Estudi matemàtic i criptològic: anàlisi de la complexitat d'enigma i comparativa amb l'RSA

Nira Nasr Romero

ESTUDI MATEMÀTIC I CRIPTOLÒGIC: ANÀLISI DE LA COMPLEXITAT D'ENIGMA I COMPARATIVA AMB L'RSA

Nira Nasr Romero
2n de Batxillerat Científic
Anna Mañé Jané i
Ana Ponce León
IES Andreu Nin
El Vendrell
02/12/2021

«Either mathematics is too big for the human mind or
the human mind is more than a machine»
- Kurt Gödel -

AGRAÏMENTS

M’agradaria donar les gràcies, en primer lloc, a les meves dues tutores del treball de recerca; a l’Anna Mañé, per encoratjar-me a seguir endavant amb aquest tema sense restriccions i a l’Ana Ponce, per la seva revisió i consells finals.

En segon lloc, el desenvolupament de la memòria tampoc hauria estat possible sense els ànims constants de la meva mare i la seva inculcació sobre l’excel·lència.

En tercer lloc, voldria agrair enormement a la Universitat Rovira i Virgili el fet d’haver-me seleccionat al programa *Talent Jove* i brindar-me l’oportunitat d’adquirir coneixements d’un nivell superior a l’altura del Batxillerat Internacional, en especial al matemàtic i professor Ramon Masip per les seves classes de probabilitat i combinatòria, que m’han ajudat a verificar la validesa dels càlculs que necessitava fer.

Per últim estic agraïda als professors de l’institut durant la meva etapa a la secundària que m’han inculcat el gust per les matemàtiques, la informàtica i la física, així com a tothom que s’ha interessat i preocupat pel meu treball.

ABSTRACT

Cryptography used to be a very secluded science of little relevance to everyday life, but in the Information Age, it builds the base for the right to privacy and freedom of speech, association and the press.

In this paper I endeavor to study two ciphers: the Enigma machine as the main one, an electromechanical device built in the 20th century, and the RSA, a public-key cryptosystem widely used nowadays, into which we will look secondarily.

The aim is to determine whether or not a secrecy system such as Enigma can be as secure as modern asymmetric ciphers used for data transmission in the *Word Wide Web*, by describing a hypothetical and optimized version heuristically.

In order to accomplish this, a rigorous study on the mathematical and complexity theory principles behind them has to be done, and thus be able to estimate the resources needed to solve the problem proposed.

The overall result also helps us understand why symmetric key algorithms need smaller key lengths compared to public key algorithms.

RESUM

La criptografia solia ser una ciència molt aïllada i de poca rellevància per a la vida quotidiana, però a l'era de la informació, construeix la base del dret a la intimitat, la llibertat d'expressió, associació i premsa.

En aquest treball intento estudiar dos xifratges: la màquina Enigma com a principal, un dispositiu electromecànic construït al segle XX, i l'RSA, un criptosistema de clau pública àmpliament utilitzat actualment, el qual analitzarem de manera secundària. L'objectiu és determinar si un sistema secret com l'Enigma pot arribar a ser tan segur com els xifratges asimètrics moderns utilitzats per a la transmissió de dades a la *Word Wide Web*, mitjançant la descripció heurística d'una versió hipotètica i optimitzada.

Per aconseguir-ho, s'ha de fer un estudi rigorós dels principis matemàtics i de la teoria de la complexitat que hi ha darrere, i així poder estimar els recursos necessaris per resoldre el problema proposat.

El resultat global també ens ajuda a entendre per què els algorismes de clau simètrica necessiten longituds de clau més petites en comparació amb els algorismes de clau pública.

ÍNDEX

VI. ANNEXOS

ÍNDEX DE TAULES I GRÀFIQUES

ÍNDEX DE FIGURES

I. PREÀMBUL

1. INTRODUCCIÓ

La matemàtica és un model de pensament basat en la bellesa de la seva estructura, serietat i potència. De fet aquestes característiques no radiquen en les seves conseqüències, que són simplement l'evidència del seu rigor, o així ho expressava el matemàtic G.H Hardy en la seva apologia.
"La certesa de les matemàtiques - diu Alfred Whitehead - depèn de la seva generalitat completament abstracta".
Molts cops, quan es plantegen i demostren teoremes, amb els quals al final s'obté una extraordinària relació amb el món tangible, gairebé mai es fa tenint present aquestes conseqüències pràctiques, sinó a partir de les idees que connecta. Així doncs, no es podia predir que la teoria de grups, les matrius, la combinatòria, la teoria de conjunts o la teoria de nombres especialment, que són fruit de la matemàtica pura, tinguessin aplicacions tan eficaces en els camps de la física, l'enginyeria, o en el cas que aquí es vol estudiar, en la criptografia.

Quan parlem d'aquesta branca de la matemàtica aplicada, la majoria de gent pensa en el *Bitcoin*, el comerç electrònic o en agències de seguretat, però potser no tenen present que pràcticament des del començament de l'escriptura, els humans han estat escrivint en forma de codi per tal d'amagar els seus secrets, i que l'art de la criptografia ha determinat molts cops el destí de nacions senceres o el transcurs d'una guerra.
Fins fa pocs anys la criptografia potser era interessant per a empreses i governs, però avui dia té un pes immens a l'hora de protegir les nostres dades a la xarxa i de mantenir secrets els nostres missatges.

En aquesta monografia el que pretenc és analitzar el funcionament i estructura de dos sistemes de xifratge en concret: la màquina Enigma, un sistema de xifratge mecànic dissenyat a principis del segle XX, i l'RSA, l'algorisme de clau pública més emprat a la xarxa actualment. Aquest objectiu queda enfocat amb les següents proposicions.

1.1 Hipòtesi

A partir de l'estudi de la complexitat i base matemàtica de dos sistemes criptogràfics, la hipòtesi d'aquest Treball de Recerca és la següent: pot un sistema de xifratge mecànic de clau simètrica com la màquina Enigma ser igual de segur que un dels sistemes de clau pública més emprats a la xarxa actualment, l'RSA? Aquesta hipòtesi sorgeix d'un plantejament inicial corresponent a la pregunta "*de* què depèn la seguretat d'un sistema?¨.

1.2 Objectius

La resolució de les qüestions plantejades en aquesta investigació m'ha portat a proposar-me una sèrie d'objectius a complir:

- Capir el funcionament de la màquina Enigma i com va ser desxifrada i el funcionament de l'algorisme RSA.
- Ser conscient de la importància del desxiframent de la màquina Enigma durant la Segona Guerra Mundial i el canvi de direcció que va produir tant en la criptografia com en la informàtica.
- Comprendre i conèixer els mètodes matemàtics darrere de diferents sistemes de xifratge: teoria de nombres, combinatòria, la probabilitat i estadística, aritmètica modular, les característiques dels nombres primers i comprovar-ne la importància en aquesta branca de la matemàtica aplicada.
- Conèixer la terminologia de la disciplina criptogràfica.
- Aprendre a xifrar i desxifrar missatges mitjançant diferents algorismes.
- Conèixer els mètodes criptoanalítics més significatius.
- Adonar-se de la feblesa de sistemes de xifratge clàssic i com això comporta una millora i evolució progressives.
- Tenir una visió històrica general de la criptografia.
- Saber la diferència entre els esquemes de xifratge simètric i asimètric.
- Expressar matemàticament totes les proposicions i hipòtesis formulades per tal de poder manipular-les com a funcions.

- Familiaritzar-se amb la teoria de complexitat computacional, i entendre el fonament de les màquines de Turing i el problema *P=NP?*
- Conèixer els algorismes que s'utilitzen avui dia a la xarxa.
- Demostrar que la seguretat d'un sistema no depèn exclusivament del nombre de claus possibles, però sí és un factor crucial.
- Ser conscient de la transcendència de la criptografia en el món actual.
- Reflexionar sobre el futur d'aquesta disciplina en relació amb la capacitat de còmput dels ordinadors quàntics i com perillen els sistemes actuals.
- Saber quins són els avantatges i inconvenients de fer servir diferents sistemes criptogràfics.

D'altra banda, com a objectius personals:

- Guanyar destresa amb el llenguatge formal matemàtic.
- Conèixer les tècniques de demostració matemàtiques principals.
- Adquirir coneixements que s'avancin als estudis universitaris.
- Aprendre a organitzar-me el temps.
- Saber buscar fonts d'informació i distingir-ne les essencials.
- Aprehendre els mètodes d'investigació adients i la formalització d'anàlisis.
- Ésser capaç de sintetitzar la informació i d'escollir exclusivament el que és realment important, i de cohesionar-la de manera congruent.
- Poder explicar teoremes i conceptes matemàtics complexos de manera senzilla, entenedora i assequible.

1.3 Estructura del treball i metodologia emprada

Per entendre el desenvolupament d'aquesta memòria cal conèixer l'estructura proposada. El mètode utilitzat ha sigut principalment qualitatiu.
El treball es divideix en 3 blocs principals. El primer estableix una base teòrica i els dos restants arrangen les seves conclusions independents.

1. **Fonaments conceptuals**

   Aquesta part constitueix la base teòrica que permet comprendre les dues posteriors i servir-ne de justificació i referència. S'hi fa una síntesi de les característiques principals de la criptografia i la criptoanàlisi, es donen els fonaments matemàtics necessaris per analitzar aquests dos sistemes i per últim s'explica la importància de la complexitat i el temps computacionals.

2. **Anàlisi de la màquina enigma**

   En aquesta primera anàlisi s'introdueix al lector a la màquina Enigma, es presenta el seu funcionament i es planteja la solució matemàtica que van utilitzar els aliats britànics durant el transcurs de la Segona Guerra Mundial, per després estudiar-ne la seguretat i complexitat. A continuació es pretén optimitzar aquest sistema en un cas hipotètic, fent una simulació de què passaria si s'emprés digitalment en l'actualitat. Per justificar-ho matemàticament, es fan uns càlculs amb els fulls de treball *Sage* de CoCalc®, una plataforma al núvol per a matemàtiques computacionals, que permeten operar amb nombres molts grans i m'han facilitat l'avaluació de les llargades de les claus. Finalment, s'extreuen les conclusions dels resultats obtinguts.

3. **Anàlisi de l'RSA**

   En tercera instància, es dona a entendre el funcionament de l'algorisme RSA i s'estudia en què rau la seva complexitat segons els seus principis matemàtics, per després fer una comparació amb els resultats anteriors i determinar la seva seguretat partint del nombre de bits de solidesa.

2. JUSTIFICACIÓ I MOTIVACIONS

Aquesta monografia sorgeix de l'ambició de fer un treball complex i que anés més enllà de l'ensenyament al batxillerat. A partir de veure la pel·lícula *The Imitation Game*, el personatge d'Alan Turing va anar convertint-se en una proposta rellevant en l'encaminament del treball de recerca.

Inicialment, em vaig proposar estudiar exclusivament la màquina Enigma i com va ser desxifrada. A mesura que vaig anar avançant en la recerca, vaig descobrir dues branques magnífiques que vaig tenir clar que volia incloure: la primera, de la mà de Turing, em va apropar a la teoria de la complexitat computacional i a les màquines de la ment. Les màquines de Turing van proporcionar en la seva època els esbossos del que seria un ordinador, al mateix temps que pretenia resoldre els teoremes de Gödel. La segona se'm va presentar mentre investigava sobre la branca de la criptografia. L'aparició dels sistemes de clau pública va ser una gran revolució que utilitzava com a fonament una de les part més importants de la matemàtica; la teoria de nombres. Tot i que en un principi pretenia fer-ne una implementació amb codi, aviat vaig veure que aquesta pràctica no tindria res d'especial. De manera progressiva, l'anàlisi de la complexitat va anar guanyant pes i extensió fins al punt de relegar la programació als annexos i centrar-me en l'anàlisi teòrica i en les seves estructures matemàtiques. És en aquest moment que vaig tenir clar que volia desembocar aquests dos vessants cap a una mateixa hipòtesi, la qual em permetia enfocar un treball sobre criptografia de manera més original i diferent. Plantejar el problema d'aquesta manera m'ha permès cohesionar una sèrie de conceptes pels quals tenia molt d'interès i que d'altra manera no hagués tingut sentit ajuntar.

Aquesta és doncs una recerca inclinada cap a la part més abstracta i conceptual de la temàtica a tractar, motivada pel meu interès a cursar matemàtiques durant l'etapa universitària, i que m'ha facultat amb coneixements sobre llenguatge i demostracions matemàtics més avançats. En alguns moments, però, la complexitat de la qüestió a tractar superava els límits conceptuals a comprendre i per això es remet al lector a l'abundant bibliografia. Les matemàtiques donen la possibilitat de sortir de nosaltres mateixos, veure el món d'una manera més objectiva i predir el seu comportament. Són una part fonamental de la nostra societat i això és el que pretenc reflectir.

II. FONAMENTS CONCEPTUALS

1. CRIPTOLOGIA

La criptologia és la ciència que estudia l'escriptura secreta. Dins d'aquesta distingim tres branques principals, representades a la Figura 1. Els conceptes inclosos s'aniran definint al llarg d'aquest primer bloc teòric, centrat exclusivament en la criptografia i la criptoanàlisi.

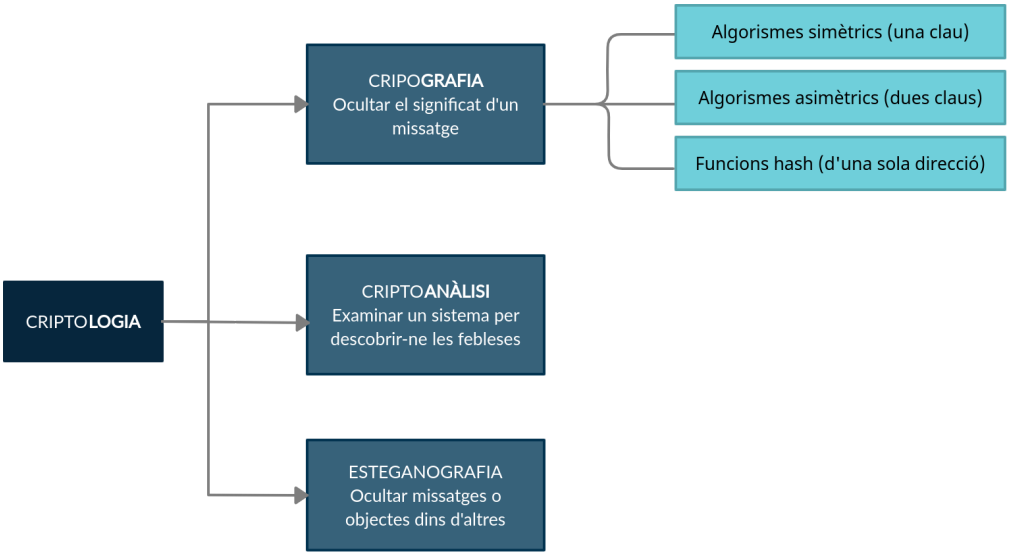


Figura 1. Esquematització de la criptologia. Font pròpia.

La paraula criptografia prové del grec *κρυπτός* (kriptos), que significa ocult, i *γραφή* (graphos), que es tradueix com escriure. La seva definició clàssica és "l'art d'escriure missatges en clau secreta o enigmàticament". Va començar a ser considerada una ciència aplicada a mitjans del segle XX, atenent la seva relació amb les branques de la teoria de la informació, la teoria de nombres, l'estadística, la teoria de la complexitat i per últim les tecnologies de la informació i de les comunicacions. Podem dir també que és l'estudi de sistemes matemàtics que solucionen dos problemes: privacitat i autentificació. Aquests sistemes prevenen l'extracció de la informació per part de tercers no autoritzats quan els missatges s'han de transmetre per un canal públic, i així assegurar que únicament són llegits per al receptor intencionat.

La criptoanàlisi, per contra, és l'estudi de tècniques eficients per vulnerar la seguretat dels mètodes criptogràfics. Criptoanalitzar un sistema vol dir examinar-lo exhaustivament, sovint des d'un punt estadístic o matemàtic, per trobar els punts on falla la seguretat i proposar mètodes per endevinar el missatge original sense tenir accés a la clau.[1]

La protecció de la informació que es persegueix en la criptografia es porta a terme variant la seva forma. Per referir-nos al procés criptogràfic més endavant cal descriure'l tècnicament.

Anomenem **xifratge** a una transformació del text original, que el converteix en un text xifrat o **criptograma**. Analògicament, anomenem **desxifrat** a la transformació que permet recuperar el text original a partir del text xifrat.

Cadascuna d'aquestes transformacions està determinada per un paràmetre anomenat **clau.** El conjunt dels seus possibles valors s'anomena **espai de claus K.** La família de transformacions es denomina **sistema criptogràfic** $\mathbf{S=\{S_k / k\in K\}}$**,** denotant-se amb $\mathbf{S_k}$ l'operació de xifratge i com a $\mathbf{D_k}$ **(o** $\mathbf{S_k^{-1}}$ **)** el desxifratge amb la clau **k.** L'espai de missatges originals es denota en la imatge com a **P** (en anglès, *plaintext*), mentre que ens referim al conjunt de textos xifrats com a **C** (en anglès, *ciphertext*).

C es trobarà aplicant a l'operació $\mathbf{S_k}$, i si volem recuperar **P**, aplicarem el procediment invers o $\mathbf{D_k}$ al text xifrat **C**.

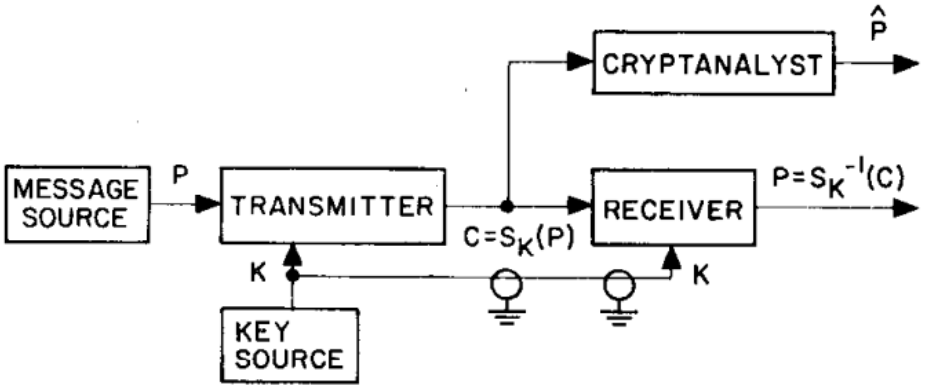


Figura 2. Flux d'informació en un sistema criptogràfic convencional. Font: Diffie and Hellman: New directions in cryptography. IEEE 1976

[1] Juher, D. L'art de la comunicació secreta. El llenguatge de la criptografia, 2004

1.1 Criptografia simètrica

La criptografia simètrica, clàssica o de clau secreta és considerada tan antiga com el llenguatge. Les primeres civilitzacions van desenvolupar mètodes més o menys enginyosos per restringir l'accés a la informació.[2] Podem pensar en la criptografia com un calaix tancat amb clau que conté un missatge secret. Qualsevol persona que vulgui trobar-lo, li caldrà tenir aquesta clau, és a dir, la clau per xifrar el missatge és la mateixa que per desxifrar-lo; anàlogament, el receptor haurà de saber quina clau ha utilitzat l'emissor. Aquest esquema es mostra en la Figura 2

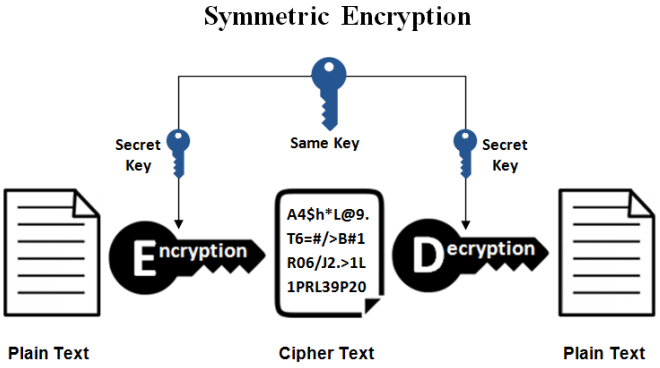


Figura 3. Criptografia simètrica. Font: medium.com

La criptografia simètrica està basada en principis matemàtics i criptogràfics que normalment empren simples primitives com la substitució, la rotació i la permutació. Molts d'aquests sistemes serveixen de base per a molts protocols moderns, com els xifratges en bloc *AES* i *DES*.[3]

Normalment, aquest tipus de xifrat és susceptible a atacs d'anàlisi de freqüència o de força bruta, per això cal que la llargada de la clau sigui prou extensa (al llarg d'aquesta primera part s'aniran aclarint aquests conceptes). Tot i això, solen ser segurs i l'únic inconvenient que presenten és el fet d'haver d'acordar la clau abans. Aquest problema es resol amb l'aparició de la criptografia asimètrica a finals del segle XX.

[2] González Vasco & Pérez del Pozo. *Introducción a la criptografí*a. 2021.

[3] Advanced Encryption Standard i Data Encryption Standard respectivament. El DES ja no s'utilitza per considerar-se insegur (només utilitza una clau de 56 bits), mentre que l'AES s'utilitza en els esquemes WPA2 (la Wi-Fi) i utilitza claus de 128 bits.

1.1.1 AES

L'AES és un sistema criptogràfic de clau simètrica consistent en un xifratge de blocs. És essencial en la protecció d'informació classificada en governs (és l'estàndard dels Estats Units), plataformes a la xarxa com les aplicacions *Whatsapp* o *Telegram*, grans empreses com Google i software d'encriptació de discs com *BitLocker,* de Microsoft.
El va desenvolupar el NIST[4] el 1997 com a alternativa al *DES*. Actualment trobem tres versions: *AES-128* (10 rondes)*, AES-192* (12 rondes) i *AES-256* (14 rondes).
L'AES funciona realitzant transformacions a dades emmagatzemades en una matriu. Dins d'aquesta, les transformacions de xifratge es repeteixen en diverses rondes. L'alterament de cada ronda es fa amb substitucions, desplaçaments i barreges de columnes. Com més llarga és la clau, més rondes necessita per completar-se.
Cadascun xifra blocs d'informació de 128 bits amb longituds de claus diferents, esmentades anteriorment, que es mostren a la Taula 1.

Longitud de la clau	Combinacions possibles		32 bits	4.2×10^{9}
1 bit	2		56 bits	7.2×10^{16}
2 bits	4		64 bits	1.8×10^{19}
4 bits	16		128 bits (AES)	3.4×10^{38}
8 bits	256		192 bits (AES)	6.2×10^{57}
16 bits	65536		256 bits (AES)	1.1×10^{77}

Taula 1. Nombre de combinacions en funció del nombre de bits (longitud de la clau)

1.2 Criptografia asimètrica

Abans de 1977, qui volgués enviar un missatge secret es trobava amb un problema substancial: abans de transmetre el missatge, remitent i destinatari havien de decidir quin sistema de codificació adoptarien i la clau que utilitzarien. Podem imaginar les dificultats logístiques que sorgirien si haguéssim d'usar un sistema de criptografia d'aquest estil per comprar per Internet. Abans que poguéssim enviar les nostres informacions bancàries amb seguretat, les empreses que gestionen els llocs d'Internet

4 National Institute of Standards and Technology

ens haurien d'enviar una carta protegida per explicar-nos com codificar la informació[5]. Donat l'enorme trànsit de dades, hi hauria altíssimes probabilitats que moltes d'aquelles cartes acabessin per ser interceptades. Es feia imprescindible, als inicis de l'era de les comunicacions ràpides, desenvolupar un sistema criptogràfic adaptat a les noves necessitats. I de la mateixa manera que, durant la guerra, eren els matemàtics de Bletchley Park els que van desxifrar Enigma, serien els matemàtics els qui crearien una nova generació de codis que han afavorit el naixement de la que avui es coneix amb el nom de criptografia de clau pública o criptografia asimètrica .[6] Aquest esquema va ser proposat per primer cop en 1976, en un important article científic escrit pels matemàtics de la Universitat de Stanford, a Califòrnia, Whit Diffie i Martin Hellman, anomenat *New directions in cryptography.*

La simple idea de funcionament va ser revolucionària, pel fet d'utilitzar dos parells de claus: un parell consistia en la clau pública, que podia ser coneguda per qualsevol, i l'altre parell era secret, només conegut pel remitent. Quan un usuari vol trametre un missatge a un altre, posem per cas el número de targeta bancària, per encriptar-lo usarà la clau pública del destinatari, i només aquest últim, amb la clau privada, el podrà desencritpar. Aquesta generació de claus depèn d'algorismes basats en problemes matemàtics coneguts com a funcions d'una sola direcció: és molt fàcil, donat un *input* o entrada, computar un resultat, però, en canvi, revertir l'operació és molt més difícil i poc pràctic. La Figura 3 mostra esquemàticament el procés d'encriptació mitjançant el sistema de clau pública.

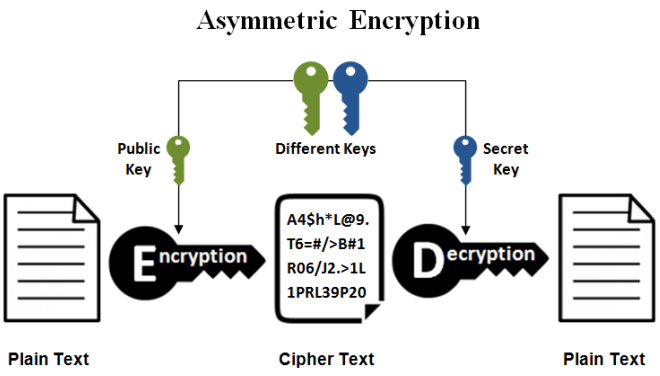


Figura 4. Criptografia asimètrica. Font: medium.com

[5] De fet durant els inicis de la criptografia a la xarxa, quan només era utilitzada per governs i agències de seguretat com la NSA (l'Agència de Seguretat Nacional), el remitent havia d'enviar al destinatari les claus.

[6] Du Sautoy, M. *La música de los números primos*. 2007

El fet que un dels parells de clau sigui públic no compromet en cap sentit el text xifrat, ja que el que s'ha de mantenir en secret és la clau privada.

Això suposa una sèrie d'avantatges respecte al sistema anterior, primer perquè és més segur i confidencial, i segon perquè també permet autenticar (es pot garantir que el remitent és qui diu ser perquè només ell coneix la seva clau privada, amb la qual també es poden xifrar missatges i ser desxifrats amb la clau pública, és a dir, a l'inrevés (això ens assegura que només aquella persona els ha pogut enviar).[7]

No obstant també es presenten una sèrie d'inconvenients: la clau ha de ser molt més llarga, el text xifrat ocupa més espai que l'anterior i el procés de xifratge és més lent.

Tant Diffie com Hellman estaven convençuts que les seves idees havien de ser públiques i no ser propietat exclusiva del govern. Era una innovació, almenys en la teoria, però després d'alguns intents infructuosos, molts començaven a dubtar de la possibilitat de dissenyar un algorisme que satisfés aquestes proposicions, fins que el 1977, Ron Rivest, Adi Shamir i Leonard Adleman, del Massachusetts Institute of Technology (MIT), van trobar una solució; l'RSA, un algorisme la base del qual és el producte de dos nombres primers molt grans.

1.3 Trobant els punts febles

Un dels axiomes sobre els quals es basa aquesta disciplina és que sempre hi haurà algun agent extern, és a dir, que no és ni l'emissor ni el receptor, que podrà ser capaç d'interceptar la comunicació i voler desxifrar el missatge. Aquí és on entra en joc la criptoanàlisi, que, dins la criptologia, el que pretén és justament el contrari que la criptografia: trobar mètodes per endevinar la clau sense tenir-la.

Podem dir que la criptografia i la criptoanàlisi són disciplines duals o contràries; res més lluny de la veritat. Cada cop que un matemàtic proposa un sistema, apareixen articles científics en què altres expliquen els possibles atacs al sistema ideat. Aquesta lluita entre criptògrafs i criptoanalistes és fonamental per l'avenç de la criptografia, sempre que es dugui a terme de manera pública i transparent.

[7] Tot aquest procés l'hem de visualitzar via comunicació mitjançant una xarxa web.

Hi ha 4 situacions en què es pot trobar un criptoanalista a l'hora d'atacar:

1. Atac només amb text xifrat: Aquesta és la pitjor situació possible pel criptoanalista, ja que només coneix el criptograma.
2. Atac amb el text original conegut: En aquesta situació el criptoanalista té informació d'una part o de tot el text inicial i també el xifrat. Podem trobar aquesta situació quan es coneix el tema del qual parla el missatge perquè això crea una correspondència entre les paraules més habituals i els texts xifrats més repetits.
3. Atac amb text original escollit: Ara l'enemic pot obtenir a partir del criptograma que intenta desxifrar, el xifratge de qualsevol altre text que ell esculli.
4. Atac amb text xifrat escollit: aquí l'enemic pot obtenir el text original corresponent a determinats textos xifrats de la seva elecció per poder fer-ne les comparacions estadístiques pertinents i detectar patrons que li permetin desxifrar el codi.

D'altra banda, el procés constant d'evolució en les tècniques criptoanalítiques ha resultat en nombrosos mètodes específics per a cada tipus d'algorisme. En aquest cas, però, ens interessen només 3:

1. **Força bruta o *Bruteforce*:**

Aquest és un dels mètodes més antics i el primer al qual per lògica s'acut. Consisteix en provar totes les claus possibles. Si un sistema pot trencar-se fàcilment mitjançant aquest mètode, llavors no és gens segur. Cal trobar claus llargues que no sigui computacionalment possible haver de provar-les totes, però tampoc que siguin d'una llargada massa extensa i aleshores es trigui molt de temps en el procés de xifratge.

2. **Anàlisi de freqüència:**

Aquest tipus d'atac pertany a la família d'atacs analítics, on es pretén buscar una errada o feblesa fonamental en la base matemàtica del sistema. En els mètodes de substitució, entre d'altres, els textos xifrats no solen ser només galimaties. Quan es fa un simple xifratge monoalfabètic (cada lletra es converteix en una altra independentment), es pot analitzar la distribució de freqüències i comparar-la amb les freqüències mitjanes de les lletres d'un idioma, el que permet deduir quina lletra es correspon amb cadascuna.

Posem un exemple:

Tenim el text "*Treball de recerca"*. El codifico utilitzant un sistema de substitució simple la clau del qual és "dyxwvutsrqponmlkjihgfezcba", és a dir, la lletra *a* la substituiré per la *d*, la *b* per la *y*, la *c* per la x, i així successivament. Em queda: "Givydoo wv ivxvixd".

Una tercera persona que volgués desxifrar-lo utilitzant aquest mètode, primer calcularia la freqüència relativa de les lletres d'aquest text, com es mostra a la Figura 4.

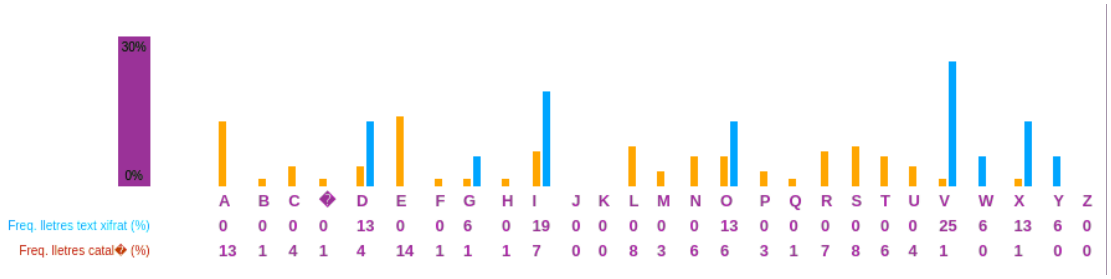


Figura 5. Anàlisi de la freqüència relativa del text d'exemple proposat. Web utilitzada: https://web.mat.upc.edu/fme/codescryptography/CambraNegra/letterfrequency.htm

A continuació, faria la comparació amb les dades oficials de la distribució de freqüències de la llengua emprada, en aquest cas el català, a partir de la Figura 5.

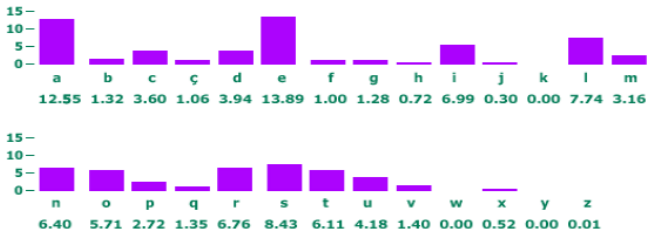


Figura 6. Freqüències mitjanes de les lletres en la llengua catalana. Font: Institut d'Estudis Catalans

Ràpidament veuria que la lletra que més apareix al text xifrat és la *v*, de manera que l'associa amb la lletra amb més freqüència de la llengua, la e. A partir d'aquí aniria provant i substituint i fàcilment pot arribar a deduir el text.[8]

En un cas així, on les permutacions de l'alfabet (26 lletres) són tantes com el factorial de 26 (26!), surt a compte fer servir aquest mètode.

[8] L'exemple utilitzat és molt senzill i només serveix com a guia. En els casos reals no s'usen variables tan simples. *Veure apartat III, punt 3.*

Si per exemple utilitzéssim el xifrat de Cèsar (un mètode molt conegut, mostrat a la figura 6), com només hi poden haver 26 permutacions possibles, és molt més ràpid utilitzar la força bruta ja que la llargada de la clau és ínfimament petita.

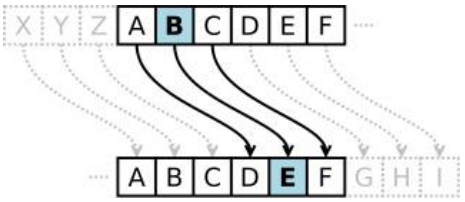


Figura 7. Xifrat de Cèsar.

3. **Atacs de cerca de clau:**

Aquest mètode s'utilitza per a sistemes de clau pública Consisteix, donada la clau pública, trobar la clau privada a partir de factoritzar-la (més endavant s'exemplificarà amb l'RSA). Ja hem vist que en ser funcions d'una sola direcció, procedir a l'inversa té un cost computacional alt.

1.3.1 Principis de Kerckhoffs

Kerckhoffs (1883) va proposar una sèrie de principis relatius a les propietats òptimes d'un sistema. Aquestes característiques ajudaran més tard a analitzar la seguretat dels dos sistemes a estudiar. Les que destaquem i serà necessari tenir en compte són:

1. La seguretat d'un sistema no ha de dependre de mantenir secret el text xifrat sinó de mantenir secreta la clau. No ha de ser un problema que el sistema caigui en mans de l'enemic.
2. En cas que el sistema no sigui matemàtica o teòricament irrompible, ho ha de ser en la pràctica.
3. El sistema ha de ser fàcil d'utilitzar.
4. El sistema ha de ser operable per una única persona
5. La clau ha de ser fàcil de memoritzar i de comunicar, sense necessitat d'haver d'escriure-la, i poder-se modificar.

Per antonomàsia, el segon principi és el que anomenem principi de Kerckhoffs. És el més destacat i segueix sent la base dels algorismes d'avui dia: la seguretat ha de recaure en la privadesa de la clau, no la del sistema.

1.3.2 El secret perfecte de Shannon

Claude Shannon (1916-2001) va ser el matemàtic pare de la teoria de la informació, fonamentació de la comunicació a la xarxa avui dia. Va explicar la relació entre aquesta i la criptografia en els seus treballs [SHA48] i [SHA49][9]. Des de llavors s'utilitza la seva nomenclatura per a l'estudi teòric de la criptografia.

En ambdues ciències l'objecte d'estudi és la informació i la seva protecció, però en la teoria de la informació s'analitza el procés de transmissió sorollosa d'un missatge mentre que en la criptografia s'estudia el procés de xifratge d'un text.

En el primer cas, la distorsió del missatge no és intencionada perquè és deguda al canal. Per això s'ha de fer una descripció probabilística d'aquest canal sorollós per poder rebre la informació de la manera més clara possible. En l'altre cas, es tracta de fer incomprensible el missatge de manera que l'enemic no el pugui conèixer.

Cal matisar llavors la diferència entre codificar i encriptar, que molts cops causa confusió. Codificar vol dir transformar el text en un altre de resultant només perquè sigui possible la seva transmissió en diferents vies, mentre que encriptar té com a objectiu ocultar-lo i fer-lo inintel·ligible.

La teoria de Shannon permet establir una sèrie de regles generals que han de complir els sistemes en un cas ideal o teòric en què són "perfectes".

"El "secret perfecte" es defineix exigint a un sistema que després que un criptograma sigui interceptat per l'enemic, les probabilitats *a posteriori* que aquest criptograma representi diversos missatges siguin idèntiques a les probabilitats *a priori* dels mateixos missatges abans de la intercepció. " *C. E. SHANNON, Communication Theory of Secrecy Systems, 1949, pg 659.*

[9] Shannon, C. E., "A Mathematical Theory of Communication," Bell System Technical Journal, July 1948, p.379; Oct. 1948, p.623.

La síntesi d'aquest problema es pot expressar amb el teorema de Bayes:

$$P_E(M) = \frac{P(M)\, P_M(E)}{P(E)}$$

En què

P(M) = Probabilitat *a priori* del missatge M.

P_M(E) = Probabilitat condicionada del criptograma *E* si el missatge *M* està escollit i.e la suma de les probabilitats de totes les claus que produeixen el criptograma *E* a parir del missatge *M.*

P(E) = Probabilitat d'obtenir el criptograma *E* en qualsevol cas.

P_E(M) = Probabilitat *a posteriori* del missatge *M* si s'obté el criptograma *E*.

Amb altres paraules, a partir d'un missatge xifrat K, totes les claus han de ser equiprobables.[10] El text xifrat no pot donar cap pista sobre l'original i el conjunt ***M*** de tots els missatges possibles, amb sentit o sense, ha de constituir una distribució de probabilitats idèntiques.

Per fer que el problema sigui matemàticament tractable assumirem que el l'enemic sap el sistema que s'utilitza. És a dir, coneix la família de transformacions i les probabilitats d'escollir diverses claus. Es podria objectar que aquesta suposició no és realista, ja que sovint el criptoanalista no sap quin sistema s'ha utilitzat ni les probabilitats en qüestió, però en tot cas ens assegura un disseny el més òptim possible.

D'altra banda, Shannon dona les indicacions a l'hora d'avaluar la seguretat d'un sistema criptogràfic. Aquestes indicacions s'utilitzaran en el marc analític.

1. ***Quantitat de privadesa o secret:*** Els criptogrames no tenen una única solució i la quantitat d'informació no fa més senzilla la tasca de desxiframent. Hi ha molta labor per tal d'efectuar aquesta solució i la quantitat de material necessària per fer-ho és materialment poc probable d'aconseguir. Això permet evitar atacs analítics.

[10] En aquest apartat s'ha obviat gran part de la teoria de probabilitat perquè no és estrictament necessari saber-ho per al transcurs del treball. Es remet al lector a l'article de la cita 9 per a més detall.

2. ***Mida de la clau:*** La clau no ha de ser llarga perquè s'ha de transmetre fàcilment. No s'ha de confondre aquesta característica amb la quantitat de claus possible, que per contra sí volem que sigui un nombre molt alt per evitar atacs per força bruta.
3. ***Complexitat de les operacions de xifrat i desxifrat:*** Xifrar i desxifrar hauria de ser el més senzill possible. Si es fa manualment, la complexitat comporta pèrdues de temps, errors, etc. Si es fa mecànicament, la complexitat condueix a màquines més dificultoses.
4. ***Propagació d'errors*****:** En alguns casos, un error de transmissió o de xifratge d'una lletra pot causar una propagació d'error elevada en el desxiframent, i provor una pèrdua d'infromció.
5. ***Expansió del missatge:*** S'ha d'evitar que el tamany del missatge augmenti significativament quan s'aplica l'operació de xifratge.

2. FONAMENTS MATEMÀTICS

Per afrontar l'estudi d'aquests dos sistemes criptogràfics de manera seriosa és imprescindible manipular amb soltesa algunes nocions matemàtiques.

2.1 Estudi d'algorismes i funcions

Def.1 [Funció]. La idea subjacent del concepte de funció és la d'un mecanisme que permet assignar a cada element d'un conjunt inicial, anomenat domini de la funció, un únic element d'un altre conjunt final, anomenat codomini de la funció. És una relació entre paràmetre i resultat. Sigui f una funció. Escriurem $f : A \longrightarrow B$ per indicar que A és el domini de f i B és el seu codomini. Si un element a del domini es transforma en altre element b del codomini, diem que b és la imatge de a en f i escrivim $b = f(a)$. D'altra banda hem d'especificar que un algorisme no és el mateix que una funció. Entenem per algorisme un conjunt d'instruccions per dur a terme un procés de càlcul o de resolució d'un problema.

Per seguir amb fluïdesa els següents apartats, cal fer un repàs de les funcions elementals que s'empraran. La Figura 7 mostra esquemàticament els seus comportaments.

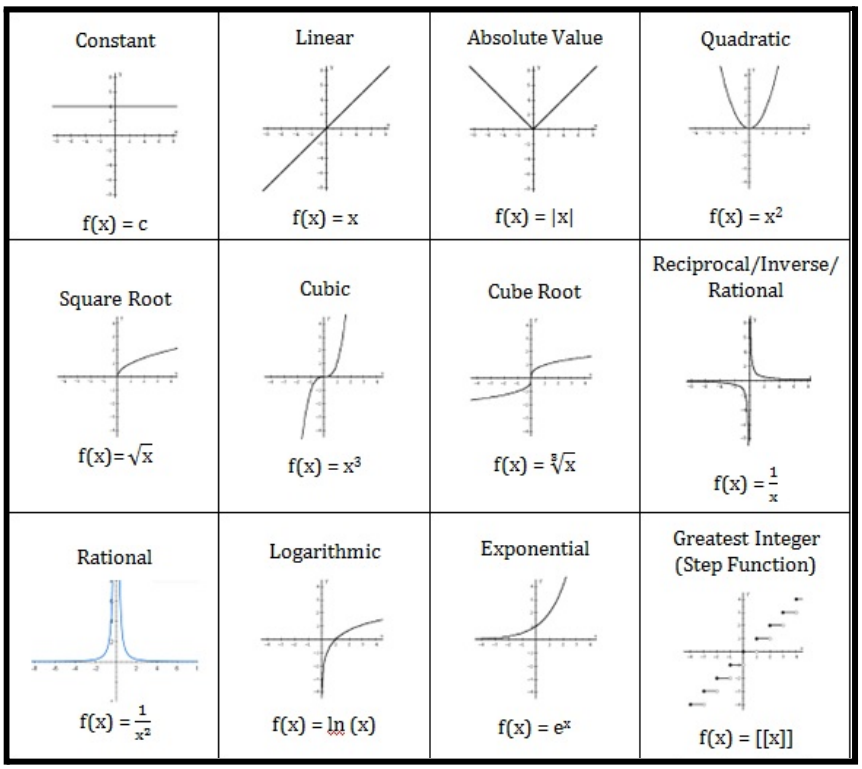


Figura 8. Funcions elementals.

2.2 Combinatòria

Def.2 [Combinatòria]. La combinatòria és una branca de la matemàtica discreta que en essència estudia les agrupacions que es poden formar amb *n* elements, sense tenir en compte l'ordre, a les que anomenem *combinacions*. L'expressió numèrica d'aquestes combinacions s'anomena nombre combinatori o coeficient binomial.

Hi ha diverses fórmules que permeten el càlcul de les variacions, permutacions o combinacions possibles en funció de les propietats dels nombres combinatoris. En alguns casos sí és lògic tenir en compte l'ordre, però aleshores cal recordar que no parlem de combinacions. La taula 2 expressa matemàticament aquestes agrupacions, que caldrà tenir presents per després posar-les en pràctica.

AGRUPACIONS	SENSE REPETICIÓ	AMB REPETICIÓ
VARIACIONS: Es prenen alguns elements. Importa l'ordre.	$V_{n,r} = n(n-1)(n-2)\dots(n-r+1)$ $V_{n,r} = \frac{n!}{(n-r)!}$	$VR_{n,r} = n^r$
PERMUTACIONS: Es prenen tots els elements (n = r). Importa l'ordre.	$V_{r,r} = P_n = n! = n(n-1)(n-2)\dots 1$ *on n! = factorial de n*	$P_n^{a,b,c,} = \frac{n!}{a!b!c!}$
COMBINACIONS: És el nombre de variacions entre el nombre de permutacions. No importa l'ordre.	$C_{n,r} = \frac{n!}{r!(n-r)!}$ *on* $C_{n,r}$ és un nombre combinatori i es llegeix "n sobre r"	

Taula 2. Fórmules combinatòries. Font pròpia.

Al nombre combinatori o coeficient binomial l'entenem com el valor numèric de les combinacions ordinàries (sense repetició) d'un conjunt de *n* elements agafats en grups de *r*, tals que $n,r \in \mathbb{N}$.

Els problemes combinatoris són molt presents en la topologia, la informàtica, la teoria de probabilitats i la física estadística. Una de les seves aplicacions principals és l'optimització, branca de les matemàtiques aplicades i ciències de la computació relacionada amb la investigació operativa, teoria d'algorismes i teoria de la complexitat computacional. L'optimització combinatòria és coneguda pel seu plantejament molt senzill però de difícil resolució.

2.3 Els nombres primers, els àtoms de l'aritmètica

«Els nombres primers són els autèntics àtoms de l'aritmètica. Es defineixen com primers els nombres enters indivisibles, és a dir, els que no es poden expressar com a producte de dos enters menors.» (du Sautoy, 2007).

En aquest apartat introductori, només es vol fer incís en el que fa especials als nombres primers entre tota la resta: Els nombres primers tenen el seu origen en els mateixos inicis dels sistemes de numeració, i molt aviat van destacar a causa del seu comportament poc predictiu. Euclides, Fermat, Euler, Gauss, Riemann, Ramanujan i una llarga llista dels matemàtics més importants de la història van intentar posar ordre en l'estructura dels nombres primers sense èxit, només per l'afany de trobar un patró.

A partir del seu estudi, van sorgir múltiples teoremes i propietats, que han passat de ser simples curiositats matemàtiques a formar part del nostre dia a dia i ser de gran utilitat. Són aquest caràcter tan poc predictiu, la falta de coneixement profund que tenim actualment sobre ells i alhora la seva importància com a fonament numèric els que permeten que puguem utilitzar-los per la protecció de les nostres dades a la xarxa, complementant els sistemes de clau pública més emprats.

La funció Z de Riemann (Figura 9), com a exemple més significatiu, és fruit de l'evolució de mètodes que pretenien trobar els nombres primers compresos entre 1 i *x* en funció de *x.* És d'una importància eminent en el món de la física i les matemàtiques i la seva demostració oficial produiria avenços inimaginables.

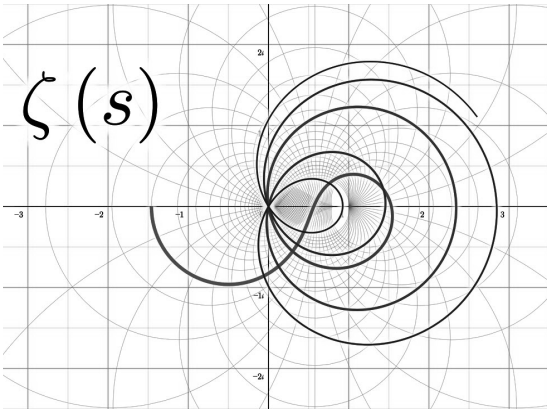


Figura 9. Funció Z de Riemann

2.3.1 La conjectura de Goldbach

Christian Goldbach (1690-1764) va ser un matemàtic prussià molt proper a Euler que va demostrar que <<tot nombre parell major que 2 pot ésser escrit com la suma de dos nombres primers.>> Per exemple: 4 = 2+2, 8 = 3+5, 14= 3+11. Euler va comprovar-ho fins al nombre 2500. En l'actualitat, ha sigut verificada informàticament per tots els nombres parells menors que dos mil bilions, però la conjectura encara no ha sigut comprovada i és considerada com un dels problemes més difícils de la ciència.

2.4 Teoria de nombres

La teoria de nombres constituïa, segons Gauss, la reina de les matemàtiques, a la qual considerava la reina de les ciències. Des del sorgiment de la criptografia de clau pública, la teoria de nombres és molt present a la nostra garantia de privacitat d'una manera sorprenent. Exposaré d'una manera molt visual alguns dels teoremes bàsics d'aquesta branca de la matemàtica pura per tal d'utilitzar-los com a justificació més endavant.

2.4.1 Teorema fonamental de l'aritmètica

El teorema fonamental de l'aritmètica verifica que tot enter positiu n tal que $n > 1$ pot ser expressat com un únic producte de potències de nombres primers.

$$N = p_1^{e_1} \; p_2^{e_2} \ldots \; p_r^{e_r},$$

on p és un nombre primer, e el nombre de vegades que apareix i r el nombre de divisors diferents.

2.4.2 Identitat de Bézout

[Teorema] Donats dos enters $a, b \in \mathbb{Z}$ tals que el seu màxim comú divisor sigui d ($\text{mcd}(a,b) = d$), aleshores existeixen $x, y \in \mathbb{Z}$ tals que

$$ax + yb = d$$

2.4.3 Algoritme estès d'Euclides

L'algorisme estès d'Euclides permet calcular, de manera eficient, el *m.c.d* de dos nombres enters, sense necessitats de factoritzar-los. La seva versió estesa proporciona una identitat de Bézout entre ells, que pot utilitzar-se, entre moltes altres finalitats, per calcular inversos en aritmètica modular.[11]

{Pseudocodi. *Algorisme* [estès d'Euclides]}

Entrada: a, b, enters amb a>b>1
Sortida: (d,u,v), on d = m.c.d(a,b) i d = a·u + b·v

r[0]:= *a*; r[1]:=b ; //*Denotem amb ":=" un operador d'assignament, diferent de "="*

s[0]:= 1; s[1]:=0 ;

t[0]:= 0; t[1]:=1 ;

i :=1 ;

Mentre r[i]: ≠ 0 fes

q[i]= int(r[i]/r|i-1]) ; // int és la part sencera d'un nombre

r[i+1l:= r[i-1] - r[i]q[i] ;

s[i+1]:= s[i-1] - s[i]q[i] ;

t[i+1]:= t[i-1] - t[i]q[i] ;

i := i+1 ;

Repeteix mentre sigui cert;

Sortida: (r[i-1], s[i-1], t[i-1])

[11] Veure Annex VI per una millor interpretació del codi.

2.5 Matemàtica discreta: aritmètica modular

2.5.1 El rellotge de Gauss

Gauss (1977) va introduir la idea d'aritmètica modular o aritmètica del rellotge. Els estudis de Gauss sobre aquesta nova aritmètica van revolucionar la matemàtica de principis del segle XIX. Això va ajudar a molt matemàtics a descobrir en l'univers dels nombres estructures que havien estat ocultes durant generacions. Avui dia, l'aritmètica modular de Gauss és fonamental per la seguretat a Internet.

Aquesta calculadora de rellotge permetia fer càlculs amb nombres que en aquella època es consideraven inabordables. El funcionament d'aquesta calculadora era el mateix que el d'un rellotge normal, i de fet molts dels càlculs que fem intuïtivament es basen en aquest principi: imaginem un rellotge de 12 hores convencional. Quan diem que són les setze hores, volem dir que són les quatre de la tarda. Això és perquè estem calculant la resta de dividir 16 entre 12. Acabem de realitzar l'operació 1*6 mod 12 = 4*, que llegim com a *16 mòdul 12*. L'operació *mòdul* ens retorna el residu d'aquesta divisió.

La potència d'aquesta calculadora es va posar de manifest quan Gauss va comprovar que podia saber sense esforços el residu de divisions de xifres com 7^{99} entre qualsevol nombre sense haver de calcular aquests nombres tan grans primer.[12]

2.5.2 Congruències

La noció de congruència va aparèixer abans de Gauss amb Euler. Legendre i Lagrange, però Gauss introduí la notació moderna en la primera secció de *Disquisitiones Arithmeticae.*

Diem que *a* és congruent amb *b* mòdul p , $a \equiv b\ (mod\ p)$, si i només si p divideix a (*a - b)* , que és el mateix que dir que $a\ (mod\ p) = b\ (mod\ p)$.

El residu de la divisió de *a* entre p i de *b* entre p és el mateix, per tant la diferència és divisible entre p, donant com a residu 0. Vegem un exemple:

$71 \equiv 15\ (mod.\ 8) \rightarrow \quad 71 \div 8 = 8,875 \quad i \quad 15 \div 8 = 1,875$ Per tant, $(71 - 15) \equiv 0\ (mod.\ 8)$ *ja que* $56\ (mod.\ 8) = 0$

[12] Veure Annex IV

2.5.3 Petit teorema de Fermat

Pierre de Fermat (1607) va comprovar que:

$$a^{p-1} \equiv 1 \,(mod\, p)$$

on p és un nombre primer i a és un nombre enter qualsevol no múltiple de p.
Aquesta propietat és extremadament útil quan entra en joc l'aritmètica modular.

2.5.4 Funció φ d'Euler i Teorema d'Euler

Prem atenció ara al conjunt format pel conjunt d'elements invertibles de $\mathbb{Z}_n$, format pels elements $x \in \mathbb{Z}$ tals que m.c.d(x, n)=1, és a dir, tots els elements més petits i coprimers amb n, de tal manera que el seu màxim comú divisor és 1.
Aquesta quantitat és definida per la *funció phi d'Euler,* $\varphi(N)$, amb $n \geq 2$ combinant les següents propietats:

- Si p és un nombre primer, $\varphi(p) = p - 1$.
- Si p és un nombre primer i $k \geq 2$, $\varphi(p^k) = p^{k-1}(p - 1)$.
- Si a i b compleixen que mcd(a,b) = 1, $\varphi(a \cdot b) = \varphi(b) \cdot \varphi(b)$.

Exemple: si tenim un nombre $N = p \cdot q$, on p i q són nombres primers, aleshores la quantitat de nombres més petits que N tals que el seu màxim comú divisor amb N sigui 1 és $\varphi(N) = (p - 1)\cdot(q - 1)$. Si tenim que $N = 15 = 3\cdot5$, aleshores $\varphi(N) = 2\cdot4 = 8$
Ho podem comprovar amb el llistat de nombres de l'1 al 15. Els nombres en negreta són els coprimers amb 15, que efectivament són 8 (l'1 no compta perquè $n \geq 2$):

{1, **2**, 3, **4**, 5, **6**, **7**, **8**, 9, 10, **11**, 12, **13**, **14**, 15}

D´altra banda, a conseqüència d'altres propietats que no dona l'abast esmentar, es pot demostrar que:
Teorema. [d'Euler]. Donat un a qualsevol en $\mathbb{Z}_n$, es compleix que

$$a^{\varphi(N)} \equiv 1 \,(mod\, n)$$

3. FONAMENTS DE TEORIA DE COMPLEXITAT

En criptografia, és essencial quantificar d'alguna manera *l'esforç computacional* que requereix portar a terme certa tasca. Quan augmentem la seguretat d'un disseny, cal justificar per què l'adversari no serà capaç de penetrar les garanties de seguretat que perseguim. La teoria de la complexitat és l'encarregada de proporcionar el llenguatge adequat per aquesta justificació, donant una classificació útil per categoritzar els problemes computacionals en funció dels recursos necessaris per resoldre'ls (en la teoria)[13]. Aquests recursos són principalment temps i espai (o memòria física).

3.1 Tipus de complexitat

Les classes de complexitat han d'entendre's com conjunts de problemes contingut dels quals estem lluny de conèixer per complet. Un primer enfocament són els problemes de decisió, els que la seva solució forma part del conjunt binari, per exemple, decidir si un nombre enter és primer o no. Com hi ha una enorme quantitat de problemes computacionals que poden formular-se de manera equivalent com problemes de decisió, si ens restringim només a aquest tipus de problema, no serà una limitació molt gran.
Acceptarem abans però aquestes premisses:

- La dificultat d'un problema es mesura en funció dels recursos que consumeixin els algorismes que calculen una solució.
- La dificultat es mesura en un escenari pessimista (*worst-case*), és a dir, la mesura de la complexitat és el cost computacional del valor d'entrada més desfavorable.
- Si existeixen algorismes eficients per resoldre un problema, aquest és considerat "fàcil". En cas contrari, considerarem que el problema és "difícil".

D'altra banda, què considerem que és un algorisme eficaç? Per contestar-ho, establirem com a unitat de mesura el nombre d'operacions elementals que realitza l'algorisme. Aquest nombre dependrà de la mida de l'entrada que rebi l'algorisme. Així, la complexitat d'un algorisme $\mathcal{A}$ es representarà a través d'una funció $f_{\mathcal{A}} : \mathbb{N} \to \mathbb{R}^{+}$, que ha d'interpretar-se formalment així:

[13] González Vasco & Pérez del Pozo. *Criptografía esencial. Principios básicos para el diseño de esquemas y protocolos seguros.* 2021

$f_A(n)$:= nombre d'operacions elementals que realitza l'algorisme $\mathcal{A}$ amb una entrada de tamany *n.* Ho podem exemplificar amb l'algorisme $\mathcal{A}_1$ que determina si un nombre és o no primer[14]:

Pseudocodi [Complexitat de *PRIME*]	// Aquest algorisme realitza contínuament la reducció modular n (mod. x) i després suma una unitat a x fins a trobar per a quin valor de x el residu és 0, i, per tant, haurem trobat un dels factors de n. //Com hem vist, el factor més petit d'un nombre és sempre menor que la seva arrel quadrada. Per tant, aquest algorisme farà com a molt, en el pitjor dels casos, $2\sqrt{n}$ operacions.

Input: n // nombre enter ≥ 2
Desenvolupament (sedàs no optimitzat)
x:= 2
While (n(mod x)$\neq$ 0) and ($x \leq \sqrt{n}$) do
x:= x + 1
EndWhile
If n(mod x) = 0 output: FALSE
Else output TRUE	

Si suposem que n es codifica en binari, com és habitual, aleshores $\mathcal{A}_1$ rep una cadena de $k = log_2 n$ i realitza $2\sqrt{2^k} = 2^{\frac{k}{2}+1}$ operacions, per tant, es tractaria d'una funció exponencial.

3.1.1 P=NP?

El problema de si P és igual a NP és un dels set problemes del mil·lenni, encara per resoldre. Consisteix a determinar si els problemes de classe P poden estar inclosos dins dels NP-complets o no (poc a poc anirem definint aquests conceptes). Comprovar-ho suposaria un gran avenç tant en la teoria de la complexitat computacional com en la informàtica i les matemàtiques.

- Anomenem P al conjunt de problemes als quals podem trobar una solució en un temps raonable.
- Anomenen NP al conjunt de problemes en els podem determinar, en un temps raonable, si tenen o no solució.

Està clar que els problemes P estan dins dels NP. El que no està clar és que tots els problemes NP estiguin dins dels P. Que puguem comprovar fàcilment que hi ha una solució no implica que puguem trobar-la fàcilment.

Un exemple seria el següent: ens donen una sèrie de nombres positius i negatius i ens demanen que trobem un conjunt d'ells suma dels quals sigui 0. Si ens donen la resposta, comprovar-la és molt senzill perquè només els haurem de sumar.

[14] Veure Annex VII per una implementació complementària

Però trobar aquest conjunt, en canvi, no ho és gens. La manera en què s'intenta resoldre aquesta qüestió és trobant els problemes NP més difícils (els NP-complets i NP-hard), i intentar trobar-ne la solució. La Figura 10 mostra amb conjunts les dues hipòtesis possibles.

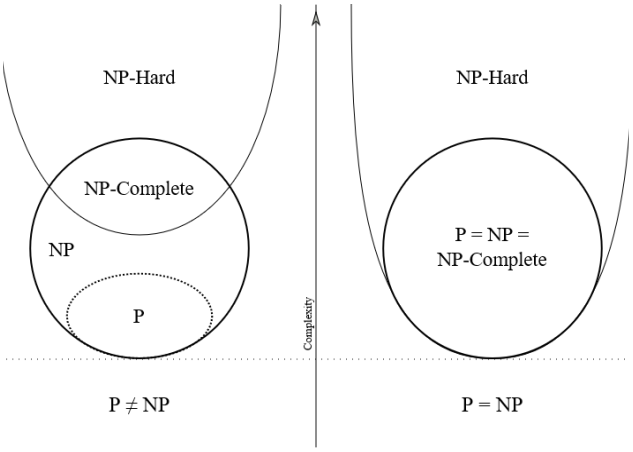


Figura 10. P vs NP. Font: https://stackoverflow.com/

3.2 Màquines de Turing

Al principi del segle XX, els matemàtics estaven fascinats per saber quantes matemàtiques podem fer utilitzant algorismes molt precisos, i no en base a la intuïció. Alan Mathison Turing (1912) estarà sempre associat al desxiframent d'Enigma[15]. Tanmateix, la inspiració que va portar a Turing a dissenyar una màquina capaç de descodificar Enigma va venir de la seva etapa com a estudiant a Cambridge, on va desenvolupar la teoria de les màquines de Turing per tal de demostrar el caràcter irresoluble del *problema de la parada o Halting Problem* [16], en relació amb les qüestions de lògica formal del segle. En el desenvolupament de la informàtica tal i com la coneixem actualment, va destacar John von Neumann, creador de l'arquitectura que utilitzen tots els ordinadors avui dia, la qual precisament va ser la implementació de la idea de la màquina universal de Turing, el nom que ocupa la base d'aquesta recerca. Tot el que un ordinador pot fer ara, ho pot fer aquesta màquina en la teoria. Turing va haver d'elaborar una noció de qualsevol algorisme possible, i aconseguir dissenyar una màquina genèrica que capturés com funcionarien.

15 Veure apartat III, punt 3

16 Frank Vega. *On P versus NP and Infinite Turing machines.* 2017

La manera en què la descriu és una cinta infinita, tan llarga com es necessiti, dividida en caselles que poden contenir un 1, un 0 o res. A més, hi ha un capçal (que podem imaginar com la CPU[17]) que llegeix una casella cada cop i segueix una sèrie d'instruccions (definició d'un estat, modificació de la casella i sentit de desplaçament). Una hipòtesi tan simple pot en realitat arribar a fer tots els càlculs que puguem imaginar. Amb aquesta proposició va ser capaç de respondre a les preguntes de Hilbert, a partir de la hipòtesi de la capacitat de determinar si un algorisme tindrà o no solució finita (és a dir, si la màquina pararà o no en algun punt amb la resposta codificada a la cinta).

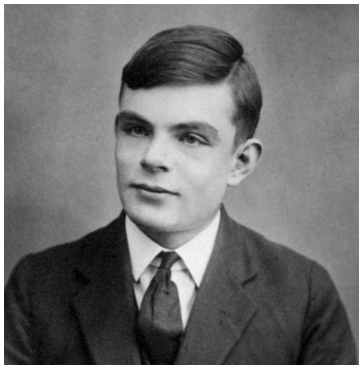


Figura 11. Turing el 1928

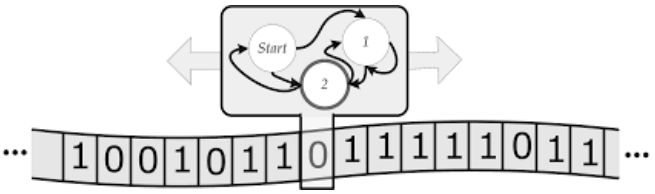


Figura 12. Esquema d'una màquina de Turing.

3.3 Notació asimptòtica o *Big(O)*

Hem ressaltat la importància de classificar els problemes avaluant l'eficiència dels algorismes que els resolen, i això només ho podem fer a través de funcions associades que mesuren com creix el cost computacional en relació amb el creixement del problema. Per això, utilitzem el que anomenem notació asimptòtica, que utilitzen tres lletres especials (O, Θ, Ω). Aquesta notació ens permet parlar d'ordres de complexitat en què agrupem funcions que, en el seu límit a l'infinit, creixen més o menys amb la mateixa proporció, i així fer aproximacions per poder comparar funcions de manera menys farragosa.

Així, per exemple, direm que una funció $f_A : \mathbb{N} \rightarrow \mathbb{R}^+$ és:

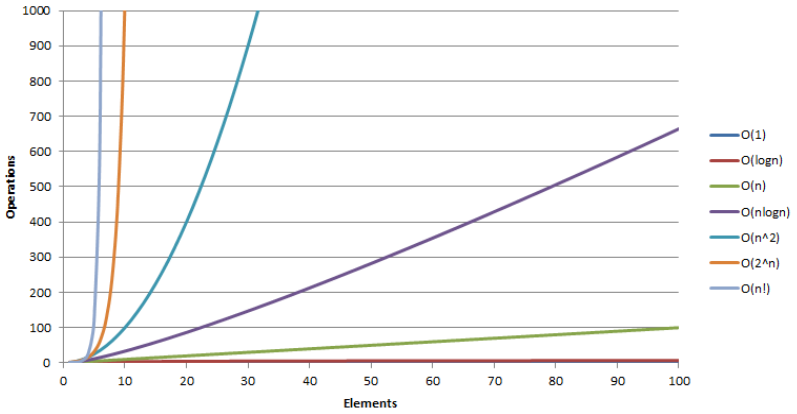
[17] *Central Processing Unit o unitat central de processament*

- Polinomial si existeix un $k \in \mathbb{N}$ tal que $f(n) = O(n^k)$. És a dir, la funció es comporta asimptòticament com un polinomi de grau menor o igual que k.
- Logarítmica si $f(n) = O(\log n)$
- Exponencial si existeix una constant positiva a tal que $f(n) = O(a^n)$

Quan dos funcions f, g: $\mathbb{N} \rightarrow \mathbb{R}^+$ creixents, podem assegurar que f creix més de pressa que g quan $\lim_{n \to \infty} \frac{g}{f} = \infty$. Aquest mètode ens permet comprovar l'ordre de la Taula 3, que expressem generalment com: $O(1) \leq O(\log n) \leq O(n \cdot \log n) \leq O(n^2) \leq O(n^k) \leq O(a^n)$, amb $2 \leq k$ i $1 < a$. També podem visualitzar una comparativa clara a la Gràfica 1.

Notació	**Nom (ordre)**
$O(1)$	Constant
$O(\log \log n)$	Sublogarítmica
$O(\log n)$	Logarítmica
$O\sqrt{n}$	Sublineal
$O(n)$	Lineal (primer ordre)
$O(n \cdot \log n)$	Lineal logarítmica
$O(n^2)$	Quadràtica (segon ordre)
$O(n^3)...O(n^c)$	Polinòmica (a partir del tercer ordre)
$O(c^n)$, n > 1	Potencial fixa
$O(n!)$	Factorial
$O(n^n)$	Exponencial

Taula 3. Notacions i ordres de funcions de complexitats.



Gràfica 1. Complexitat Big O

3.4 Sobre el temps de computació i la longitud de claus

Considerem, en criptografia, que un algorisme ens serà summament segur si no hi ha cap mètode analític per trencar-lo i el criptoanalista ha d'acudir a la *força bruta*.
Potser has sentit alguna vegada que es trigarien milers d'anys en trencar els algorismes que ens protegeixen avui dia a la xarxa. Com és possible?

Considerem, per exemple, un de 256 bits. Això vol dir que la clau utilitzada és de l'ordre de 2^{256}, si recordem les propietats del sistema binari, un nombre immensament gran. El nombre de bits que calculem són el nombre de bits de seguretat, que normalment es redueixen a la pràctica i no és el mateix nombre que s'estipula, ja que és possible que sistema de xifratge tingui alguna característica que redueixi el nombre de claus possibles (per exemple que no faci el nombre suficient de permutacions o substitucions). Això és el que coneixem com un trencament acadèmic, en què per exemple reduiríem el nombre 2^{256} a 2^{250}, que de totes maneres segueix sent inassequible.

En un cas ideal, la seguretat es basaria enterament en aquesta clau. Per demostrar aquesta impossibilitat d'abastament, prenem per exemple una CPU convencional de 3Ghz. Aquesta mesura és la freqüència del processador i ens indica que per cada segon, la CPU realitza 3 mil milions (o 3 bilions) d'instruccions per segon.

$\frac{2^{256}\ claus\ a\ provar}{3\times10^{9} instruccions\ per\ segon} \simeq 3{,}86 \times 10^{67}$ segons = $1{,}63 \times 10^{65}$ anys, que podem estar segurs que és molt més que l'edat de l'Univers. Encara que tinguéssim milions d'ordinadors treballant alhora, seguiria sent summament improbable trobar la clau correcta.

III. ANÀLISI DE LA MÀQUINA ENIGMA

1. REPÀS HISTÒRIC

El 1918, l'inventor alemany Arthur Scherbius va fundar la seva companyia Scherbius & Ritter, una firma d'enginyeria innovativa que entre els seus projectes estava el fet d'intentar substituir els sistemes criptogràfics inadequats utilitzats durant la Primera Guerra Mundial, de *llapis i paper*, per formes de xifratge que revolucionessin la tecnologia del segle XX. Així, va desenvolupar una peça de maquinària criptogràfica anomenada Enigma, basada en la versió elèctrica del xifrat de disc d'Alberti. La invenció de Scherbius es convertiria llavors en un dels sistemes més temibles d'encriptació en tota la història. Tot i que al principi no va tenir molta difusió, va anar millorant fins guanyar rellevància en molt poc temps fins al final convèncer el govern alemany d'utilitzar-la per a la transmissió d'informació secreta, convençut que era un sistema altament segur i pràcticament indesxifrable. En tot cas sabien que aquesta possibilitat existia, però com a mínim descoratjava l'espionatge casual. Tan el desenvolupament com el desxiframent d'aquesta màquina van canviar per complet el rumb de la Segona Guerra Mundial i és per això que el treball que l'envolta va ser tan significatiu.

Figura 13 . *Enigma D. Font: https://www.cryptomuseum.com/*

2. VISIÓ GENERAL DE L'ALEATORITZADOR

2.1 Especificacions tècniques

La màquina Enigma consistia en una sèrie de components electromecànics combinats de manera intricada. Tot i això, si descomponem la màquina en les seves parts, els seus principis són molt senzills. La forma bàsica de la invenció de Scherbius consistia en 3 elements connectats amb cables: un teclat semblant al de les màquines d'escriure, on s'introduïa el missatge original, una unitat aleatoritzadora (formada per rotors) que encriptava cada lletra del missatge original en una corresponent lletra en el text xifrat, i per últim un tauler de visualització consistent en diverses llums que il·luminaven la lletra a la qual s'havia xifrat l'original. Més tard, però, s'introduí un tauler de connexions que encara acabava de filar més prim. Aquest emparellava lletres de l'abecedari, de manera que la lletra xifrada pels rotors tornava a canviar de nou per la parella assignada al tauler, característica que augmentava considerablement el nombre de claus possibles. En última instància aparegué el reflector, que feia possible la simetria d'aquestes operacions combinades: un text xifrat que s'introduís amb les mateixes configuracions inicials, donava lloc al text original. Aquesta última versió, coneguda com a Enigma D, va ser la que els aliats britànics van haver de desxifrar. La Figura 14 mostra visualment les parts principals de la màquina.

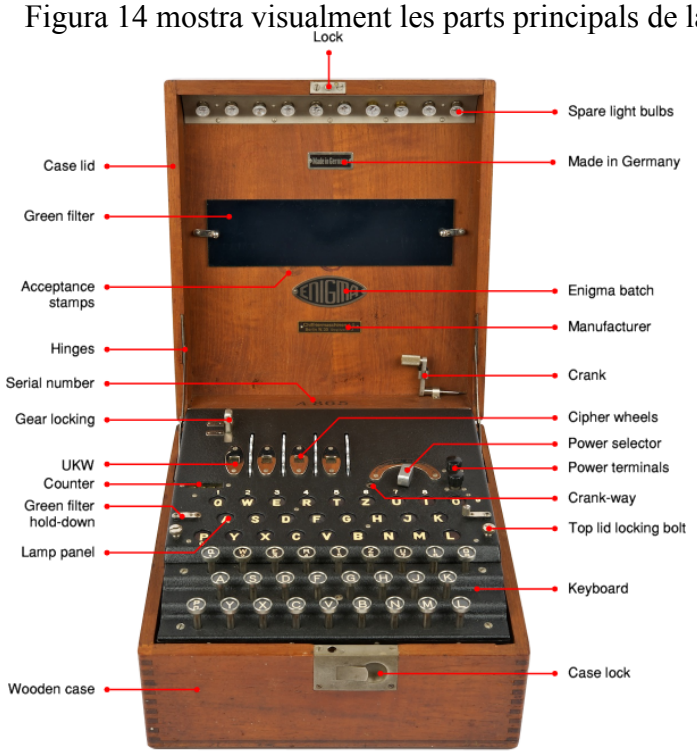


Figura 14. Parts de la màquina enigma

2.2 Funcionament

En aquest apartat veurem detalladament les parts que constitueixen la màquina per saber a quin problema es van enfrontar els criptoanalistes.

1. ROTORS

Els rotors o rodes de xifratge són els elements més importants de la màquina. Cadascun consisteix en una carcassa giratòria amb els números de l'1 al 26 (corresponent a les lletres de la *A* fins la *Z*. El centre del rotor està fixat a la roda selectora que conté tots els rotors de la màquina, entre els quals s'estableix una connexió elèctrica que uneix els contactes dels rotors en sèrie. En principi, la màquina introduïa 3 rotors, marcats amb nombres romans (I, II, III), però se'n van afegir fins a 8 per augmentar la seguretat. Cada rotor connectava una lletra d'entrada amb una de sortida mitjançant un cablejat interior. La Figura 15 mostra l'interior de cada rotor.

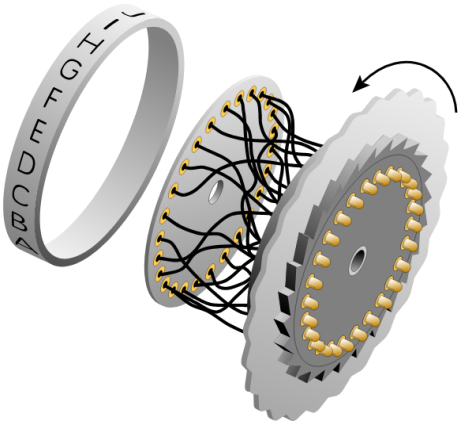


Figura 15. Interior del rotor

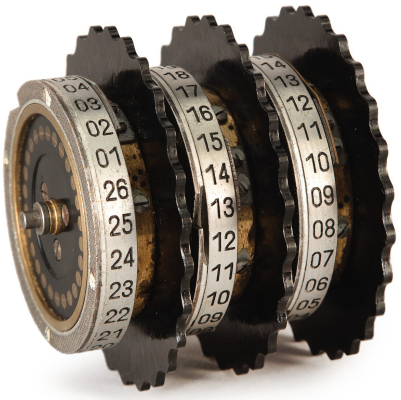


Figura 16. Tres rotors col·locats conjuntament

Els rotors de la màquina enigma tenen 3 variacions diferents a considerar a l'hora de configurar-los:

1. Es poden escollir 3 rotors (o 4 en l'Enigma M4) de 8 diferents, o els que hi hagués segons l'any i l'exèrcit que utilitzava la màquina. Cada tipus de rotor, com és d'esperar, està cablejat en creus de manera única i diferent.

2. Aquests rotors es poden disposar en ordres diversos. Obtindríem resultats diferents si poséssim els rotors II, V i IV en aquest ordre que si col·loquéssim aquests mateixos a l'inrevés.

3. Un cop col·locats, es pot escollir la posició inicial de cadascun d'ells, a través d'una finestra que permet veure-la, fent que les connexions inicials canviïn.

A més a més, cada cop que es prem una lletra, el primer rotor es desplaça una posició, de manera que el xifratge de la següent lletra té una configuració inicial diferent. Quan el primer rotor dona la volta sencera, és el segon el que es mou una posició. Podem imaginar-ho fent una analogia amb el comptador de quilòmetres d'un cotxe antic. Quan les dècimes arriben a 9, la roda de les unitats canvia a la xifra següent, i quan aquesta arriba a 9, la de les desenes es mou, i així successivament. Aquesta característica és un *plus* que garanteix que el procés de xifratge no sigui monòton, i permet que una lletra s'encripti en lletres diferents cada cop que es prem.

La Taula # mostra les transformacions de cadascun dels rotors dissenyats. Cal tenir present que hi podien haver tants rotors possibles com el factorial de 26, ja que aquest és el nombre d'ordenaments possibles de 26 elements.

I	II	III	IV	V	VI	VII	VIII
E - A	A - A	B - A	E - A	V - A	J - A	N - A	F - A
K - B	J - B	D - B	S - B	Z - B	P - B	Z - B	K - B
M - C	D - C	F - C	O - C	B - C	G - C	J - C	Q - C
F - D	K - D	H - D	V - D	R - D	V - D	H - D	H - D
L - E	S - E	J - E	P - E	G - E	O - E	G - E	T - E
G - F	I - F	L - F	Z - F	I - F	U - F	R - F	L - F
D - G	R - G	C - G	J - G	T - G	M - G	C - G	X - G
Q - H	U - H	P - H	A - H	Y - H	F - H	X - H	O - H
V - I	X - I	R - I	Y - I	U - I	Y - I	M - I	C - I
Z - J	B - J	T - J	Q - J	P - J	Q - J	Y - J	B - J
N - K	L - K	X - K	U - K	S - K	B - K	S - K	J - K
T - L	H - L	V - L	I - L	D - L	E - L	W - L	S - L
O - M	W - M	Z - M	R - M	N - M	N - M	B - M	P - M
W - N	T - N	N - N	H - N	H - N	H - N	O - N	D - N
Y - O	M - O	Y - O	X - O	L - O	Z - O	U - O	Z - O
H - P	C - P	E - P	L - P	X - P	R - P	F - P	R - P
X - Q	Q - Q	I - Q	N - Q	A - Q	D - Q	A - Q	A - Q
U - R	G - R	W - R	F - R	W - R	K - R	I - R	M - R
S - S	Z - S	G - S	T - S	M - S	A - S	V - S	E - S
P - T	N - T	A - T	G - T	J - T	S - T	L - T	W - T
A - U	P - U	K - U	K - U	Q - U	X - U	P - U	N - U
I - V	Y - V	M - V	D - V	O - V	L - V	E - V	I - V
B - W	F - W	U - W	C - W	F - W	I - W	K - W	U - W
R - X	V - X	S - X	M - X	E - X	C - X	Q - X	Y - X
C - Y	O - Y	Q - Y	W - Y	C - Y	T - Y	D - Y	G - Y
J - Z	E - Z	O - Z	B - Z	K - Z	W - Z	T - Z	V - Z

Taula 4. Transformacions alfabètiques de cada rotor preestablert durant la Segona Guerra Mundial.

2. TAULER DE CONNEXIONS (*PLUGBOARD)*

El tauler de connexions augmenta considerablement les possibles condicions inicials. El van utilitzar les forces aèries per primer cop per complicar-ho encara més.

Consisteix en un panell amb endolls que connecten una lletra amb una altra, tant a l'inici com al final del procés de xifratge. Simplement enllacen dues lletres amb cables en parelles, i es poden encreuar tantes com 13 (normalment s'emparellaven 10).

Figura 17. Tauler de connexions d'Enigma

3. REFLECTOR

L'últim component a destacar és el reflector. És una peça que envia els impulsos elèctrics que arriben des de l'últim rotor de volta al circuit en ordre invers. Cada pin del reflector connecta les lletres en 13 parelles, de manera que tornen per un camí diferent de l'inicial. L'avantatge d'aquest disseny és que el xifrat i el desxifrat es realitzen amb el mateix procés electromecànic i els mateixos ajustaments, però, d'altra banda, provoca una deficiència en el codi, perquè és llavors físicament impossible que una lletra d'entrada es xifri per ella mateixa si realitza un camí de tornada diferent. En el següent apartat veurem com això va ser una via d'entrada gegant en la criptoanàlisi del sistema.

Un cop vists els components, com podem expressar matemàticament el procés de xifratge d'aquest sistema electromecànic?

Si repassem l'ordre de les operacions, tenim que:

1. S'introdueix una lletra d'entrada amb el teclat (*keyboard*).
2. Aquesta lletra d'entrada passa pel tauler de connexions (*plugboard*) i es xifra en una altra. Anomenarem aquest procés *P*
3. Aquesta lletra passa ara a la unitat aleatoritzadora composta per *n* rotors. El conjunt d'aquestes *n* transformacions l'anomenarem *M*
4. Per últim, aquesta segona lletra xifrada arriba al reflector, *R,* per ser enviada pel camí del procés contrari. Cada procés contrari l'identificarem com la inversa de cada lletra o X^{-1}.
5. La lletra resultant es veurà il·luminada al panell de llums (*lightboard* o *lampboard*).

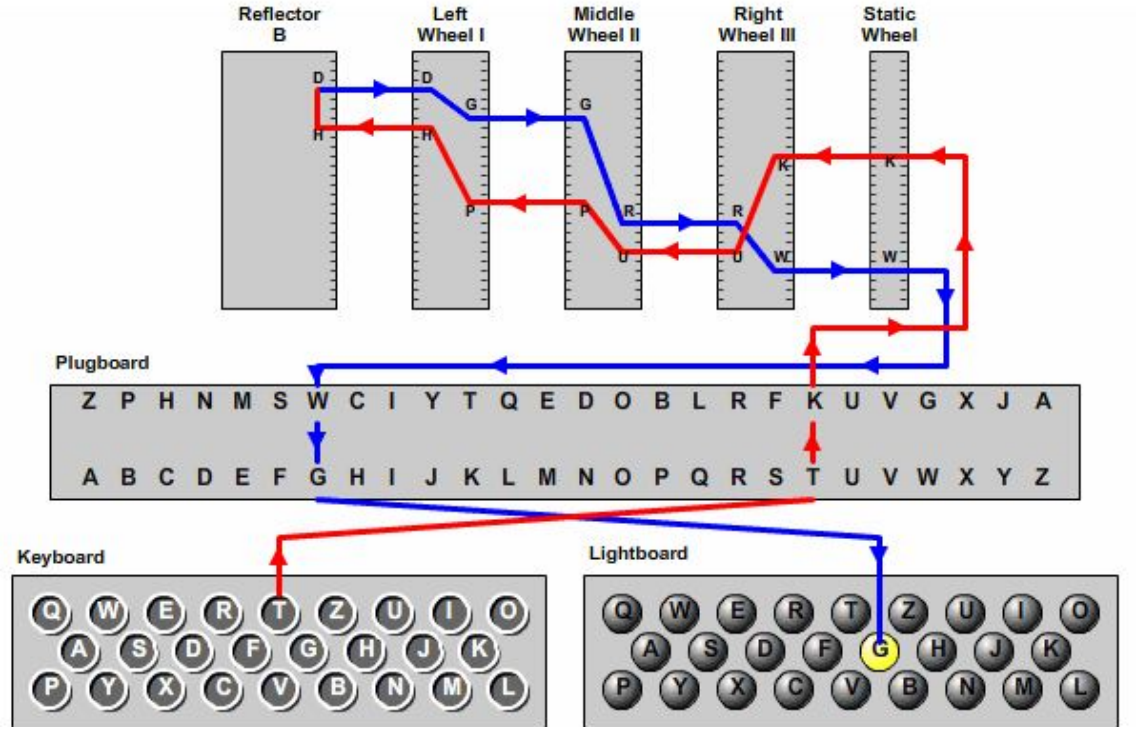


Figura 18. Esquematització del xifratge electromecànic d'Enigma. Font: Louise Dade, 2006.

El procés conjunt de cada capa de xifratge el podem escriure com:

$$K \rightarrow P \rightarrow M \rightarrow R \rightarrow M^{-1} \rightarrow P^{-1} \rightarrow L$$

3. CRIPTOANÀLISI DE L'ENIGMA

El buró de xifratge Polonès ja va fer avenços en els mètodes per desxifrar Enigma abans que l'oficina britànica (situada a la instal·lació militar de Bletchley Park), però va ser aquesta, amb Turing i Welchman al capdavant, la que va aconseguir desxifrar el codi de l'exèrcit naval i preparar-se per a qualsevol canvi. El problema al qual s'enfrontaven era el següent: havien de descodificar els missatges transmesos per ràdio diàriament pels alemanys, els quals utilitzaven una configuració inicial diferent cada dia. En menys de 24 hores havien de trobar la clau que els permetia recuperar el text original d'entre un nombre enorme. El primer que es va fer, evidentment, va ser calcular aquest nombre de claus possibles en funció dels seus elements:

→ En primer lloc, l'Enigma naval utilitzava 3 rotors a escollir-ne entre 5. Això fa un total de $5 \times 4 \times 3 = \mathbf{60}$ possibilitats.

→ En segon lloc, aquests rotors tenien una posició inicial a partir de la qual es feien les connexions entre sortides i entrades de cada rotor. Com cada un té 26 lletres, aquest nombre és $26 \times 26 \times 26 =$ **17576**.

→ Per últim havien de determinar les combinacions del tauler de connexions. El nombre de maneres diferents en què es poden formar n parelles de lletres en un alfabet de 26 lletres en funció de n és $f(x) = \frac{26!}{(26-2x)!\ x!\ 2^x}$

A partir de les propietats dels nombres combinatoris, sabem hem de calcular 26! i dividir entre les lletres restants no emparellades, el factorial del nombre de parelles (perquè no importa l'ordre d'aquestes) i entre dos elevat al nombre de parelles perquè considerem la parella (D, G) la mateixa que (G, D), per tant aquest ordre tampoc el considerem.

n	f(x)	n	f(x)	n	f(x)
1	325	6	100391791500	10	150738274937250
2	44850	7	130509328950	11	205552193096250
3	3453450	8	10767019638375	12	102776096548125
4	164038875	9	53835098191875	13	7905853580625
5	5019589575				

Taula 5. Nombre de combinacions del panell de connexions en funció del nombre de parelles

Tot i que el nombre de parelles òptim seria 11, els alemanys utilitzaven habitualment 10. Això equival a **150,738,274,937,250** combinacions possibles. El nombre total és, doncs, $60 \times 17576 \times 150738274937250 = \mathbf{158,962,555,217,826,360,000}$, que és aproximadament 159 trilions. Si fem la conversió a bits, hem de calcular $log_2 158962555217826360000 \simeq 67$ bits. Això vol dir que la clau és de 2^{67} dígits, que és bastant assequible per computadors actuals.

3.1 El defecte d'Enigma

És evident que no podien provar totes aquestes claus en un sol dia, per tant, havien de recórrer a mètodes cripotanalítics que reduïssin considerablement aquest nombre. Van aprofitar un dels errors més significatius de la màquina: que una lletra mai podia ser ella mateixa. A més a més, molts dels malencerts en els procediments alemanys van facilitar que es poguessin dur a terme estudis probabilístics i d'anàlisi de freqüència de les lletres, però va ser la màquina dissenyada per Turing la que va filar més prim.

3.2 Treball a Bletchley Park; *Bombe*

Fins i tot amb la producció en sèrie de la màquina Enigma, Berlín havia de manar agents que fessin arribar als capitans dels submarins i d'altres divisions els llibres amb la descripció detallada de la posada a punt de les màquines per a codificar els missatges diaris d'un mes sencer[18]. Si aconseguien aquest llibre, el problema estava resolt, cosa gairebé infreqüent (al·ludim aquí al problema de distribució de claus dels sistemes simètrics: que destinatari i remitent s'hagin de posar d'acord abans pot facilitar l'acció del criptoanalista si no es va amb cura). Però com es podia construir un mecanisme que permetés desxifrar el codi produït per Enigma sense saber les configuracions inicials?
Turing va trobar la manera d'utilitzar l'error esmentat per trencar els codis en un temps raonable. Així que va dissenyar junt amb Welchman la màquina *Bombe,* que trobava la clau en menys de 20 minuts. *Bombe* funcionava per mitjà de descartar les claus que no podien ser perquè conduïa a contradiccions lògiques, treballant en la possible resolució del panell de connexions.

[18] Veure Annex I

Per fer-ho, abans agafaven un text d'exemple i provaven de col·locar alguna frase o paraula que sabien segur que podia estar al missatge. Per exemple la paraula *wetterbericht*, que volia dir "informe del temps" i s'usava en totes les transmissions.
Aleshores lliscaven aquest text en el criptograma de manera que cap lletra coincidís amb el supòsit, i, a partir d'aquí, era possible fer deduccions de les possibles connexions del panell. Aquí és on entra en joc *Bombe:* com era factible, donada una lletra d'entrada i una de sortida, hipotetitzar parelles del taulell, el que feia aquesta gran màquina era anar provant diverses possibilitats fins que s'arribava a una contradicció lògica. Donem per cas que havíem suposat que la *N* estava connectada amb la *G*. Si després de seguir fent càlculs a partir d'aquesta proposició, s'arribava al fet que la *G* estava connectada amb la *U,* llavors això no era possible i es descartaven totes les operacions fetes. [19]
Això es feia amb desenes de simulacions d'Enigma funcionant alhora, que amb impulsos elèctrics podien acomplir les operacions anteriors instantàniament. Cada tercet de rotors de la màquina (semblants a un rellotge) mostrat a la Figura 19, n'eren una. Es provaven totes les posicions possibles dels rotors, gairebé una per segon en cada simulació, i cada canvi era el descart de centenars de combinacions. Al final, quan la màquina parava, el que mostrava era l'opció que no donava cap errada lògica i, per tant, era la més probable, i només calia posar-la a prova.

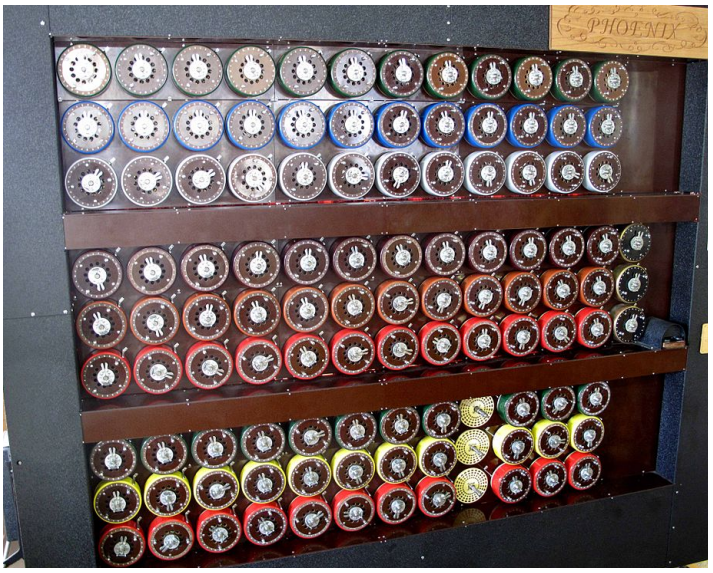


Figura 19. Bombe machine. Font: cryptomuseum.com

[19] Explicació al canal de *Numberphile - Flaw in the Enigma code.* https://www.youtube.com/watch?v=V4V2bpZlqx8

4. PRÀCTICA I: CÀLCUL DE LA SEGURETAT DEL SISTEMA ENIGMA OPTIMITZAT.

Utilitzar un ordinador per xifrar un missatge és, en gran manera, molt semblant a les formes tradicionals de xifratge. De fet, només hi ha tres diferències significatives entre el xifratge informàtic i el tipus de xifratge mecànic que va ser la base de xifratge com l'Enigma[20].

La primera diferència és que una màquina de xifratge mecànica està limitada pel que es pot construir pràcticament, mentre que un ordinador pot imitar una màquina de xifrat hipotètica d'una complexitat immensa. Per exemple, es podria programar un ordinador per imitar l'acció d'un centenar de codificadors. Aquesta màquina mecànica seria pràcticament impossible de construir, però el seu equivalent informatitzat "virtual" oferiria un xifrat altament segur. La segona diferència és simplement una qüestió de velocitat. L'electrònica pot funcionar molt més ràpidament que els codificadors mecànics: un ordinador programat per imitar el xifrat Enigma podria xifrar un missatge llarg en un instant. Alternativament, un ordinador programat per dur a terme una forma de xifratge molt més complexa encara podria executar la tasca en un temps raonable. La tercera diferència, i potser la més significativa, és que un ordinador codifica els números en lloc de les lletres de l'alfabet.

Els ordinadors només tracten nombres binaris: seqüències d'uns i zeros conegudes com a dígits binaris, o bits per abreviar. Per tant, abans del xifratge, s'ha de convertir qualsevol missatge en dígits binaris. Aquesta conversió es pot realitzar d'acord amb diversos protocols, com l'*American Standard Code for Information Interchange*, conegut familiarment per l'acrònim ASCII. L'ASCII assigna un nombre binari de 7 dígits a cada lletra de l'alfabet de moment, n'hi ha prou amb pensar en un nombre binari com un patró d'uns i zeros que identifica de manera única una lletra (taula #). Hi ha 2^7 (128) maneres d'ordenar una combinació de 7 dígits binaris (cal tenir en compte que el sistema binari és de base 2)[21].

---

20 Singh, S. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography.* 1999

21 Per més detall del sistema binari, visitar https://www.bbc.co.uk/bitesize/guides/zwsbwmn/revision/1

Això dona molt de lloc per definir les lletres majúscules i minúscules, així com els símbols de puntuació, entre d'altres.

La intenció en aquest apartat és fer el supòsit que es vol emprar una versió d'aquest sistema per xifrar la nostra informació utilitzant els recursos computacionals actuals, dissenyant una "màquina Enigma ideal". Hem d'imaginar que volguéssim fer servir aquest sistema simètric per encriptar missatges o arxius a la xarxa, de ela mateixa manera que es faria amb l'AES o l'RSA.

Per tant, s'ha de fer una interpretació completament virtual. En el cas hipotètic, donat que tinguéssim els recursos computacionals i el codi adequats, els propòsits són els següents.

1. La màquina ideal es basa en un alfabet de 128 dígits d'acord amb el sistema ASCII.
2. Hi ha tants rotors com ordenacions possibles d'aquest codi de 128 dígits, és a dir, 128!
3. S'agafen a l'atzar 14 rotors d'aquests 128! (versemblança amb el sistema AES-256, que té 14 capes de xifratge), operació que només podem donar cabuda amb una funció aleatoritzadora digitalment (*random choice)*.
4. Al tauler de connexions s'enllaçaran el nombre òptim de parelles que doni un nombre més gran de combinacions possibles. Aquest és el càlcul que es farà amb *Sage* de *CoCalc.*
5. L'element del reflector se suprimeix per tal que una lletra sí que pugui codificar-se per ella mateixa. El procés de desxifratge, per tant, només és possible digitalment però no mecànicament.

4.1 Càlcul de la llargada de la clau

Hem de trobar la manera que aquests patrons dels quals s'ha parlat a l'apartat 3 no es repeteixin, és a dir, que com a molt les repeticions es produeixin amb una distància *n* més llarga que el mateix text a xifrar, de manera que per molt llarga que sigui la cadena de caràcters, sigui molt difícil trobar cap patró. En els xifratges monoalfabètics, és molt fàcil determinar els patrons basant-se en l'anàlisi de freqüències, quan cada lletra és

substituïda per un únic element. Per això és convenient aplicar les regles de Kerckhoffs i les bases del secret perfecte de Shannon de manera que el text xifrat no pugui donar cap pista sobre l'original.
El que pretenc aconseguir és una clau que, expressada en bits, sigui un nombre summament gran. Per a un criptoanalista, el nombre total de missatges possibles a partir d'un text xifrat ha de ser completament inassequible.

El mètode emprat serà més aviat heurístic perquè vull determinar una possible resolució al problema, però sense poder garantir al 100% que sigui totalment òptima.

El càlcul de la llargada de la clau es farà basant-se en els propòsits esmentats:

1- El nombre de variacions per escollir 14 rotors de 128! és definit per $V_{n,r}=\frac{n!}{(n-r)!}$. Com aquest nombre no és calculable amb els recursos que tinc disponibles, simplement reduiré el nombre a 1000. Tenim com a resultat $\frac{1000!}{(1000-14)!}\simeq 9.12\times 10^{41}$

2- La quantitat de posicions inicials dels 14 rotors és $128^{14} = 3.17\times 10^{29}$

3- Per calcular el nombre òptim de parelles hem de trobar el màxim de la funció que dona el nombre de combinacions en funció del nombre de parelles. De manera semblant al càlcul anterior amb enigma, $f(x) = \frac{128!}{(128-2x)!\times x!\times 2^x}$.
En el nostre cas només s'han de computar un nombre finit de valors discrets, ja que amb 128 lletres només podem formar com a màxim 64 parelles, per després trobar el major. És una qüestió resoluble en tan sols uns segons.

Per resoldre-la, la defineixo com es mostra a la primera línia de codi de la Figura 20, per a continuació donar les instruccions *retornar el màxim* i *dibuixar el gràfic*.
Tot i que no retorna un valor discret, sabem que es tracta de 59.
Veiem com el gràfic és semblant a una distribució gaussiana, com era de preveure.
A la Figura 21 faig també el càlcul de totes les solucions possibles.

El valor obtingut és aproximadament 1.33×10^{111}

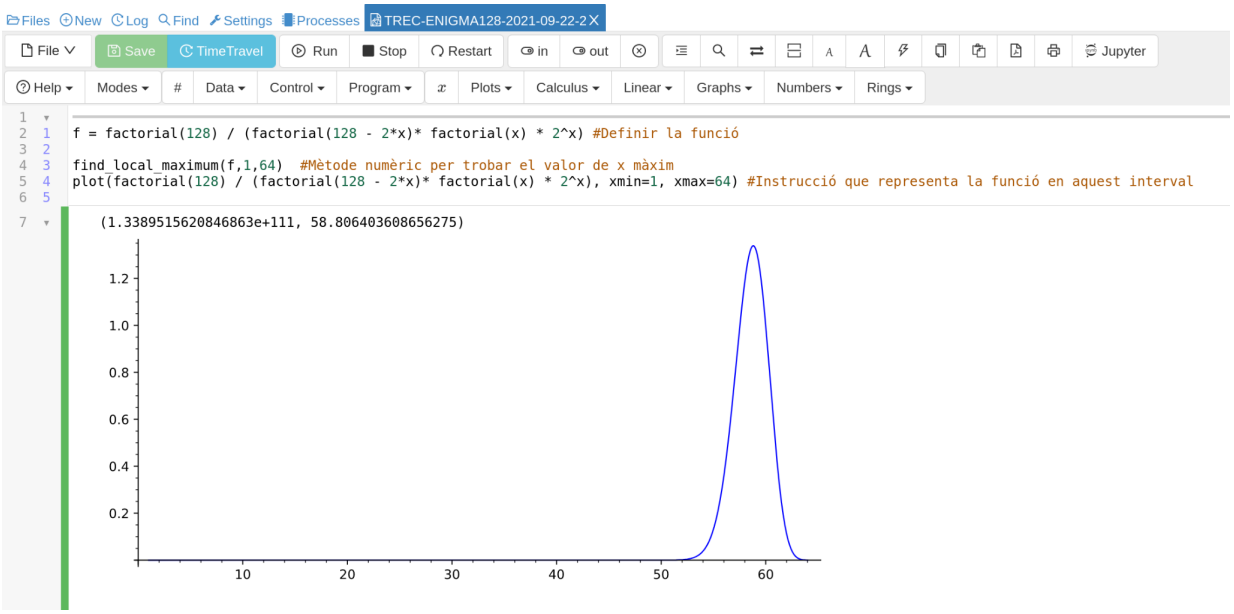


Figura 20. Càlcul amb Sage del valor màxim de la funció i la seva representació gràfica.

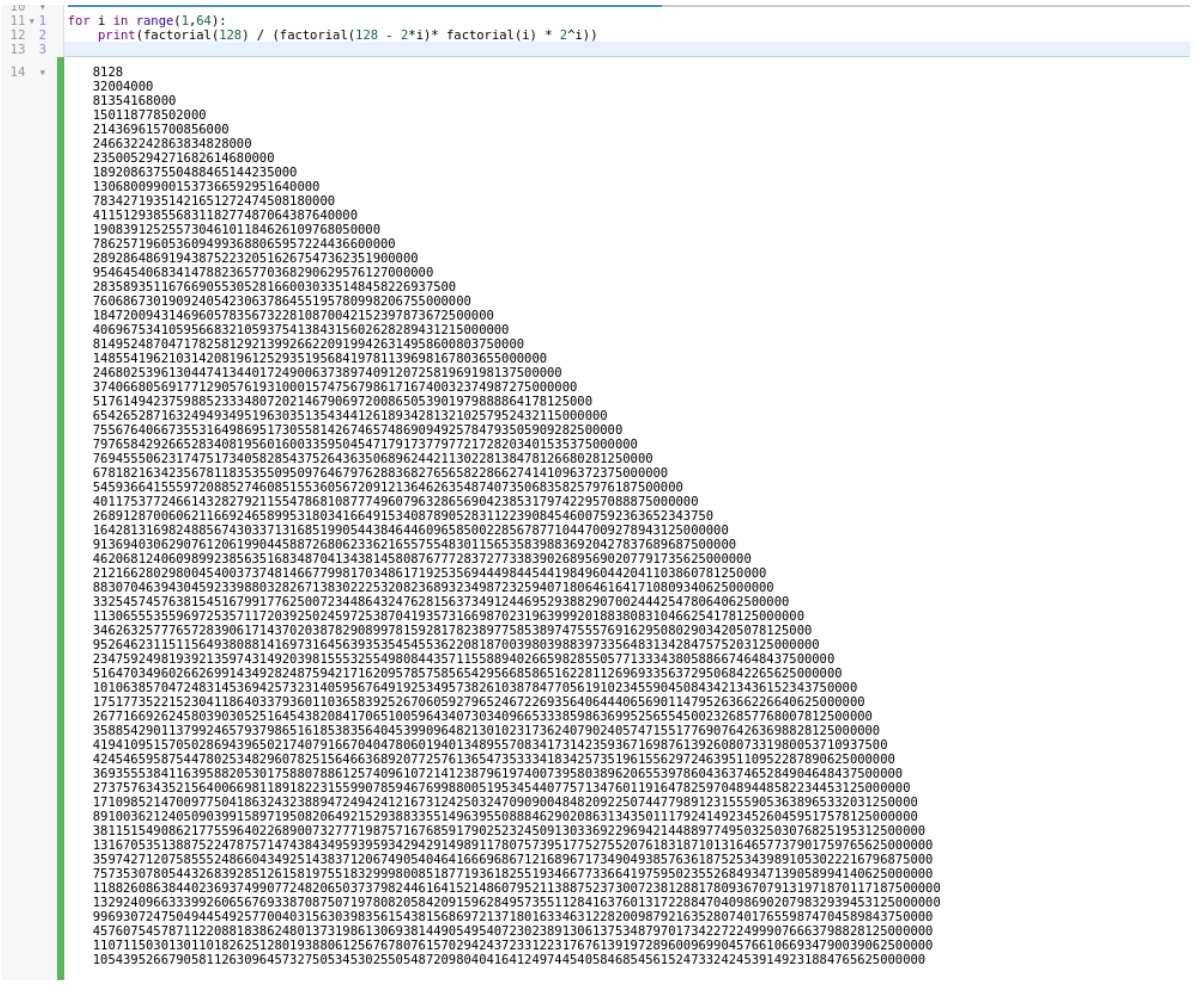


Figura 21. Combinacions possibles en funció del nombre de parelles.

4.2 Càlcul del temps de computació

El nombre total de claus serà el producte dels resultats obtinguts anteriorment en cada etapa. El nombre de claus final equival a

$$9.12 \times 10^{41} \times\ 3.17 \times 10^{29} \times\ 1.33 \times 10^{111} \simeq 3.84\ \times\ 10^{182}$$

El nombre de bits serà el logaritme en base 2 d'aquest nombre:

$$log_2 (3.84\ \times\ 10^{182})\ \simeq\ 606\ bits$$

Acabem de verificar que podem construir un sistema hipotètic amb un nombre de bits de seguretat major que els sistemes simètrics més emprats com l'AES. A més a més, el procés de xifratge és aproximadament lineal. Això és un avantatge que busquem: que el temps que es triga en xifrar no sigui massa gran ni suposi molt de cost computacional.
Com en teoria no existeix cap atac analític que pugui trencar-lo, calculem ara el temps que trigaria algú que volgués recórrer a la força bruta.
Agafem el processador més ràpid existent al mercat a dia d'avui (Novembre 2021), el *Ryzen ThreadRipper 3990X* d'AMD. Aquest processador realitza 2,356,230 MIPS[22] a 4.35 GHz, és a dir, més de dos milions de milions d'instruccions per segon. Si dividim el nombre obtingut anteriorment entre el nombre d'operacions per segon que és capaç de realitzar aquesta CPU, tenim que trigaria:

$$\frac{3.84 \times 10^{182}}{2.35623 \times 10^{12}} \simeq\ 1.63\ \times 10^{170} \text{segons} = 5.16 \times 10^{162}\ anys$$

Podem estar segurs que ningú que intercepti un missatge xifrat d'aquesta manera podrà trobar el missatge original. Ara bé, com no és possible, amb els coneixements de base, implementar aquest algorisme (el nivell supera el límit computacional i teòric que tinc disponible), és possible que aquest nombre de bits es redueixin lleugerament, perquè els algorismes de selecció aleatòria mai són del tot aleatoris, però en tot cas la seguretat segueix sent pràcticament la mateixa i el nombre de capes de xifratge garanteix ser més que suficient.

[22] Milions d'Instruccions Per Segon, no confondre amb l'arquitectura *MIPS. El càlcul és aproximat i està simplificat per tal de ser comprensible. No es tenen en compte algunes de les variables reals.*

IV. ANÀLISI DE L'RSA

1. EL XIFRATGE RSA

Als anys setanta, tres científics —Ron **R**ivest, Adi **S**hamir i Leonard **A**dleman, a qui devem el nom de l'algorisme— van transformar la investigació sobre els nombres primers en una aplicació comercial seriosa: explotant un descobriment de Pierre de Fermat al segle XVII, tots tres van idear una manera d'utilitzar els nombres primers per protegir les nostres targetes de crèdit mentre viatgen pels centres comercials electrònics del mercat global. La seva idea va ser publicada en un informe tècnic d'IBM[23] el 1977, assolint una gran difusió. La versió d'aquest primer article, així com la que s'esquematitzarà en aquest apartat, no és ben bé la utilitzada a la xarxa, però els principis són els mateixos. S'empra als motors de cerca (Google, Firefox…), a les VPN (xarxa privada virtual), a moltes aplicacions i a més, pel seu caràcter com a xifratge de clau pública, és possible firmar i autentificar digitalment (firma RSA).
El seu algorisme havia de complir les propietats d'un sistema de clau pública establides per Diffie i Hellman:

- Qualsevol usuari pot calcular les seves claus pública i privada en temps polinomial
- L'emissor pot xifrar el missatge amb la clau pública del receptor en temps polinomial.
- El receptor pot desxifrar el criptograma amb la seva clau privada en temps polinomial.
- El criptoanalista que intenti esbrinar la clau privada mitjançant la pública trobarà un problema intractable.
- El criptoanalista que intenti desxifrar un criptograma tenint la clau pública es trobarà amb un problema intractable.

[23] *International Business Machines Corporation*

2. FUNCIONAMENT DE L'RSA

La representació Alice - Bob - Eve és molt coneguda per fer una simulació del que passaria en aquest xifratge. Fixem-nos en la Figura 22. Alice utilitzarà la clau pública de Bob per enviar-li un missatge, d'ara endavant K_{pub} , la qual pot conèixer tothom. Només Bob, amb la seva clau privada, K_{priv} , podrà desxifrar aquest missatge. Eve, que ha aconseguit interceptar la transmissió, no ha de poder desxifrar-lo tot i tenir la K_{pub}.

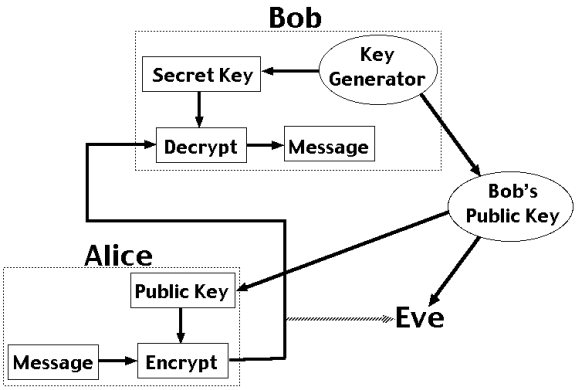


Figura 22. Transmissió d'informació en un canal insegur

2.1 Generació de les claus

L'algorisme de generació de les claus rep com a entrada un paràmetre de seguretat, un nombre natural *n* que fixarà les característiques en les quals es basa el disseny.

(1) Selecciona dos nombres primers, p i q, expressió dels quals en binari ocupi *n* bits (grandària similar a 2^n). Aquests dos nombres han de ser molt grans per garantir la seguretat. Aquí apareix una de les dificultats principals de l'RSA; la cerca de nombres primers adequats.

(2) Calcula $N = pq$ la funció d'Euler $\varphi(N) = (p-1)\cdot(q-1)$.

(3) Escull un exponent *e* que compleixi que *M.C.D(e,* $\varphi(N)$) = 1, és a dir, que sigui coprimer amb $\varphi(N)$, i tal que $1 \leq e \leq \varphi(N)$.

(4) Obté l'invers modular *d* de *e* mòdul $\varphi(N)$, és a dir, un enter *d* tal que $ed \equiv 1 (\text{mod } \varphi(N))$

(5) Dona com a sortida la $K_{pub} = (e, N)$ la $K_{priv} = (d, N)$

2.2 Xifrat de missatges

L'algorisme *xifrat* rep com a entrada la K_{pub} (e, N) i un missatge en clar m. Donarà com a sortida el text xifrat c construït a partir de $c = m^e \pmod N$.

Per xifrar un text és necessari codificar-lo prèviament en un sistema numèric en base **b** i dividir-lo en blocs de mida j-1 de manera que $\mathbf{b}^{j-1} < N < \mathbf{b}^j$

2.3 Desxifrat de missatges

L'algorisme *desxifrat* rebrà com a entrada c i la clau secreta d. Per retornar el text inicial m, calcularà $m = c^d \pmod N$.

2.4 Exemple

Per visualitzar-ho, faré una mostra amb paràmetres molt petits que siguin assequibles.

Un exemple molt fàcil és suposant que envio un missatge m consistent només una lletra, "X". En ASCII és 1011000, equivalent a 88 en decimal.

Així, m = 88.

Escullo els nombres primers p i q : 11 i 17. Aleshores $N = 11 \times 17 = 187$, i $\varphi(N) = 10 \times 26 = 160$.

Amb l'algorisme d'Euclides puc trobar $e = 7$, per exemple, i el seu invers modular $d = 23$ $(23 \cdot 7 = 161 \equiv 1 \pmod{160})$.

Per xifrar el missatge amb aquetsa K_{pub} (e, N), calcularé $88^7 \pmod{187}$. Obtinc 11 com a resultat. Per tant $c = 11$.

Si vull desxifrar aquest missatge, calcularé $11^{23} \pmod{187}$, que efectivament dona com a resultat 88.

```
88^7%187
    11

11^23%187
    88
```

El càlcul del mòdul l'he fet amb Sage, ja que amb la calculadora no és possible. Normalment s'empra el símbol % per indicar l'operació mod.

3. SEGURETAT DEMOSTRABLE

Sabem que aquest sistema funciona, però com podem tenir la certesa que és segur i com s'expressa en bits?

L'RSA es basa en funcions d'una sola via basades en el problema de la factorització d'enters. Si ens posem en el punt de vista d'*Eve*, la persona que vol desxifrar el missatge, el problema que hauria de resoldre és, donats *e*, *c,* i *N,* trobar *m* tal que $m^e = c \pmod N$. La funció necessària per resoldre'l, f_{RSA} , és considerada d'una sola via quan els nombres són prou grans. Revertir l'operació de xifratge és calcular una preimatge del text xifrat, el qual és molt difícil.

Per confiar en què ho sigui, cal fer una reflexió sobre la generació de les claus de l'RSA: quan es calculen *p i q*, s'ha de verificar que són primers, no utilitzant la solució polinomial esmentada al problema *PRIME* sinó amb un test de primalitat[24] amb certa probabilitat d'error, perquè sinó la tasca computacional seria massa alta. Després, el càlcul de l'exponent *e* es fa amb els objectius de minimitzar el cost del càlcul del xifratge (s'ha de fer una exponenciació), però també de limitar l'abast d'atacs coneguts[25]. Per últim, l'invers modular *d* s'ha de calcular amb l'algorisme estès d'Euclides. L'RSA té una complexitat exponencial en aquest sentit.[26]

Sabem, d'altra banda, que si aconseguim factoritzar el mòdul N, podrem calcular $\varphi(N)$ i trobar la clau secreta *d* associada a qualsevol implementació de l'RSA. Per això existeixen molts mètodes amb sistemes d'equacions no lineals, o si ens trobem en el cas d'un entorn multiusuari que utilitza el mateix exponent *e*, també serà fàcil trobar *p* i *q*.

L'expert en seguretat Simson Garfinkel va estimar que una CPU Intel Pentium de 100Mhz trigaria aproximadament 50 anys en factoritzar un nombre primer de l'ordre de 10^{130}. Si expressem en bits aquest nombre amb $log_2\ (10^{130})$, sabem que són aproximadament 435 bits. Fàcilment comprovem que per tal que sigui segur l'RSA, *N* ha de ser molt gran.

---

[24] Veure Annex VII

[25] Veure Annex IX

[26] En la definició clàssica això és cert però cal mencionar que en els últims anys s'han trobat funcions superpolinòmiques però subexponencials que resolen f_{RSA}

Avui dia s'ha de garantir que *p* i *q* siguin cadascun com a mínim de 512 bits.
N sol ser de 1024 bits (uns 309 dígits decimals) i de 2048 bits (617 dígits decimals).

Cal fer esment dels rècords de factorització de l'RSA. Un dels primers va ser l'RSA129, en què es va aconseguir factoritzar el 1994 el nombre
114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958705058989075147599290026879543541

en *3490529510847650949147849619903898133417764638493387843990820577*
× 327691329932667095499619881908344614131776429679929425397982885 33

L'última versió més gran factoritzada va ser l'RSA768 l'any 2009. Les competicions segueixen obertes i es premia amb grans sumes monetàries els primers que aconsegueixin trencar l'RSA-1024 i l'RSA-2048.

3.1 Esment de la seguretat en un futur

En última instància és necessari mencionar que amb l'arribada dels computadors quàntics a la realitat, factoritzar nombres grans seria una feina abordable i en el moment en què es trenqués un d'aquests dos sistemes, tota la seguretat a la xarxa estaria en perill, i s'hauria de reformular per complet trobant algorismes resistents a atacs quàntics. S'estima que això serà en unes quantes desenes d'anys, però aquests nous sistemes ja comencen a dissenyar-se i a posar-se a prova.[27]
D'una manera semblant, demostrar que trencar qualsevol criptosistema és NP-complet seria un gran assoliment en criptografia, però P≠NP és probablement el supòsit més "estàndard". Per al cas de l'RSA, el problema de la factorització d'enters és poc probable que sigui NP-complet. Tanmateix, òbviament és a NP, però si la teoria s'aconseguís unificar, ja no seria segur.

[27] El llibre de Simon Singh *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography* fa una explicació molt acurada d'aquests sistemes basats en els principis de la quàntica

V. CONCLUSIONS

1. CONCLUSIONS GENERALS

En un treball tan condensat i complet com aquest, les conclusions a extreure podrien no acabar mai. Tot i que cada apartat és bastant resolutiu, tot s'enfoca capa un vessant que vull acabar de concloure en aquesta secció.

Després de moltes hores de programar, entendre algorismes, comparar funcions i esquematitzant possibles sistemes criptogràfics, s'ha aconseguit respondre a la hipòtesi de manera positiva:

Sí que és possible que un sistema amb els mateixos principis que Enigma sigui igual o inclús més segur que sistemes de base RSA. S'ha aconseguit, a partir de la teoria, analitzar aquest nou sistema Enigma idealitzat, que podria perfectament ser usat per a l'encriptació de missatges a aplicacions com *Whatsapp* garantint la privacitat dels usuaris.

No ha sigut fàcil determinar-ho, però s'han pogut establir les següents característiques:

- És factible un disseny optimitzat d'Enigma enfocat a l'aplicació avui dia que opera de manera pseudoaleatòria i, per la senzillesa del procés de xifratge, el temps per dur-lo a terme no supera l'ordre polinòmic.
- En el cas hipotètic donat que totes les propietats de Shannon es compleixin, l'ordre del trencament de la clau és d'ordre factorial
- A diferència de l'RSA, l'encriptació té un cost computacional més baix i no es fa partint de l'exponenciació.

A més a més, aquesta anàlisi ens permet autenticar que els sistemes de clau pública basats en el producte de dos nombres primers requereix longituds de clau més grans, ja que de no ser així és possible trobar la clau secreta amb mètodes numèrics i els suficients recursos computacionals, a diferència dels de clau simètrica, que principalment són blocs de substitucions i permutacions i els mètodes analítics són menys freqüents.

2. CONCLUSIONS PERSONALS I CONCLUSIONS DELS OBJECTIUS

Aquesta recerca ha sigut probablement el repte intel·lectual més transcendent al que mai m'he enfrontat. Tan sols el fet d'haver trobat una hipòtesi que relacionés tots els conceptes pels quals tenia interès no ha estat un procés fàcil.

Tot i haver-hi un final i una data d'entrega, estic segura que seguiria investigant i trobant resposta a molts dels dubtes que han sorgit pel camí. És per això que m'agradaria continuar la investigació i acabar de determinar com actuaria realment aquest sistema *pseudorandom* heurístic com a proposta de tesi doctoral.

El que puc assegurar és que he gaudit moltíssim realitzant aquesta memòria, malgrat els moments de tensió, frustració i de creure que no ho aconseguiria. Els esforços han donat uns resultats originals, íntegres i autèntics amb un caràcter d'unicitat propis.

D'altra banda, cal dir que quan per primera vegada vaig llegir sobre aquest tòpic, no m'imaginava que el Treball de Recerca el faria relacionat amb el tema, ni molt menys que tindria la intenció d'enforcar-lo des del punt de vista abstracte i d'anàlisi matemàtica, perquè no coneixia la relació existent entre aquestes ciències.

Vull remarcar el meu agraïment per haver tingut l'oportunitat de reflectir en un treball el meu interès per les matemàtiques i la informàtica, i que m'ha permès aprendre més que amb qualsevol altre mètode. Tots els llibres que d'alguna manera o una altra em van fer veure que aquest era el tema a escollir es troben a la bibliografia. És una pena no poder incloure tot el que he aprehès, sobretot de la branca de la matemàtica, per falta d'espai i de temps, i per la simplicitat a la qual m'havia de limitar. He descobert com la demostració és l'eina més important del matemàtic, i l'elegància amb què s'empra el llenguatge matemàtic per descriure qualsevol fenomen, no per intuïció, sinó seguint un conjunt de regles molt específiques i completes. Tot això reafirma la meva passió per les matemàtiques i la meva intenció de cursar-les en els estudis universitaris.

En última instància concloc assegurant l'assoliment de tots objectius amb èxit, fins i tot de molts que no he esmentat al preàmbul.

No obstant això, m'he trobat amb dificultats sobretot per la necessitat de perfeccionament i síntesi de la informació; seleccionar el que s'ha de mantenir i el que no no és obvi i he necessitat un punt de vista extern per millorar-ho.

Faig al·lusió, per acabar, a tots els pensaments i reflexions a posteriori del treball amb els quals he pogut veure el pes i l'abast d'aquesta disciplina científica, que tot i semblar molt teòrica, en la vida quotidiana té un valor inimaginable.

V. BIBLIOGRAFIA

1. LLIBRES

[1] Caballero Gil, P. (2002). *Introducción a la Criptografía* (2a edición actualizada ed Ra-Ma.)

[2] du Sautoy, M. (2007). *La música de los números primos*. Acantilado.

[3] Enrique Gracián. (2011). *Los números primos*. RBA National Geographic.

[4] Gauss, C. F. (1990). *DISQUISITIONES ARITHMETICAE*. Universidad de Costa Rica. http://elaprendiz.es/disquisitionesarithmeticae_Gauss.pdf

[5] González Vasco, M. I., & Pérez del Pozo, Á. L. (2021). *Criptografía esencial. Principios básicos para el diseño de esquemas y protocolos seguros*. Ra-Ma.

[6] Grimaldi, R. P. (1998). *Matemáticas discreta y combinatoria. Una introducción con aplicaciones* (3a edición ed.). Pearson.

[7] Grime, J. (n.d.). *Maths from the talk "Alan Turing and the Enigma Machine"*. http://www.singingbanana.com/enigmaproject/maths.pdf

[8] Hardy, G.H. (1967). *Apología de un matemático*. Capitán Swing.

[9] Hodges, A. (1983). *Alan Turing: the Enigma*. Burnett Books.

[10] Hodges, A. (1994). *Alan Turing, Enigma* (10.1007/978-3-7091-9381-5 ed.). Computerkultur.

[11] Hofstadter, D. R. (1987). *Gödel, Escher, Bach. Un Eterno y Grácil Bucle*. Tusquets.

[12] Juher, D. (2004). *L'art de la comunicació secreta. El llenguatge de la criptografia*. Llibres de l'índex.

[13] Maurer, U. (2015). *Cryptography and Computation after Turing*. Cambridge University Press 2. https://crypto.ethz.ch/publications/files/Maurer16b.pdf

[14] Miller, D. A. R. (n.d.). *The Cryptographic Mathematics of Enigma*. Center for Cryptologic History National Security Agency. https://www.nsa.gov/portals/75/documents/about/cryptologic-heritage/historical-figures-publications/publications/wwii/CryptoMathEnigma_Miller.pdf

[15] Prasad, K., & Kumari, M. (2020). *A review on mathematical strength and analysis of Enigma*. University of Jharkhand, India. https://arxiv.org/pdf/2004.09982.pdf

[16] Schoof, R. (n.d.). *Four primality testing algorithms*. arxiv.org. https://arxiv.org/pdf/0801.3840.pdf

[17] Shannon, C. E. (1949). *Communication Theory of Secrecy Systems*. Bell System Technical. http://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf

[18] Singh, S. (1998). *Fermat's Enigma*. Anchor Books.

[19] Singh, S. (1999). *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor Books. 0.1145/966789.966797

[20] Talbot, J., & Welsh, D. (2006). *Complexity and Cryptography. An introduction*. Cambridge University Press.

[21] Vega, F. (2017). *On P versus NP and Infinite Turing machines*. Hal archives. https://hal.archives-ouvertes.fr/hal-01473565/document

[22] Wilcox, J. (2006). *Solving the Enigma: History of the Cryptanalytic Bombe*. Center for Cryptologic History National Security Agency. https://permanent.fdlp.gov/gpo52788/solving_enigma.pdf

2. RECURSOS WEB

[1]http://wiki.franklinheath.co.uk/index.php/Enigma/Key_Length#Calculated_Effective_Key_Length (Agost 2021)

[2]http://www.alanturing.net/turing_archive/archive/index/codebreakingindex.html#Enigma (Juny 2021)

[3] https://simonsingh.net/cryptography/ (Maig 2021)

[4] https://www.cienciaoberta.cat/criptografia/ (Agost 2021)

[5] https://www.datio.com/security/rsa-how-maths-will-protect-us-while-pnp/ (Setembre 2021)

[6] https://www.uv.es/~sto/cursos/seguridad.java/html/sjava-8.html (Setembre 2021)

[7] https://www.youtube.com/watch?v=YQw124CtvO0 (Juliol 2021)

[8] https://www.101computing.net/enigma-machine-emulator/ (Abril 2021)

[9] https://www.lri.fr/~fmartignon/documenti/systemesecurite/6-PublicKey.pdf (Octubre 2021)

[10] https://keisan.casio.com/calculator (Novembre 2021)