffie-Hellman*

Partint d'aquest resultat, si apliquem la propietat 5 enunciada a l'apartat *4.3 Propietats de la congruència modular*, podem elevar els dos costats a un nombre k, que és qualsevol enter.

$$(a^{\varphi(N)})^k \equiv 1^k \ (mod\ N)$$
$$a^{\varphi(N)\cdot k} \equiv 1 \ (mod\ N)$$

Després, aplicant la propietat 3 demostrada al mateix apartat, multipliquem per a als dos costats:

$$a \cdot a^{\varphi(N)\cdot k} \equiv 1 \cdot a \ (mod\ N)$$

Utilitzant les propietats de les potències ho podem escriure d'aquesta forma:

$$a^{\varphi(N)\cdot k+1} \equiv a \ (mod\ N)$$

I aquesta és la segona expressió de la qual van partir.

Un cop ja tenim les dues congruències plantejades només ens falta plantejar un sistema d'equacions amb les dues expressions i utilitzar la transitivitat per arribar a:

$$\left.\begin{array}{l} a^{\varphi(N)\cdot k+1} \equiv a \ (mod\ N) \\ a^{e\cdot d} \equiv a \ \ (mod\ N) \end{array}\right\} \ a^{\varphi(N)\cdot k+1} \equiv a^{e\cdot d} \ (\ mod\ N)$$

Situem-nos ara en el cas on no només hi ha congruència, sinó també igualtat, és a dir $a^{\varphi(N)\cdot k+1} = a^{e\cdot d}$. Com que les bases de les dues expressions són iguals, els exponents també ho han de ser, ergo $\varphi(N)\cdot k + 1 = e \cdot d$. Si recordem la definició de congruència modular veurem que l'expressió anterior la podem escriure d'aquesta forma $e \cdot d \equiv 1 \ (mod\ \varphi(N))$. I com veure més endavant, aquesta expressió ens permet trobar d a partir de e.

5.4.5. Seguretat

La seguretat de l'RSA resideix en l'equació que ens permetia trobar d a partir de e: $e \cdot d \equiv 1 \ (mod\ \varphi(N))$. Tot i que es podria pensar que qualsevol persona podria trobar d a partir de e, el problema amb què es trobaria és que per computar $\varphi(N)$, sempre i quan N sigui un nombre molt gran, necessitaria anys, fins i tot amb un gran poder computacional.

Per al receptor del missatge però, és senzill computar-ho ja que, al obtenir N a partir del producte de dos nombres primers, coneix la factorització en nombres primers de N: $N = p \cdot q$, per tant ho podem calcular així $\varphi(N) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q)$ i tenint en compte que p i q són nombres primers, aleshores $\varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1)$. Per això nosaltres podem calcular $\varphi(N)$, amb facilitat perquè coneixem p i q, però qualsevol oponent que vulgui saber el valor de $\varphi(N)$, hauria de factoritzar en nombres primers N, que també requereix massa temps a computar com per comprometre el nostre sistema.

Per fer-nos una idea de com de complicat és trobar la factorització en nombres primers d'un nombre enorme, presentaré les següents dades: per fer una multiplicació de 2048 bits de llargada (unes 617 xifres decimals i una de les llargades usada en sistemes RSA) es necessiten aproximadament 53,75s amb un ordinador normal. D'altra banda, si volguéssim factoritzar un nombre de la mateixa llargada, tardaríem aproximadament 6.871.455.104.760.852.000 anys! Per fer-nos una idea del que suposa aquesta xifra, si aquest ordinador hagués començat la factorització a l'inici de l'univers, tot just passaria pel 0,0000002%!

5.4.6. Càlcul de d

La clau per desencriptar d és l'únic nombre necessari per a l'RSA que requereix un procediment més complex que una simple multiplicació, per aquest motiu explicaré com calcular-lo. Coneguts e i $\varphi(N)$, hem de trobar una solució de la següent equació $e \times d \equiv 1\ (mod\ \varphi(N))$. Per trobar d ràpidament sense haver de recórrer al tempteig s'utilitza l'algorisme d'Euclides ampliat. Però per poder-lo usar es necessitava una igualtat del tipus $ax + by = mcd(a, b)$, per tant que s'ha de justificar que és vàlid aplicar aquest mètode.

Per fer-ho, utilitzem la propietat simètrica per escriure la congruència modular així $1 \equiv e \times d\ (mod\ \varphi(N))$, que per definició vol dir que existeix un nombre enter k tal que $\varphi(N) \cdot k + e \cdot d = 1$. Com podem observar, el costat esquerre de la igualtat ja és idèntic al de l'algorisme d'Euclides ampliat. Per igualar el costat esquerre posem la condició que e i $\varphi(N)$ han de ser nombres coprimers, ja que si ho són voldrà dir que $mcd(\varphi(N), e) = 1$ i ja podrem aplicar l'algorisme d'Euclides que he explicat anteriorment i la solució serà el nombre que estigui multiplicant e en la igualtat resultant. En cas que la solució obtinguda sigui un nombre negatiu, per evitar nombres amb decimals quan desencriptem el missatge hem de trobar un representant de la seva classe d'equivalència que sigui positiu.

5.4.7. Característiques d'e

Com he dit a l'apartat nombres de l'RSA, e ha de complir que: $1 < e < \varphi(N)$. La primera part d'aquesta condició, ha de ser major a 1, és obligatòria, en canvi la segona és altament recomanable però no necessària teòricament.

En primer lloc e no pot ser menor a 0 perquè si treballéssim amb exponents negatius ens donaria nombres decimals. Això no ho podem fer ja que quan treballem amb mòdul ho fem sempre amb enters.

Tampoc pot ser 0 perquè si ho fos, seria impossible trobar el valor de d degut a què aquesta congruència no tindria solució $0 \cdot d \not\equiv 1 \ (mod\ N)$. El fet que no tingui solució es deu a què qualsevol nombre elevat a 0 dona 1, i com que 1 elevat a qualsevol enter dona 1, és impossible que elevant-ho a d ens torni el missatge original.

D'altra banda, tot i que és matemàticament possible que e sigui 1, com que elevar a 1 no afecta al valor d'un nombre, el missatge original i el xifrat serien el mateix. Per tant estaries enviant el missatge sense cap mena de xifratge que eviti que algú n'extregui informació.

Finalment, tot i que és matemàticament possible i no afecta a la seguretat de l'RSA, es demana que $e < \varphi(N)$. Aquesta condició es posa perquè, si per casualitat el missatge que es vol enviar és coprimer amb N, aleshores segons el Teorema d'Euler-Fermat que el valor de e, ja sigui gran o petit, no afecta a la seguretat de l'RSA. I si tenim en compte que e és l'exponent al que elevem el missatge, que el seu valor sigui molt elevat només complica el procés de computació del missatge xifrat sense fer-lo més segur.

5.4.8. Enviament de missatges llargs

Com que, en computació, el representant de les classes d'equivalència de la congruència modular és el valor que està entre 0 i $N - 1$, si intentem enviar un missatge amb un valor major a N, quan elevem aquest a d, no obtindrem el missatge original ja que no estarà entre 0 i $N - 1$. Encara que pot semblar un gran inconvenient per aquest mètode, en realitat la solució és molt senzilla: enviar el missatge en blocs que tinguin, com a molt, una xifra menys que N.

5.4.9. Mètode per agilitzar els càlculs

Com que la xifra RSA utilitza nombres molt grans, sovint ens trobem càlculs que costen de computar, per exemple: $987462^{74323}\ (mod\ 25173904653) = ?$. Com que per definició, un nombre és congruent en mòdul n amb el residu que obtenim quan el dividim entre n, i si tenim en compte que en computació calculem el residu de la següent forma $a\ (mod\ n)$, podem aplicar la propietat 5 enunciada i demostrada a l'apartat *4.3. Propietats de la congruència modular* perquè ens quedi el següent:

$$a^k \equiv (a\ mod\ n)^k \quad (mod\ n)$$

Per veure com s'aprofita aquesta propietat posaré un exemple: suposem que volem calcular $3^{250}\ mod\ 67$. Com que 3^{250} és un nombre enorme amb el que hem d'operar, podem simplement fer el següent: $(3^{50})^5 \equiv (3^{50} mod\ n)^5\ \ (mod\ n)$. D'aquesta forma no hem de fer operacions amb nombres tant grans, ergo el temps requerit per computar la operació serà reduït dràsticament.

5.4.10. Exemple

Per veure tot això en acció posaré un exemple amb nombres naturals petits perquè es vegi de forma clara tots els passos de l'RSA. Imaginem-nos que l'Alícia i en Bob són dos adolescents que volen quedar en una hora concreta sense que els seus pares ho sàpiguen. En primer lloc, en Bob agafa dos nombres primers p i q, per exemple 7 i 13. I els multiplica per obtenir N: $N = 13 \cdot 7 = 91$. A continuació calcula $\varphi(N) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1) = (13-1) \cdot (7-1) = 72$ i escull una clau per xifrar e que sigui major a 1 i coprimer amb 72, per exemple 11. A continuació envia la següent informació públicament, a l'Alícia: $N = 91\ i\ e = 11$. Mentre espera a la resposta de l'Alícia, en Bob calcula la seva clau secreta d aplicant l'algorisme ampliat d'Euclides:

$$72 = (11) \cdot 6 + 6$$
$$11 = (6) \cdot 1 + 5$$
$$6 = (5) \cdot 1 + 1$$
$$5 = (1) \cdot 5 + 0$$
$$1 = 6 - (5) \cdot 1$$
$$1 = 6 - (11 - (6) \cdot 1) \cdot 1$$
$$1 = (6) \cdot 2 - 11 \cdot 1$$
$$1 = (72 - 11 \cdot 6) \cdot 2 - 11 \cdot 1$$
$$1 = (2) \cdot 72 + (-13) \cdot 11$$

Com que li ha donat un nombre negatiu $d = -13$, computa $-13 \equiv 59 \pmod{72}$, és a dir $d = 59$. D'altra banda l'Alícia ha decidit que quedaran a les 16:00, per tant, fa la operació següent utilitzant el mètode descrit a l'apartat *5.4.9. Mètode per agilitzar els càlculs*:

$$16^{11} \equiv (16^5)^2 \cdot 16 \equiv (16^5 \pmod{91})^2 \cdot 16 \ ((\mod 91)) \equiv 74 \cdot 74 \cdot 16$$
$$\equiv 74 \pmod{91}$$

I envia el resultat a en Bob. Aquest eleva el resultat a 59 utilitzant el mateix mètode per desxifrar-lo:

$$74^{59} \equiv (74^{29})^2 \cdot 74 \equiv (74^{29} \pmod{91})^2 \cdot (74 \pmod{91}) \equiv 16 \cdot 16 \cdot 74$$
$$\equiv 16 \pmod{91}$$

I com hem pogut veure, d'aquesta forma l'Alícia i en Bob han intercanviat un missatge de forma segura sense la necessitat de trobar-se físicament per fer-ho.

5.5 ElGamal

5.5.1. Introducció

ElGamal és un xifrat de clau pública que fou proposat pel criptògraf egipci Taher ElGamal el 1984. A l'igual que l'RSA, va sorgir a partir de l'Algorisme Diffie-Hellman, amb el que comparteix moltes característiques. En aquest cas l'explicació a nivell conceptual seria, aproximadament, la mateixa que s'utilitza per explicar l'RSA a l'apartat *5.4.1. Introducció*.

5.5.2. Nombres del ElGamal

En aquest apartat es defineixen els nombres que s'utilitzaran en els següents apartats, les condicions que han de complir.

$\boldsymbol{p}$: És el valor del mòdul. L'escollim de forma aleatòria, amb la condició que ha de ser primer i tenir unes 150 xifres, i és públic.

$\boldsymbol{g}$: Nombre públic que ha de ser coprimer amb p.

$\boldsymbol{x}$: Nombre privat del receptor del missatge que ha de complir que $1 < x < p - 1$.

$\boldsymbol{y}$: Nombre privat de l'emissor del missatge que ha de complir que $1 < y < p - 1$.

$\boldsymbol{m}$: Missatge que volem enviar. Ha de ser menor a p pel mateix motiu esmentat a *5.4.8. Enviament de missatges llargs*.

$\boldsymbol{h}$: Nombre públic calculat a partir de la següent operació: $g^x \equiv h \pmod{p}$.

$\boldsymbol{a}$**:** Nombre que forma part del missatge xifrat que enviem. El calculem a partir de la següent operació: $g^y \equiv a \pmod{p}$.

$\boldsymbol{b}$: Nombre que forma part del missatge xifrat que enviem. El calculem a partir de la següent operació: $m \cdot h^y \equiv b \pmod{p}$.

$\boldsymbol{c}$: missatge xifrat. En aquest cas són dos nombres, l'a i el b.

5.5.3. Conceptes necessaris

5.5.3.1. Petit teorema de Fermat

Aquest teorema és un cas particular del teorema Euler-Fermat enunciat a l'apartat *5.4.4. Relació entre e i d*. Recordem que diu el següent:

Teorema Euler-Fermat: $a^{\varphi(N)} \equiv 1 \pmod{N}$, sempre i quan a i N siguin nombres coprimers, és a dir, que no tinguin cap divisor en comú.

El cas particular d'aquest teorema es dona quan N es tracta d'un nombre primer. En aquest cas, si apliquem el cas particular de la funció φ, enunciat a l'apartat *5.4.2.1. Funció φ d'Euler*, següent: si p és un nombre primer, aleshores $\varphi(p) = p - 1$. Si substituïm al teorema d'Euler-Fermat ens quedaria el cas particular d'aquest:

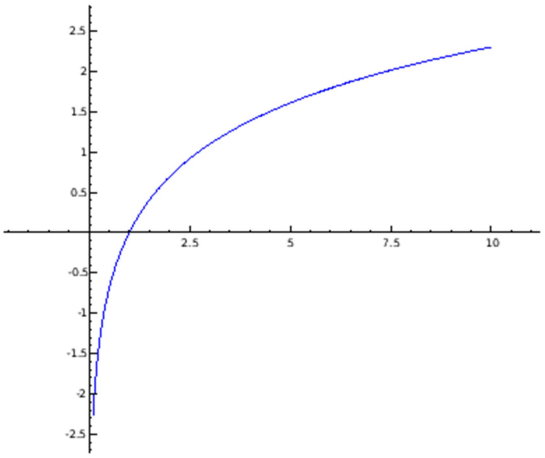
Petit teorema de Fermat: $a^{p-1} \equiv 1 \pmod{p}$, sempre i quan és a sigui coprimer amb p i p sigui un nombre primer.

El petit teorema de Fermat s'utilitzarà per a l'explicació d'algunes propietats dels nombres de ElGamal i per la demostració de l'algorisme

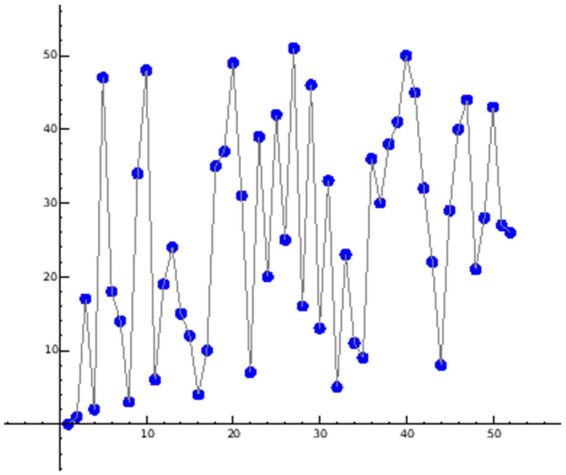
5.5.3.2 Problema del logaritme discret

Logaritme: és un exponent. El logaritme d'un nombre, diguem-li x, en base N és el nombre al que hem d'elevar N per obtenir x. És a dir, si $x = N^y$, aleshores $\log_N x = y$. Per aclarir el seu ús posaré un exemple: suposem que volem calcular un valor y sabent que $3^y = 243$, per fer-ho simplement hem d'aplicar el logaritme: $y = \log_3 243 = 5$.

El logaritme discret amb el que treballarem es basa en aplicar aquest mateix concepte, però en la congruència modular. És a dir, si $x \equiv N^y \pmod{p}$, aleshores $\log_N x \equiv y \mod (p)$. I la qüestió és que, aquesta funció, és realment difícil de computar per a nombres relativament grans. La millor forma de comprendre la dificultat que comporta calcular un logaritme discret és comparar la gràfica que obtenim amb una funció de logaritme discret amb una logarítmica normal.



9. Gràfica d'una funció logarítmica normal



10. Gràfica d'una funció logarítmica mòdul 53

Com podem observar en les dues gràfiques, mentre que la logarítmica normal segueix una tendència de forma clara, quan hi afegim "mòdul 53" passa a ser una sèrie de punts que no segueixen cap mena de patró. De fet, avui en dia, el millor mètode que es coneix per calcular y sabent que $x \equiv N^y \pmod{p}$, és el d'anar provant nombres fins a trobar una solució aproximada. Òbviament, si el valors amb els que es treballen són relativament petits, no suposa un gran problema. Per tant, perquè realment presenti un problema, el nombre del mòdul ha de ser d'unes 150 xifres.

5.5.4. Encriptar i desencriptar

Tot i que ElGamal parteix, a l'igual que l'RSA, de l'algorisme Diffie-Hellman, el procés d'encriptar i desencriptar són completament diferents. El procés és el següent:

1. En Bob escull p, g i x. I calcula h a partir de $g^x \equiv h \pmod{p}$.
2. A continuació fa públics els nombres p, g i h. El nombre x el manté en secret.
3. Després l'Alícia, per enviar-li el missatge m a en Bob, calcula a i b a partir de $g^y \equiv a \pmod{p}$ i $m \cdot h^y \equiv b \pmod{p}$.
4. L'Alícia envia a i b a en Bob.
5. Finalment, en Bob calcula $a^{p-1-x} \cdot b \equiv m \pmod{p}$ per obtenir el missatge original de l'Alícia.

5.5.5. Demostració de l'algorisme

En aquest apartat, a no ser que es digui el contrari, totes les propietats que s'utilitzaran són enunciades i demostrades a l'apartat *4.3. Propietats de la congruència modular*. Amb l'excepció de la propietat transitiva, que és enunciada i demostrada a l'apartat *4.2. Congruència modular*.

Per demostrar aquest algorisme hem de partir de la següent expressió:

$$a^{p-1-x} \cdot b \equiv m \pmod{p}$$

I arribar a:

$$m \equiv m \pmod{p}$$

És a dir, hem de demostrar que l'operació per desxifrar el missatge és congruent amb el missatge original. Per fer-ho aplicarem diverses propietats per substituir els valors fins arribar a m. En primer lloc substituïm b utilitzant la propietat transitiva i la 3:

$$\left.\begin{array}{l} b \equiv m \cdot h^y \pmod{p} \\ a^{p-1-x} \cdot b \equiv m \pmod{p} \end{array}\right\} \; a^{p-1-x} \cdot m \cdot h^y \equiv m \pmod{p}$$

Després utilitzem la propietat 5, la 3 i la transitiva per substituir h:

$$\left.\begin{array}{l} h \equiv g^x \pmod{p} \\ a^{p-1-x} \cdot m \cdot h^y \equiv m \pmod{p} \end{array}\right\} \; a^{p-1-x} \cdot m \cdot g^{xy} \equiv m \pmod{p}$$

Ara tornem a aplicar la propietat 5, la 3 i la transitiva però per substituir a:

$$\left.\begin{array}{l} a \equiv g^y \pmod{p} \\ a^{p-1-x} \cdot m \cdot g^{xy} \equiv m \pmod{p} \end{array}\right\} \; g^{y\cdot(p-1-x)} \cdot m \cdot g^{xy} \equiv m \pmod{p}$$

Utilitzant les propietats de les potències ho reordenem i descartem els termes que se'ns anul·len:

$$g^{y\cdot(p-1)} \cdot m \cdot g^{xy} \cdot g^{-xy} \equiv m \pmod{p}$$

$$(g^{(p-1)})^y \cdot m \equiv m \pmod{p}$$

Finalment utilitzem el cas particular del Teorema Euler-Fermat, el Petit teorema de Fermat, la propietat 5, la 3 i la transitiva per substituir g^{p-1}.

$$\left.\begin{array}{l}(g^{(p-1)})^y \cdot m \equiv m \pmod p \\ g^{p-1} \equiv 1 \pmod p\end{array}\right\} \; 1^y \cdot m \equiv m \pmod p$$

Com que 1 elevat a qualsevol nombre és 1, arribem a la conclusió que desitjàvem:

$$m \equiv m \pmod p$$

5.5.6. Condicions de x, y, g i p

En primer lloc tenim la condició de x, que també ha de complir y i pels mateixos motius, següent: $1 < x < p - 1$. Aquesta la dividirem en dues condicions: $1 < x$ i $x < p - 1$. La justificació de la primera és exactament la mateixa que l'explicada en el segon i tercer paràgraf de l'apartat *5.4.7. Característiques d'e*.

La segona condició ($x < p - 1$) es demana perquè la grandària del valor del nombre x no afecta gaire a la seguretat del xifrat però sí que dificulta la computació dels càlculs ja que provoca que el valor de la resta de nombres creixi exponencialment. A més a més, si es dona el cas que $x = p - 1$, pel petit teorema de Fermat enunciat a l'apartat *5.5.4.1. Petit teorema de Fermat*, $h \equiv g^x \equiv g^{p-1} \equiv 1 \pmod p$. Això provocaria que enviéssim el missatge sense xifrar perquè: $b \equiv m \cdot h^y \equiv m \cdot 1 \equiv m \mod p$.

En el cas que $y = p - 1$, ens trobaríem amb un problema similar que també el provoca el Petit teorema de Fermat enunciat al paràgraf anterior. Ja que, si apliquem la propietat 3 i la transitiva enunciades i demostrades a l'apartat *4.3. Propietats de la congruència modular*, podem arribar a:

$$\left.\begin{array}{l}b \equiv m \cdot h^{p-1} \pmod p \\ h^{p-1} \equiv 1 \pmod p\end{array}\right\} \; b \equiv m \cdot 1 \equiv m \pmod p$$

Això vol dir que estaríem enviant el missatge sense xifrar.

D'altra banda tenim els nombres: p; que ha de ser primer i g; que ha de ser coprimer amb p. Aquestes dues condicions es demanen pel mateix motiu, que seria el de que compleixin el petit teorema de Fermat mencionat a l'inici d'aquest apartat. Ja que, si no el complissin, no podríem realitzar l'última substitució que hem realitzat a l'apartat *5.5.5. Demostració de l'algorisme*. Ergo, no obtindríem m al realitzar la operació:

$$a^{p-1-x} \cdot b \equiv m \pmod p$$

5.5.7. Seguretat

La seguretat del xifrat ElGamal resideix en el problema del logaritme discret explicat a l'apartat *5.5.3.2. El problema del logaritme discret*. Per veure que realment és el que fa que el xifrat sigui segur ens hem de fixar en les claus x i y, ja que són els nombres privats que necessitem saber per a poder obtenir la informació del missatge encriptat. Per una banda, si tenim x podem realitzar la operació següent per obtenir el missatge original:

$$a^{p-1-x} \cdot b \equiv m \pmod{p}$$

D'altra banda, si coneixem y també podem arribar a conèixer el missatge original (encara que no d'una forma tant directa com quan coneixes x) a través de l'expressió següent:

$$b \equiv m \cdot h^{y} \pmod{p}$$

En aquest cas però, per trobar m hauríem d'anar provant valors de m fins a descobrir-lo. Per tant que, tot i que també involucraria certa dificultat, aquesta es veuria reduïda. Un cop comprovat que, descobrint el valor de x o de y, es pot obtenir el missatge. Hem d'assegurar-nos que només podem descobrir aquests dos nombres a través d'un logaritme discret. Per tant que ens hem de mirar les relacions a partir de les quals podries obtenir algun d'aquest dos valors, que són:

$$h \equiv g^{x} \pmod{p}$$
$$a \equiv g^{y} \pmod{p}$$

Si ara comparem aquestes dues expressions amb la de la definició de logaritme discret esmentada a l'apartat *5.3.4.2. El problema del logaritme discret*, comprovarem que, efectivament, la forma de trobar x i y a partir dels nombres públics és a través del logaritme discret que, com ja hem explicat, és massa difícil de computar.

5.5.8. Exemple

Per veure tot això en acció posaré un exemple amb nombres naturals petits.
Suposem que en Bob i l'Alícia són una parella de famosos que volen quedar a certa hora de forma secreta. En primer lloc en Bob escull els valors dels nombres g, p i x, que seran 5, 13 i 2 respectivament. Aleshores calcula:

$$h \equiv 5^{2} \equiv 25 \equiv 12 \pmod{13}$$

Per tant, $h = 12$. Després li envia h, g i p a l'Alícia de forma pública i manté el nombre x en secret. Mentrestant, l'Alícia ha decidit que quedaran a les 6 i ha escollit un nombre y, posem 3, que mantindrà en secret. Per tant, quan rep els nombres h, g i p d'en Bob, calcula a i b fent les operacions següents:

$$a \equiv 5^3 \equiv 125 \equiv 8 \pmod{13}$$
$$b \equiv 6 \cdot 12^3 \equiv 10368 \equiv 7 \pmod{13}$$

Per tant, $a = 8$ i $b = 7$. A continuació l'Alícia envia aquests dos nombres a en Bob. Finalment en Bob , per saber quina és l'hora acordada, calcula:

$$8^{13-1-} \cdot 7 \equiv 6 \pmod{13}$$

D'aquesta forma en Bob ha pogut rebre un missatge de forma secreta sense la necessitat de trobar-se físicament amb l'Alícia.

5.6. Comparació entre el ElGamal i l'RSA

5.6.1. Similituds

Ambdós mètodes pertanyen al camp de la criptografia asimètrica de clau pública. És a dir, ambdós parteixen d'una clau privada i una clau pública. Això permet que els dos mètodes siguin utilitzats per la firma digital. Aquesta ens permetria signar un contracte a la xarxa. Per explicar el funcionament d'aquesta aplicació de la criptografia de clau pública posaré un exemple: imaginem-nos que en Bob ha de signar un contracte online. Per fer-ho, en primer lloc es llegiria el contracte i afegiria a la part inferior: «Jo, Bob Smith, estic d'acord i accepto els termes d'aquest contracte». Després encripta el text amb una clau que mantindrà privada i fa públics el text encriptat i la clau de desencriptar. D'aquesta forma es pot comprovar que ha estat en Bob el que ha signat, perquè només ell té la clau que pot transformar el text desencriptat en el text encriptat públic.

També comparteixen dos grans inconvenients. El primer és que, al treballar els dos mètodes amb la congruència modular, no podem enviar missatges que tinguin més dígits que el valor del mòdul. L'altre és que no permeten a l'Alícia i en Bob mantenir-se en l'anonimat. És a dir, tot i que no es pot extreure informació dels missatges que s'envien, es pot saber la persona que ha enviat un missatge i la que l'ha rebut i això, en nombroses ocasions, ja proporciona molta informació. Per exemple: imaginem-nos que un càrrec polític envia un missatge codificat amb una d'aquestes dos xifrats a un líder del crim organitzat. Encara que sigui extremadament complicat esbrinar el

contingut del missatge, el simple fet aquestes dues persones es comuniquin dona informació molt reveladora.

Un altre aspecte en comú és que ambdós mètodes, tot i ser molt potents, poden ser "esquivats". És a dir, com que s'han de mantenir nombres en secret en un ordinador, és possible que un hacker obtingui aquests nombres mitjançant programes maliciosos o *malware*. Per exemple es pot donar el cas que l'usuari de l'ordinador que té la clau es descarregui accidentalment un programa informàtic que extregui la informació que necessita per poder desxifrar els missatges enviats i rebuts.

També hi ha una altra forma, encara que molt poc pràctica, d'obtenir els missatges enviats sense conèixer cap nombre privat. Aquesta consisteix en tenir un vehicle que estigui equipat amb una tecnologia que permet captar les pulsacions electromagnètiques que es produeixen quan es prem una tecla en el teclat d'un ordinador. És a dir, aquesta tecnologia et permet saber en tot moment quines lletres estan essent premudes en un ordinador que sigui pròxim.

5.6.2. Diferències

La primera diferència significativa que podem observar és la complexitat dels seus processos de xifrat i desxifrat. A simple vista ja és obvi que l'RSA és més simple en aquest aspecte ja que només s'ha de fer una operació per xifrar i una per desxifrar qualsevol missatge. D'altra banda, per xifrar un missatge amb ElGamal has de realitzar dues operacions. A més a més, com que has d'enviar dos nombres, el missatge xifrat sol ser el doble de llarg que el missatge original.

Una altra diferència significativa és el poder computacional que necessiten. Aquest és major en l'RSA ja que es necessita operar amb nombres de fins a 617 xifres decimals. D'altra banda, les xifres amb les que opera ElGamal són la meitat de grans que les de l'RSA. Per tant, el poder computacional necessitat per l'RSA és considerable en comparació amb el que necessita ElGamal.
Un altre aspecte que els diferencia és que, per poder-te comunicar amb ElGamal, tant l'emissor com el receptor han de generar i mantenir una clau en secret. En canvi, l'RSA, només el receptor necessita generar i mantenir certs nombres en secret. Això comporta una avantatge sobre ElGamal, ja que, si per exemple, vols enviar un missatge a un banc que utilitza RSA, el teu ordinador només necessita fer una operació. En canvi, si usa ElGamal, a més de les operacions pertinents, has de generar un nombre i assegurar-te de que es mantingui en secret.

6. El llenguatge dels ordinadors

6.1. Introducció

En aquest apartat parlarem de la codificació que és molt similar a la criptografia. En el que es diferencien és que la criptografia sol usar algorismes per fer que un missatge sigui enigmàtic, d'altra banda, la codificació sol usar mètodes de substitució per transcriure el llenguatge natural a un amb signes, nombres... I en aquest apartat ens mirarem alguns codis que no serveixen per afegir seguretat a la nostra comunicació, sinó que serveixen per comunicar-nos amb els ordinadors. És a dir, codis que s'utilitzen per transcriure un text qualsevol, en una sèrie de nombres que un ordinador pugui llegir i interpretar. Aquests codis estan tots centrats en el sistema binari, és a dir, el llenguatge basat ens uns i zeros. Cada un d'aquests uns o zeros es considera un bit d'informació i vuit bits fan un byte.

6.2. ASCII

L'ASCII (American Standard Code for Information Interchange) és un codi que s'utilitza per convertir accions o text que vulguis enviar a un ordinador a codi binari. Originalment, al 1967, els caràcters de l'ASCII estaven formats per cadenes de 7 bits, és a dir que hi havia $2^7 = 128$ caràcters possibles (ja que les formes possibles d'ordenar 7 uns i zeros és 2^7). 14 anys després, l'empresa IBM va afegir un vuitè bit per augmentar la llista de caràcters a $2^8 = 256$. Aquests 256 caràcters es poden classificar en tres blocs: els de control (accions que ha de fer l'ordinador), els imprimibles (els necessaris per escriure en anglès) i els de l'ASCII ampliat (lletres amb accents, la ñ i altres símbols). Tots els caràcters es mostren a la taula següent.

Caracteres ASCII de control

00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc.vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sínc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

Caracteres ASCII imprimibles

32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

ASCII extendido (Página de código 437)

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

11. Taula de caràcters de l'ASCII

6.3. Canvis de base

Si observem la taula present a l'apartat anterior, veurem que cada caràcter va acompanyat d'un nombre. Aquest identifica cada caràcter en base decimal. Ara bé, per identificar els mateixos caràcters, podem utilitzar altres bases com l'hexadecimal o la binària. Aquesta última és la fonamental que utilitzen tots els ordinadors, per aquest motiu els canvis de base que s'expliquen en aquest apartat són d'hexadecimal a binari i viceversa i de decimal a binari i viceversa.

De binari a hexadecimal i viceversa:

El codi hexadecimal és el segon més utilitzat després del binari. Això es deu a què, com hem vist a l'apartat anterior, el codi binari s'utilitza en tires de 8 dígits que fan 2^8 caràcters possibles. Ara bé, aquest 2^8 també el podem escriure com $2^4 \cdot 2^4 = 16 \cdot 16$. Per tant que dos caràcters hexadecimals fan 1 byte. Per convertir un byte en binari a hexadecimal tot el que hem de fer és dividir els vuit bits en dues tires de 4 bits cadascuna i substituir-ho pels valors que hi ha a la taula següent:

Binari	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	a
1011	b
1100	c
1101	d
1110	e
1111	f

12. Taula per la conversió de binari a hexadecimal i viceversa

I en cas que el nombre en binari no sigui múltiple de 4 hem d'afegir el nombre mínim de zeros a l'esquerre perquè es pugui dividir en blocs de 4. Per tant que, per exemple, el nombre en binari de 00101011 en hexadecimal seria 2b.

De binari a decimal:

Per passar de binari a decimal simplement hem d'observar com s'escriuen els nombres amb base decimal. Per exemple, el nombre 251 en base decimal, en realitat és $2 \cdot 10^2 + 5 \cdot 10^1 + 1 \cdot 10^0$. És a dir, un nombre en base decimal s'expressa multiplicant les xifres d'aquest, començant per la dreta, per $10^0, 10^1, 10^2, \ldots\ldots 10^{n-1}$ on n és el nombre de xifres que té el número. Per tant, per passar d'un nombre en binari a un nombre en decimal hem d'aplicar el mateix concepte, però a més de multiplicar per 10, multiplicar per 2. Per exemple: el nombre 01100101 en base decimal seria $0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 0 \cdot 2^1 + 1 \cdot 2^0 = 101$.

De decimal a binari:

Per passar de decimal a binari hem de fer els següents passos tenint en compte que la y serà el nombre en base decimal, q el quocient, n el nombre de divisions realitzades i r el residu:

1. Fem la divisió euclidiana del nombre en base decimal entre 2 i anotem el residu que dona. És a dir fem:
   $y = q_1 \cdot 2 + r_1$
2. Repetim el pas dos però substituint y per q_1 fins que $q_n = 0$. Les expressions d'aquest pas tindran aquest format:
   $q_{n-1} = q_n \cdot 2 + r_n$
3. Aleshores anotem, d'esquerre a dreta, l'últim quocient que no sigui igual a 0 i a continuació tots els residus excloent el del $q_n = 0$.
4. Com que l'ASCII treballa amb 8 dígits binaris, en cas que el nombre obtingut tingui menys dígits, el que s'ha de fer és afegir zeros a l'esquerra fins que tingui 8 dígits.

Per veure-ho de forma clara posaré un exemple en el que es canvia de base el nombre 83 on les expressions tindran el format descrit anteriorment en aquest apartat (els nombres en verd són els que formen el nombre en binari):

1. $83 = 41 \cdot 2 + 1$
2. $41 = 20 \cdot 2 + 1$
   $20 = 10 \cdot 2 + 0$
   $10 = 5 \cdot 2 + 0$
   $5 = 2 \cdot 2 + 1$
   $2 = 1 \cdot 2 + 0$
   $1 = 0 \cdot 2 + 1$

3. 1010011

4. 01010011

6.4. Usos de l'ASCII

L'ASCII és un codi que tots els ordinadors són capaços d'interpretar. Això ens permet escriure qualsevol dels caràcters presents a la taula de l'apartat *6.2. ASCII* amb qualsevol ordinador mantenim premuda la tecla alt i teclegem, al teclat numèric de la dreta del teclat el valor en base decimal del caràcter desitjat i deixem de prémer alt. Això ens pot ser realment útil quan fem servir ordinadors d'altres països. Per exemple: si compressis un ordinador als Estats Units, el teclat d'aquest no tindria la lletra ñ, però podries teclejar-la igualment fent: alt+164.

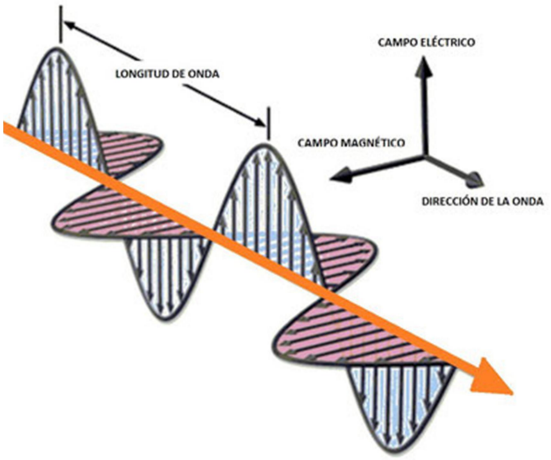
7. Criptografia quàntica

7.1. Introducció

La seguretat dels xifrats actuals, tal i com hem vist a l'apartat *5. Mètodes criptogràfics actuals: criptografia asimètrica*, resideix en que ningú disposa de prou poder computacional com per comprometre el nostre sistema. Per tant, si en un futur no massa llunyà disposéssim d'ordinadors quàntics (basats en les propietats de la física quàntica i infinitament més potents que els actuals) tots aquests xifrats quedarien obsolets. Això faria necessària la seva substitució. L'eina que ens permetria crear nous xifrats segurs que substituïssin els actuals és la física quàntica. Aquesta és la branca de la física que estudia el comportament de les partícules subatòmiques. Com que aquest camp de recerca és relativament recent, els mètodes criptogràfics que veurem són teòrics, és a dir encara no són utilitzats.

7.2. Conceptes necessaris

La llum: és una radiació electromagnètica. És a dir, està formada per un camp elèctric i un camp magnètic tal i com es mostra en la següent imatge.



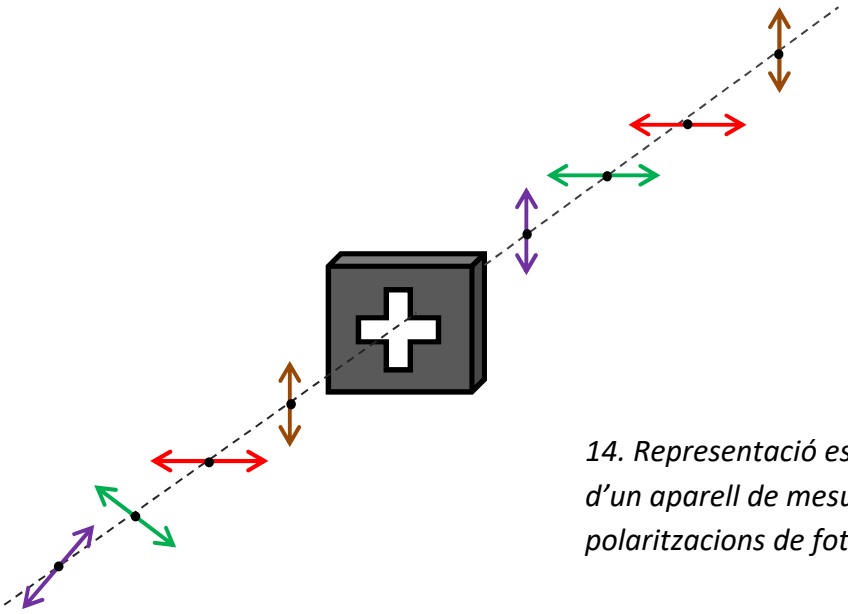
13. Representació esquemàtica de la llum

Fotó: partícula que no té massa i que conforma la llum.

Dualitat ona-corpuscle de la llum: els fotons que conformen la llum tenen un caràcter dual. És a dir, en certes circumstàncies mostren aspectes ondulatoris (es comporten com una ona) i mostren aspectes corpusculars (es comporten com una partícula) en d'altres.

Polarització d'un fotó: es defineix com la direcció en la que vibra el camp elèctric d'un fotó. N'hi ha de tres tipus: lineal, circular i el·líptica. En els mètodes criptogràfics que veurem en els pròxims apartats sempre treballarem amb polaritzacions lineals i amb les bases rectilineal (+) i diagonal (×). És a dir, considerarem que un fotó només pot estar polaritzats en les següents direccions: ↔, ↕, ↗ i ↘.

Mesura de la polarització d'un fotó: determinar amb precisió la polarització d'un fotó és molt complicat, fins i tot quan només treballem amb les dues bases mencionades abans. Això es deu a què els aparells de mesura funcionen com filtres que només deixen passar la llum que estigui polaritzada en una base determinada. Per explicar què passa quan mesurem la polarització d'un fotó hem de mirar-nos l'esquema que apareix a continuació on: els punts negres representen els fotons, les fletxes representen les seves polaritzacions que, inicialment, són; ↗ (lila), ↘ (verd), ↔ (vermell) i ↕ (marró), l'objecte quadrat que està al mig és un aparell de mesura amb base +, la línia discontínua representa la trajectòria dels fotons i aquests van des de baix a l'esquerra fins a dalt a la dreta i els colors de les fletxes identifiquen a cada fotó.

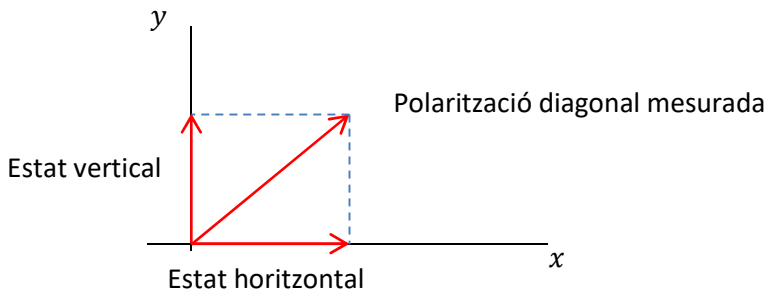


14. Representació esquemàtica d'un aparell de mesura de polaritzacions de fotons

Com podem veure a l'esquema, la polarització dels fotons que estaven polaritzats amb la base rectilineal no varia. D'altra banda, la polarització dels fotons que estaven polaritzats amb la base diagonal surten de l'aparell de mesura polaritzats rectilinealment. El mateix fenomen passa quan intentem mesurar un fotó polaritzat rectilinealment amb un detector de base diagonal. De fet, quan ens equivoquem de base amb la que mesurar, tenim un 50% de probabilitats d'obtenir cadascuna de les dues polaritzacions mesurables amb el nostre aparell. És a dir, sempre obtindrem un

dels dos resultats que possibles amb la base que estiguem usant per mesurar. La polarització mesurada serà sempre la final del fotó. Això vol dir que si ens equivoquem de base amb la que mesurar, no només obtindrem una mesura incorrecta, sinó que també canviarem la polarització del fotó. Aquest fenomen es considera vertaderament aleatori ja que, encara que reproduïm les condicions exactes en les que es trobava el fotó, pots no obtenir el mateix resultat.

Actualment l'explicació d'aquest fenomen és que el fotó està en una superposició d'estats. És a dir, si nosaltres mesurem un fotó i obtenim que està polaritzat diagonalment, en realitat, aquest fotó està en una superposició d'estar polaritzat horitzontalment i verticalment. I cada un d'aquests estats hi és en certa proporció. I és aquesta proporció la que ens dona les probabilitats. L'esquema que apareix a continuació representa amb vectors el que s'acaba d'explicar:



15. Representació vectorial de les polaritzacions

A partir d'aquesta representació vectorial de la situació podem calcular les probabilitats que tenim d'obtenir la polarització ↔ quan mesurem amb base + un fotó amb polarització ↗. La notació que s'utilitzarà serà la següent:

$|H>: vector\ unitari\ horitzontal\ (representa\ la\ polarització\ \leftrightarrow) = (1{,}0)$
$|V>: vector\ unitari\ vertical\ (representa\ l'estat\ \updownarrow) = (0{,}1)$
$P_{\varphi} = probabilitats\ de\ mesurar\ l'estat\ que\ volem\ observar$
$< a|b > = \vec{a} \cdot \vec{b} = (a_x\ ,\ a_y) \cdot (b_x\ ,\ b_y) = a_x \cdot b_x + a_y \cdot b_y$
$\varphi = estat\ que\ volem\ observar$
$\psi = estat\ mesurat\ o\ que\ tenim$
$\alpha = angle\ que\ formen\ l'estat\ mesurat\ amb\ l'eix\ de\ les\ x$

Per calcular les probabilitats que tenim d'obtenir una mesura hem de fer el producte escalar de l'estat que volem mesurar, en aquest cas $|H>$, per l'estat que tenim i elevar-ho al quadrat. És a dir $P_{\varphi} =< \varphi|\psi >^2$. Si tenim en compte que l'estat que tenim ens ve donat per la següent equació:

$$|\psi> = \cos(\alpha) \cdot |H> + \sin(\alpha) \cdot |V>$$

Podem substituir a l'altre equació per obtenir l'expressió:

$$P_H = < H|\cos(\alpha) \cdot |H > + \sin(\alpha) \cdot |V >>^2$$

Si tenim en compte que $|H> = (1\,,0)$ i que $|V> = (0\,,1)$, aleshores sabem que $< H|H> = 1$ i que $< H|V> = 0$. Per tant:

$$P_H = (1 \cdot \cos(\alpha) + 0 \cdot \sin(\alpha))^2$$

Finalment substituïm α pel seu valor, que en aquest cas com que l'estat mesurat és diagonal $\alpha = 45°$, ja podem fer el càlcul.

$$P_H = \cos(45°)^2$$
$$P_H = \frac{1}{2} = 50\%$$

Com que els angles que hi ha entre les bases $+$ i $\times$ són tots de 45°, aquest càlcul de probabilitats el podem aplicar en tots els casos que veurem.

7.3. Diners quàntics

La primera proposta que hi va haver d'utilitzar les propietats de la física quàntica amb fins criptogràfics va ser la de crear diners quàntics infalsificables. Aquesta idea fou proposada per Stephen Wiesner a finals de la dècada dels 60.

La proposta era fabricar bitllets que tinguessin un número de sèrie i 20 trampes de llum (aparells diminuts que poden contenir un fotó en el seu interior) que continguessin fotons polaritzats en una d'aquestes 4 direccions $\updownarrow, \leftrightarrow, \nearrow, \searrow$. Això comporta que per mesurar-los necessitem mesurar cada polarització amb la base $+$ o $\times$. Per tant que un falsificador hauria d'encertar, per a cada una de les trampes, amb quina base s'ha de mesurar la polarització del fotó. I, si falla, alterarà la polarització dels fotons del bitllet i obtindrà un resultat erroni. D'aquesta forma un banc (que sabria totes les polaritzacions corresponents per cada bitllet) podrà detectar amb facilitat un bitllet fals ja que les polaritzacions no encaixaran amb el nombre de sèrie.

Tot i que Wiesner va fer una bona proposta, ningú se la va prendre seriosament. Això es deu a què mai s'havia proposat res similar i a que Wiesner tot just era un estudiant a la universitat de Columbia (Nova York). A més a més, encara no es disposa de la tecnologia necessària per fabricar aparells que puguin contenir un fotó en un estat de polarització determinada. Per no mencionar que si existissin serien caríssims, i per

tant, no sortiria a compte fabricar bitllets d'aquesta forma. En conclusió, tot i que la proposta de Wiesner és vàlida a nivell teòric, és impossible de posar-la en pràctica.

7.4. BB84

7.4.1. Introducció

L'any 1984, inspirats per la proposta de Wiesner, els científics Charles Bennett i Gilles Bassard van idear una forma d'encriptar els missatges a partir de la polarització dels fotons a la que ens referim comunament com a BB84.

El concepte és el de codificar la informació utilitzant les bases i les polaritzacions mencionades a l'apartat *7.2. Conceptes necessaris* de manera que cada base tingui una polarització 1 i una 0, com per exemple: $\updownarrow = 1, \leftrightarrow = 0, \nearrow = 1, \searrow = 0$ (aquest codi serà l'usat en els pròxims apartats). I, tal i com hem vist a l'apartat *6. El llenguatge dels ordinadors* amb uns i zeros ja podem enviar el que vulguem.

Aquest xifrat té dues peculiaritats: la primera és que només serveix per compartir claus generades de forma aleatòria (que podrem usar després com a claus per el bloc/quadern d'un ús) i que ens permet identificar si una persona està intervenint la nostra línia de comunicació. Degut a aquesta segona característica, el BB84 s'explicarà segons si hi ha o no un espia per poder apreciar la diferència que comporta la presència d'aquest. Ambdós casos s'explicaran a través de l'exemple de l'Alícia en Bob i l'Eva explicat a l'apartat *1. L'Alícia, en Bob i l'Eva*.

7.4.2. Sense espia

En aquest cas l'Alícia i en Bob han de fer tres passos.

1. Imaginem-nos que volen compartir una clau d'uns 5 bits. Per fer-ho l'Alícia ha de compartir amb en Bob una clau codificada en polaritzacions el doble de llarga que la desitjada. Per tant que l'Alícia ha d'enviar a en Bob 10 fotons polaritzats de forma aleatòria amb les bases $+$ i $\times$.
2. Quan en Bob rep els fotons mesura les seves polaritzacions escollint entre la base rectilineal i la diagonal de forma aleatòria i obté una seqüència de bits. Els cops que en Bob encerti la base, obtindrà la polarització i el bit correcte. Els cops que s'equivoqui, obtindrà una polarització incorrecte pel motiu esmentat a l'apartat *7.2. Conceptes necessaris*. Tot i això tindrà un 50% de probabilitats

d'obtenir el bit correcte ja que tal i com s'ha mencionat a l'apartat anterior per cada base hi ha una polarització 0 i una polarització 1.

3. L'Alícia truca a en Bob i li diu les bases que ha fet servir, i en Bob li diu si l'ha encertat o no. I descarten els resultats on les seves bases no hagin coincidit.

D'aquesta forma poden descartar els resultats que poden ser diferents sense revelar la clau. Per aquest motiu, com que aproximadament la meitat de resultats quedaran descartats, la clau que envia l'Alícia ha de ser aproximadament el doble de llarga que la desitjada.

El procés que s'ha descrit es pot veure en la taula següent, on les columnes macades en vermell són els resultats que es descarten i aquells marcats en verd els que conformen la clau final.

Alícia	**Base**	+	×	+	×	×	+	+	×	+	×
	Polarització	↕	↘	↔	↗	↘	↕	↔	↘	↔	↗
	Bit	1	0	0	1	0	1	0	0	0	1
Bob	**Base**	×	+	+	×	×	+	+	+	×	×
	Polarització	↘	↕	↔	↗	↘	↕	↔	↕	↘	↗
	Bit rebut	0	1	0	1	0	1	0	1	0	1

16. Taula que exemplifica els tres primers passos del BB84

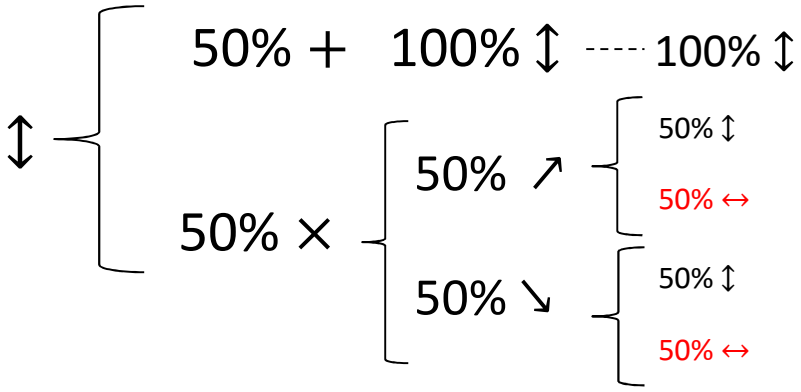
Com podem observar a la taula la clau que després faran servir per comunicar-se amb el bloc/quadern d'un sol ús serà: 010101.

7.4.3. Amb espia

Quan tenim en compte que hi pot haver algú intentant fer-se amb la informació enviada, hem d'afegir un quart pas als tres mencionats a l'apartat anterior. Això es deu a què els fotons enviats per l'Alícia, abans d'arribar en Bob, passen per l'Eva. Aquesta els mesura i torna a enviar allò obtingut a en Bob. Com que l'Eva no sap amb quina base, + o ×, ha polaritzat els fotons l'Alícia, tot el que pot fer és triar bases de forma aleatòria i esperar que hagi encertat. El problema és que si s'equivoca de base, tal i com s'ha esmentat a l'apartat *7.2. Conceptes necessaris*, alterarà el resultat i, per tant, enviarà un bit a en Bob que no serà el que ha enviat l'Alícia. Aleshores, es pot donar el cas que en Bob i l'Alícia hagin mesurat amb la mateixa base, però hagin obtingut resultats diferents.

Per veure l'efecte que ha tingut l'Eva hem d'analitzar el cas on l'Alícia i en Bob han escollit la mateixa base, ja que si n'han escollit de diferents els resultats ja són

descartats al pas 3. I per veure-ho de forma clara, l'esquema que apareix a continuació explica totes les possibles situacions en el cas que en Bob escull la mateixa base que l'Alícia. Els tants per cents a l'esquerre d'una base representen les probabilitats que l'Eva esculli aquella base, els que són a l'esquerre d'una polarització representen les probabilitats que la mesura obtinguda sigui aquesta polarització. Els resultats en vermell són aquells en els que en Bob no obté el mateix resultat que l'Alícia.



17. Esquema de les probabilitats de mesurar certes polaritzacions

A partir de l'esquema podem calcular el tant per cent de resultats obtinguts per en Bob erronis. És a dir, aquells resultats on, tot i haver encertat la base, el bit no és el mateix que ha enviat l'Alícia. Com podem observar, en Bob obtindrà el bit equivocat en dos casos on la probabilitat de cada un és d'un 50% d'un 50% d'un 50%, per tant que per calcular-ho fem: $\left(\frac{50}{100}\cdot\frac{50}{100}\cdot\frac{50}{100}\right)\cdot 2=\left(\frac{125000}{1000000}\right)\cdot 2=\frac{25}{100}=25\%$ de probabilitats d'obtenir un resultat erroni. Aquest 25% d'error és el que ens permet detectar la presència de l'Eva afegint el quart pas mencionat a l'inici de l'apartat:

4. L'Alícia truca a en Bob i acorden comunicar-se aproximadament un 50% dels bits que tenen. Si troben un bit que no coincideix, cancel·len la comunicació ja que és indici que aquella línia de comunicació no és segura. Si tots coincideixen, ja hauran obtingut la clau amb la que poder-se comunicar.

Com que 1 de cada 4 bits no coincideixen, si la meitat són comprovats és altament improbable que no en trobis cap.

Per veure com funcionaria a continuació hi ha una taula que representa on només són contemplats els resultats en què en Bob i l'Alícia han triat bases iguals. Les columnes en verd són els resultats on l'Eva ha encertat la base, les columnes en taronja són els resultats on l'Eva s'ha equivocat de base però en Bob ha obtingut el bit correcte i

aquelles en vermell són els resultats on l'Eva s'ha equivocat de base i en Bob ha obtingut el bit incorrecte.

Alícia	**Base**	+	×	×	+	×	×	×	+	+	+
	Polarització	↔	↗	↗	↕	↘	↗	↘	↕	↔	↕
	Bit	0	1	1	1	0	1	0	1	0	1
Eva	**Base**	×	×	+	+	×	+	+	+	+	×
	Polarització	↘	↗	↔	↕	↘	↕	↔	↕	↔	↗
	Bit	0	1	0	1	0	1	0	1	0	1
Bob	**Base**	+	×	×	+	×	×	×	+	+	+
	Polarització	↕	↕	↗	↗	↘	↘	↘	↕	↘	↔
	Bit rebut	1	1	1	1	0	0	0	1	0	0

18. Taula que exemplifica el quart pas del BB84

7.4.4. Seguretat

En aquest cas només ens hem de centrar en la compartició de la clau, ja que, un cop la clau ha estat compartida la comunicació és 100% segura si s'utilitza el bloc d'un sol ús tal i com s'explica a l'apartat *2.4. Seguretat del bloc/quadern d'un sol ús*.

De fet, el mètode només té un punt que podria posar en dubte la seva seguretat i és el quart pas explicat a l'apartat *7.4.3. Amb espia* ja que l'atzar hi juga un paper. Per veure que aquest fet realment no fa que el mètode sigui insegur ens hem de fer una idea de la mida de les claus que s'han d'utilitzar i de com d'ínfima és la probabilitat que no detectem a l'espia.

Això s'explicarà a través de l'exemple següent: suposem que vols tenir una clau per poder xifrar una sola línia de text. Per fer-ho en primer lloc has de calcular el nombre de bits que hauràs d'enviar. Com que només pots enviar informació en bits, que en necessites 8 per cada caràcter, i que la clau del bloc d'un sol ús ha de tenir els mateixos caràcters que el missatge que vols enviar (en aquest cas una línia té al voltant dels 85 caràcters amb lletra Calibri 12), la clau que necessitaràs constarà de $85 \cdot 8 = 680$. Degut a què, a l'últim pas hi arribaràs amb, aproximadament, el doble de bits que la teva clau, si en descartes la meitat, n'acabaràs comprovant 680 en aquest cas. L'equivalent a comprovar tots aquests dígits i no descobrir-ne cap d'erroni seria com si un crupier barregés 4 cartes: 3 dosos i 1 tres 680 vegades, cada cop en triéssim i que en cap d'aquestes 680 ocasions obtinguéssim el tres.

Quan tenim en compte que 680 és un nombre de bits relativament petits, ja que només ens permetria xifrar una línia de text, podem apreciar realment la magnitud de la improbabilitat que no detectem cap ni un error provocat per l'espia. Això ens permet afirmar que el BB84 és un mètode segur.

8. Conclusions

Per a finalitzar aquest treball ja només fa falta extreure les conclusions tant a nivell general com per a cada apartat. És a dir, obtenir una idea general sobre la criptografia i observar, per a cada subunitat del treball, les idees extretes de l'estudi dels diversos mètodes criptogràfics.

Inicialment hem pogut apreciar com, a grans trets, la criptografia ha anat progressant des de mètodes tant febles com el xifrat Cèsar fins al secret perfecte amb el bloc d'un sol ús. També podem concloure que, aquest mètode, tot i ser indesxifrable, no s'utilitza degut a la seva impracticabilitat. Aquesta es deu, principalment, a tres motius, que són els següents: haver de generar de forma vertaderament aleatòria tants nombres com caràcters que es vulguin enviar, que la clau pugui ser usada solament una vegada i el problema de compartir la clau amb el receptor de forma segura. Tot i això hem pogut veure com es va intentar, sense succeir, automatitzar aquest mètode a través de la màquina Enigma. En aquest punt també podem apreciar la rellevància de la criptografia, ja que va jugar un paper molt important en un esdeveniment tant rellevant com és la segona guerra mundial.

A continuació tindríem la congruència modular. En quant a aquesta, hem demostrat que, tot i ser una relació d'equivalència com la igualtat, té les seves propietats característiques. Per aquest motiu, a l'hora de, per exemple, resoldre una equació que estigui en congruència modular, no podem aplicar els mateixos passos que amb la igualtat. És a dir, no podem fer canvis com el de passar a dividir un factor que està multiplicant ja que només treballem amb nombres enters, no fraccionaris. D'aquest apartat també podem extreure una visió més precisa de les matemàtiques que aquella que normalment tenim. Ja que solem associar matemàtiques amb càlcul, però, la realitat és molt diferent degut a què les matemàtiques solen estudiar el comportament de certes definicions que es fan. I és només quan hem descobert una propietat i l'hem demostrada que la podem usar per a realitzar càlculs.

De la criptografia actual podríem extreure'n una gran conclusió, els xifrats actuals es basen en la dificultat de trobar certs nombres, no en la impossibilitat. És a dir, mentre el bloc d'un sol ús és indesxifrable a nivell teòric, tant l'RSA com ElGamal són xifrats que són indesxifrables si tenim en compte el temps que es tardaria a trobar certs valors amb poder computacional limitat. Això ens porta a concloure que són mètodes amb data de caducitat ja que el més probable és que hi acabin havent ordinadors quàntics, capaços de realitzar aquests càlculs amb un període de temps raonable. A més a més, sempre existeix la possibilitat que es descobreixi algun algorisme que

ofereixi una solució al problema de factorització de nombres grans o al problema del logaritme discret.

D'altra banda tindríem la criptografia quàntica, que ens ha permès apreciar el valor de codis com l'ASCII que ens permeten transcriure un text qualsevol a codi binari. Aquesta, tot i ser molt potent a nivell teòric, encara no es pot posar en pràctica degut a la tecnologia de què disposem actualment. A més a més, és un mètode molt tediós que impossibilita la comunicació immediata. Tot i això, el BB84 ens ofereix una mirada cap al futur al posar sobre la taula la física quàntica com a eina criptogràfica.

Si ens mirem tot allò après sobre la criptografia com un conjunt, veurem que la criptografia és un camp d'estudi canviant. És a dir, com que l'objectiu és el de crear un algorisme que ens permeti comunicar-nos de forma secreta, aquella matèria que estudia és variable. És més, a nivell històric la podem dividir en tres etapes: enginy, matemàtiques i quàntica. En els seus inicis, la criptografia estava conformada per mètodes, com l'escitala, que havien estat concebuts gràcies a l'astúcia. Però, a mesura que ens anem acostant al present, podem observar una tendència a acostar-se a les matemàtiques. De fet, tal i com hem vist en el treball, les matemàtiques són l'eina principal que estudia la criptografia per tal de dissenyar mètodes segurs. Ara bé, si considerem la possible existència (en un futur no molt llunyà) d'ordinadors quàntics, veurem clarament que la criptografia futura es basarà en la física quàntica. En resum, inicialment un bon criptògraf era algú astut, actualment un bon criptògraf és algú que posseeix amplis coneixements sobre les matemàtiques i, futurament, sembla ser que un bon criptògraf serà un bon físic quàntic.

A nivell personal, tot i que inicialment vaig escollir el treball per l'interès que em despertava l'objectiu de la criptografia, m'ha acabat captivant per les seves matemàtiques. És a dir, he trobat que la part matemàtica que hi ha darrere de cada mètode criptogràfic és molt més interessant que el mètode criptogràfic en sí mateix.

Per acabar m'agradaria destacar que totes les matemàtiques que apareixen en aquest treball, a excepció del logaritme normal i la probabilitat, han estat temes nous que he hagut d'aprendre des de zero. També cal remarcar que no tot el treball és teoria que hagi llegit o vist en alguna pàgina web o llibre, sinó que algunes demostracions i la justificació de certes condicions han estat realitzades per jo mateix i corroborades pel tutor de recerca. Per tant que, en conjunt, el treball ha resultat un repte que m'ha exigit molt de temps i esforç. I precisament per això, a resultat ser molt enriquidor i satisfactori elaborar-lo.

9. Fonts d'informació

9.1. Bibliografia

CORBALÁN, Fernando i GÓMEZ URGELLES, Joan. *El lenguaje secreto de los números, codificación y criptografía.* Barcelona: National Geographic, 2014.

SINGH, Simon. *Los códigos secreto.* Barcelona: Círculo de lectores, 2000.

SERRA I ESTRADA, Salvador, ARMENGOL I SOLÉ, Montserrat i MERCADÉ I CAPELLADES, Joan M. *Física 2n Batxillerat*. Madrid: McGraw-Hill, 2018.

MORILLO, Paz i PADRÓ, Carles. *Protocols criptogràfics distribuïts, esquemes per compartir secrets*. Sabadell: Fundació de Caixa Sabadell, 1999.

9.2. Recursos electrònics

CRUISE, Brit. *A journey into cryptography.* <https://www.khanacademy.org/computing/computer-science/cryptography> [Consulta: Juliol 2018]

Departament de Matemàtica Aplicada a les Tecnologies de la Informació i la Comunicació de la Universitat Politècnica de Madrid. *Introducción a la aritmética entera y modular.* <http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/criptografia.html> [Consulta: Juliol 2018]

Universitat Politècnica de Catalunya. *Freqüència de les lletres.* <https://mat-web.upc.edu/fme/codescryptography/CambraNegra/frequencyanalysis.html> [Consulta: Juliol 2018]

SINGH, Simon. *The Enigma Machine explained.* <https://www.youtube.com/watch?v=ASfAPOiq_eQ> [Consulta: Juliol 2018]

LYCETT, Andrew. *Enigma.* <http://www.bbc.co.uk/history/topics/enigma#p00chn61> [Consulta: Juliol 2018]

GIDEON, Samid. *RSA – The Math*. <https://www.youtube.com/watch?v=EOhLZRwxaVo> [Consulta: Agost 2018]

Dr. GRIME, James. *158,962,555,217,826,360,000 (Enigma Machine)*. <https://youtu.be/G2_Q9FoD-oQ>. [Consulta: Agost 2018]

Dr. GRIME, James. *Flaw in the Enigma Code*. <https://youtu.be/V4V2bpZlqx8> [Consulta: Agost 2018]

WILDSTORM, David Jacob. *Introduction to Higher Math.* <http://aleph.math.louisville.edu/teaching/2011SP-311/notes-110209.pdf> [Consulta: Agost 2018]

JANESKO, Jen. *Paper and Pencil RSA (starring the extended Euclidean algorithm).* <https://www.youtube.com/watch?v=kYasb426Yjk&feature=youtu.be> [Consulta: Agost 2018]

ARAGONESES, Andrés. *Criptografía Cuántica (1/2)* <https://www.youtube.com/watch?v=UXm9RhqQmZU>[Consulta: Agost 2018]

Dra. GARCÍA, Alfonsa, Dr. GARCÍA, Francisco i Dr. GARCÍA Jesús. *Lección 2. Criptografía cuántica.* <http://www.criptored.upm.es/crypt4you/temas/cuantica/leccion2/leccion02.html> [Consulta: Agost 2018]

ARTHUT STEIN, William. Elementary Number Theory, A Computational Approach. <https://wstein.org/edu/2007/spring/ent/ent-html/node97.html> [Consulta: Setembre 2018]

10. Annexos

10.1. Programa per a facilitar els càlculs de l'RSA

A l'hora de calcular exemples per l'RSA, els nombres solen ser massa grans com per poder-ho fer amb la calculadora i totes les pàgines web que he trobat tampoc operen amb nombres de tants dígits. Per solucionar aquest problema he escrit un programa informàtic breu per, introduint-hi els valors a, x i n, calcular el valor numèric de la funció següent:

$$f(x) = a^x \ (mod\ n)$$

El programa l'he escrit usant el portal web repl.it i en llenguatge Python 3.6.1 i és el següent:

```
a=int(input("introdueix la base"))
b=int(input("introdueix el valor del mòdul"))
c=int(input("introdueix l'exponent"))
print((a**c)%b)
input()
```

10.2. Programa per a facilitar els càlculs d'ElGamal

Pel mateix motiu i fent servir el mateix portal web i llenguatge esmentat a l'annex *9.1 Programa per a facilitar el càlcul de xifratges RSA*, he escrit un programa informàtic breu pel xifrat ElGamal. Aquest (si tenim en compte que la nomenclatura és aquella definida a l'apartat *5.5.2. Nombres de ElGamal*), introduint-hi els valors g, p, x i y, ens calcula els nombres a, b, h i realitza la operació de desxifrar que realitzaria en Bob perquè puguem comprovar que concorda amb el missatge enviat.

```
p=int(input("introdueix el valor de p"))
g=int(input("introdueix el valor de g"))
x=int(input("introdueix el valor de x"))
y=int(input("introdueix el valor de y"))
m=int(input("introdueix el missatge (un nombre enter)"))
h=((g**x)%p)
a=((g**y)%p)
b=((m*(h**y))%p)
f=(((a**(p-1-x))*b)%p)
print("h =", h)
print("a =", a)
print("b =", b)
print("missatge que obté en Bob:", f)
input()
```