COM DE FIABLES SÓN ELS ALGORISMES MÉS UTILITZATS EN LA CRIPTOGRAFIA MODERNA?

LLUÍS SABATÉ PONS

TREBALL FINAL DE GRAU
D'ENGINYERIA INFORMÀTICA
06/06/2021

Professor col·laborador: Miguel Martín Mateo

Professor : Jordi Serra Ruiz

Índex

1. Introducció 4
1.1. Motivació 4
1.2. Abstract 4
1.3. Objectiu 5
1.4. Descripció de les fases del projecte 6
1.5. Planificació Temporal 6
1.6. Replanificació Temporal 8
2. Recorregut històric de la criptografia clàssica 10
2.1. Orígens 10
2.2. Algorismes clàssics per transposició 10
2.3. Algorisme clàssic per substitució 12
3. Criptoanàlisi 15
3.1. Que és? 15
3.2. Alan Turing - Enigma 15
4. Criptografia moderna 18
4.1. Que es un criptosistema? 18
4.2. Orígens 18
4.3. Criptografia simètrica 20
4.3.1. En bloc 20
4.3.2. De flux 21
4.4. Criptografia asimètrica 22
5. Algorismes més utilitzats 24
5.1. Algorisme simètric en blocs: AES (Advanced Encryption Standard) 24
5.2. Algorisme simètric de flux: CHACHA20 33
5.3. Algorisme asimètric: RSA 37

6. Aplicacions criptogràfiques 40
6.1. Protocols de comunicació segura 40
6.1.1. Protocol SSL (Secure Socket Layer) 40
6.1.2. Protocol TLS (Transport Layer Security) 41
6.1.3. Protocol IPSec 41
6.2. Autenticació 42
6.2.1. Certificat digital 42
6.2.2. Signatura digital 43
7. Part practica 45
7.1. Packet Tracer 45
7.2. OPENSSL 53
8. Conclusions 58
9. Annex 60
9.1. Operació Mod 60
9.2. Operació XOR 60
9.3. Funció de Resum o hash 60
10. Referències 61

1. Introducció

1.1. Motivació

La informació cada vegada més, s'està convertint en un dels actius amb més valor per a les empreses i la més gran font de poder que ha conegut l'home. Avui dia té més poder qui més informació controla, ja que la informació es coneixement.

La informació que cal protegir es privilegiada, valuosa o sensible per a diferenciar-la de la informació que no esta sotmesa a condicions que restringeixen l'explotació que se'n vol fer. Cal protegir doncs,la informació privilegiada amb mesures de seguretat necessàries per garantir que la seva utilització es manté dins dels marges correctes de la legitimitat.

Segons la normativa ISO/IEC 27000 la seguretat de la informació es basa en els conceptes de:

- Confidencialitat, implica que només els autoritzats poden accedir a aquesta informació.
- Integritat, implica que la informació no ha sigut alterada, esborrada o destruïda per autoritats no autoritzades a aquesta, preservant així la seva exactitud original.
- Disponibilitat, implica que la informació sigui accessible pels autoritzats quan ho requereixin.
- Autenticitat, implica que la informació és fidedigna a l'origen real.
- No repudi, implica que es preserva que alguna de les parts, emissor o receptor, negui algun compromís o acció pres amb anterioritat.

La criptografia utilitza xifratge de clau publica o privada, funcions de resum o firmes digitals entre altres per garantir quasi tots els conceptes parlats anteriorment, menys la disponibilitat.

Per tant, la criptografia es una eina essencial per el control de la informació i té com a objectiu mantenir actius els conceptes de la ISO/IEC 27000. A més, es l'única tecnologia que pot garantir-nos nivells raonables d'anonimat. Permetre que els ciutadans utilitzin tècniques criptogràfiques per protegir la seva intimitat, limita de manera efectiva la informació.

En conclusió, la principal motivació d'aquest projecte, ve donat per la problemàtica que genera la informació avui dia, i per tant tenir com a objectiu prioritari aconseguir la seguretat d'aquesta mitjançant la criptografia.

1.2. Abstract

This end-of-degree work consists of performing in a first stage, a historical journey of classical cryptography including the Enigma machina, indicating what it represented and its cryptanalysis.

A second stage, consisting of the study of modern cryptography, including the current type of cryptography, its applications, and the reason for the most frequently used algorithms.

Finally a third stage, where it will be practical, using cryptographic tools such as OpenSSL and Packet Tracer.

These three stages will be developed by 7 tasks with the aim of analysing how reliable the algorithms most are used in modern cryptography.

****

Aquest treball de final de grau consisteix en realitzar en una primera etapa, un recorregut històric de la criptografia clàssica incloent la maquina Enigma, indicant el que va representar i el seu criptoanàlisi.

Una segona etapa, que consisteix en l'estudi de la criptografia moderna, incloent el tipus de criptografia actual, les seves aplicacions i el perquè dels algorismes més freqüents que s'utilitzen.

Finalment una tercera etapa, on serà de caràcter pràctic, utilitzant eines criptogràfiques com OpenSSL i Packet Tracer.

Aquestes 3 etapes estaran desenvolupades per 7 tasques amb l'objectiu d' analitzar com son de fiables els algorismes més utilitzats en la criptografia moderna.

1.3. Objectiu

La criptografia es la ciència que estudia les tècniques matemàtiques relacionades amb els diferents aspectes de la seguretat de la informació. L'objectiu principal d'aquest projecte es analitzar com son de fiables els algorismes més utilitzats en la criptografia moderna. Per poder arribar a l'objectiu plantejat anteriorment, s'han de definir un conjunt de tasques o fases:

1. Planificació del projecte.
2. Recorregut històric de la Criptografia
3. Criptoanàlisi Alan Turing – Enigma.
4. Criptografia Moderna: Criptosistemes.
5. Quins algoritmes s'utilitzen i perquè?
6. Aplicacions criptogràfiques.
7. OpenSSL i Packet Tracer.
8. Redactar documentació i realitzar presentació del projecte.

1.4. Descripció de les fases del projecte

A continuació es descriu amb un poc més de detall les fases d'aquest projecte:

- Planificació del projecte. Aquesta primera fase consisteix en identificar el context i justificar l' importància de realitzar aquest projecte. També hi ha la documentació prèvia per conèixer a grans trets la temàtica del projecte. A més es defineix l'objectiu del treball, les tasques necessàries per arribar a l'objectiu marcat i una planificació temporal de les tasques a realitzar.

- Recorregut històric de la Criptografia. Aquesta segona fase consistirà bàsicament en explicar l'origen de la criptografia i una descripció de la criptografia clàssica.

- Criptoanàlisi: Alan Turing – Enigma. Aquesta tercera fase consistirà en un resum de Enigma i com aquesta maquina criptogràfica va fer que l'alemanya nazi comences a perdre la segona guerra mundial.

- Criptografia Moderna: Criptosistemes. En aquesta fase s'introdueixen els conceptes de la Criptografia moderna amb els seus diferents criptosistemes.

- Quins algoritmes s'utilitzen i perquè?. En aquesta fase es parlarà dels algorismes més utilitzats en els criptosistemes avui dia.

- Aplicacions criptogràfiques. En aquesta fase s'explicarà la utilització que es dona en la criptografia moderna avui en dia, amb autentificació, certificats i firmes digitals a més de protocols de comunicació segura.

- OpenSSL i Packet Tracer. Aquesta penúltima fase serà la part practica del projecte on es faran ús de tasques criptogràfiques d'exemple.

- Redactar documentació i realitzar presentació del projecte. En aquesta ultima fase s'exposaran les conclusions a les quals s'han arribat i es realitzarà la presentació del projecte.

1.5. Planificació Temporal

Per explicar la planificació que s'ha dut a terme per el bon desenvolupament del projecte, es presenta una estimació del temps, mitjançant un diagrama de Gantt.

El diagrama de Gantt es una eina que serveix per la planificació de projectes. Proporciona una vista general de les fases o tasques programades i ajuda a saber quines fases es tenen que realitzar i en quina data.

Aquest projecte s'ha definit en 8 fases:

1. Planificació del projecte.
2. Recorregut històric de la Criptografia.
3. Criptoanàlisi Alan Turing – Enigma.
4. Criptografia Moderna: Criptosistemes.
5. Quins algoritmes s'utilitzen i perquè?
6. Aplicacions criptogràfiques.
7. OpenSSL i Packet Tracer.
8. Redactar documentació i realitzar presentació del projecte.

El treball fi de grau comença el 17 de febrer de 2021 i finalitza el 6 de juny de 2021. A continuació podem observar les fases i les seves dates:

		Nombre de tarea	Comienzo	Fin	Predecesoras
1		Planificació del projecte	mié 17/02/21	mar 09/03/21	
2		Recorregut històric de la Criptografia	mar 02/03/21	jue 11/03/21	
3		Criptoanalisi Alan Turing - Enigma	vie 12/03/21	dom 21/03/21	2
4		Criptografia Moderna: Criptosistemes	lun 22/03/21	mié 14/04/21	3
5		Quins algoritmes s'utilitzen i pequè?	jue 15/04/21	mié 05/05/21	4
6		Aplicacions Criptogràfiques	mié 12/05/21	mar 25/05/21	
7		OpenSSL i Packet Tracer	jue 15/04/21	jue 27/05/21	
8		Redactar documentació i realitzar presentació del projecte	vie 28/05/21	dom 06/06/21	7

Figura 1.1: Diagrama de Grantt

En la següent imatge podem observar la línia temporal del projecte:

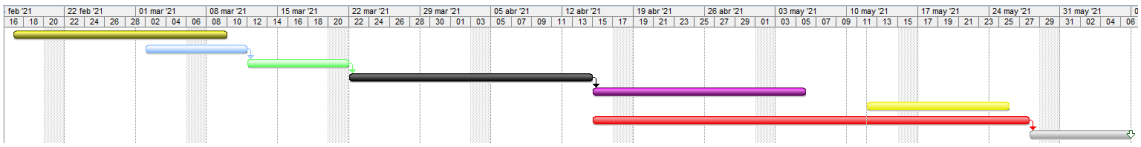


Figura 1.2: Diagrama de Grantt

A la següent taula podem veure una estimació de temps, per les fases que s'han de dur a terme durant la realització del projecte:

Iteració	Activitat	Hores
1	Planificació del projecte	30
2	Recorregut històric de la Criptografia	20
3	Criptoanàlisi Alan Turing – Enigma	20
4	Criptografia Moderna: Criptosistemes	64
5	Quins algorismes s'utilitzen i perquè?	64
6	Aplicacions criptogràfiques	35
7	OpenSSL i Packet Tracer	35
8	Redactar documentació i realitzar presentació del projecte	20

Figura 1.3: Temporització del projecte

Totes aquestes fases fan un total de 288 hores per el desenvolupament del treball fi de grau, treballant una mitjana de 3 hores al dia de dilluns a dissabte i repartides en 16 setmanes.

1.6. Replanificació Temporal

A la següent taula podem veure una replanificació de temps, per les fases que s'han dut a terme:

Iteració	Activitat	Hores invertides	Hores estimades	Diferencia
1	Planificació del projecte	30	30	0
2	Recorregut històric de la Criptografia	40	20	+20
3	Criptoanàlisi Alan Turing – Enigma	15	20	-5
4	Criptografia Moderna: Criptosistemes	50	64	-14
5	Quins algorismes s'utilitzen i perquè?	68	64	+4
6	Aplicacions criptogràfiques	20	35	-15
7	OpenSSL i Packet Tracer	45	35	+10

8	Redactar documentació i realitzar presentació	20	20	0

Figura 1.4: Replanificació temporal del projecte

Com es pot observar en la taula anterior, hi hagut 3 tasques que no han requerit tantes hores com les que s'havien estimat en un principi, ja que s'havia fet una previsió errònia. Per altre banda, les activitats de Recorregut històric de la Criptografia i Quins algorismes s'utilitzen i perquè? ha requerit més hores de les previstes, degut a la gran quantitat d'informació disponible i la seva complexitat. També ha requerit més hores de les previstes la part practica amb l' implementació d'aplicacions criptogràfiques amb els programes Packet Tracer i OpenSSL. En termes globals, la planificació ha complert les 288 hores marcades a l'inicií del projecte.

2. Recorregut històric de la criptografia clàssica

2.1. Orígens

La paraula criptografia prové de la unió dels termes grecs Krypto que significa secret i Grapho que significa escriptura. La criptografia clàssica agrupa totes les tècniques d'escriptura secreta, des del segle V a.C. fins mitjans del segle XX. Aquestes tècniques depenen del tipus d'operacions lineals que es realitzi al text en clar durant el xifratge, utilitzen mètodes de transposició de caràcters per aconseguir difusió o mètodes de substitució de caràcters per aconseguir confusió. La transposició consisteix en permutar la informació de forma distinta al missatge original, es a dir, caviar caràcters de lloc que ocupen en el text en clar. La tècnica per substitució consisteix en canviar certs caràcters del text en clar per un altre caràcter o caràcters de l'alfabet.

A continuació podem observar la classificació de la criptografia clàssica:

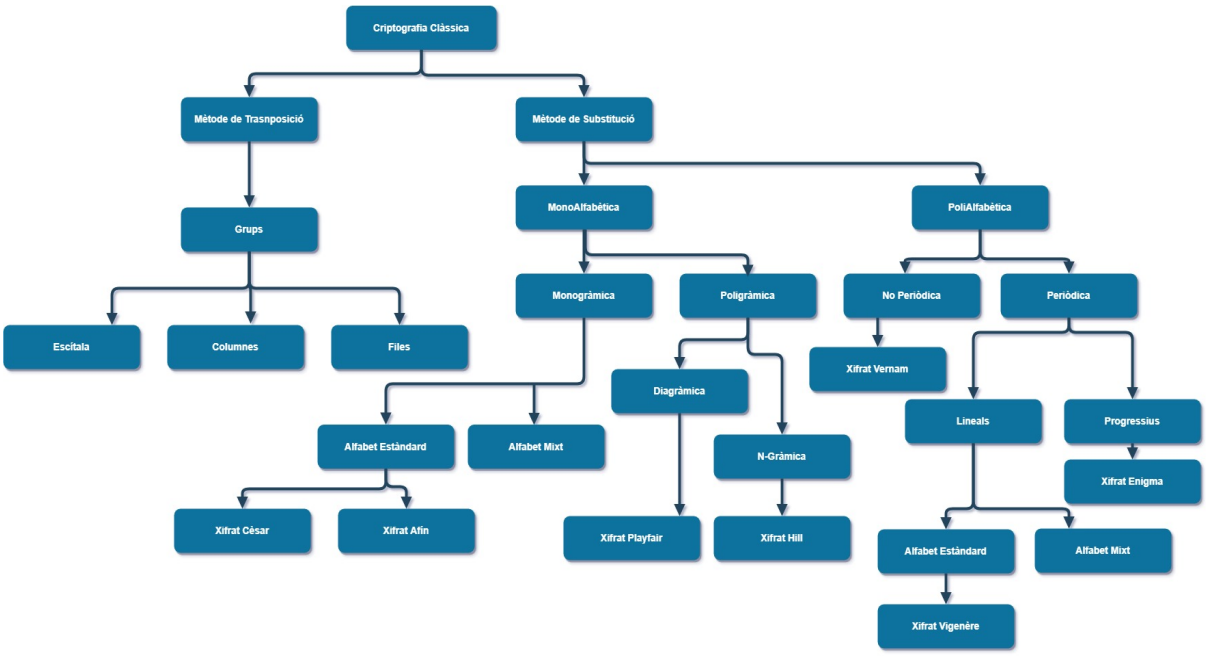


Figura 2.1: Classificació criptografia clàssica

2.2. Algorismes clàssics per transposició

El xifratge per transposició te un curt recorregut en la història de la criptografia clàssica. En aquest mètode trobem les següents tècniques:

- Escítala, aquest sistema de xifratge va ser utilitzat per els grecs al segle V a.C. Consistia en un bastó de fusta en el qual s'enrotllava una cinta de cuir i després s'escrivia en ella de forma longitudinal el missatge que es desitjava amagar. Fet això, en desenrotllar la cinta les lletres apareixeran desordenades. Per a desxifrar el criptograma i recuperar el

text en clar, calia tornar a enrotllar aquesta cinta en un bastó amb el mateix diàmetre usat la primera vegada. La clau d'aquest sistema era el diàmetre del bastó.

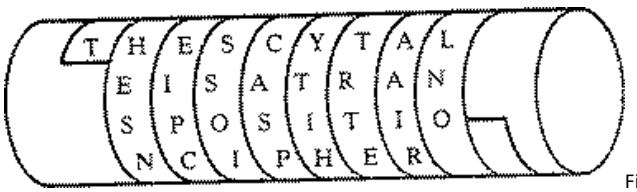


Figura 2.2: Escítala

- Xifratge per columnes, s'escriu el text en clar horitzontalment dins el rectangle, passant a la següent fila quan s'ha arribat al numero màxim de columnes, el text xifrat es l'ordenació del text en clar per columnes. La clau es el numero de columnes. En el cas que el text en clar deixes espais en blancs en el rectangle, ja que aquest rectangle fos més gran que el nombre total de lletres que conté el missatge, hauríem d'introduir la lletra X. Antigament era comú agrupar el criptograma en blocs de 5 lletres.

T	R	E	B
A	L	L	D
E	T	F	G

Figura 2.3: Xifratge per columnes

El text en clar es TREBALL DE TFG. El text xifrat es TAERL TELFB DG. La clau es 4.

- Xifratge per files, s'escriu el text en clar verticalment, passant a la següent columna quan s'ha arribat al numero màxim de files, el text xifrat es l'ordenació del text en clar per files. La clau es el numero de files. En el cas que el text en clar deixes espais en blancs en el rectangle, ja que aquest rectangle fos més gran que el nombre total de lletres que conté el missatge, hauríem d'introduir la lletra X.

T	B	L	T
R	A	D	F
E	L	E	G

Figura 2.4: Xifratge per files

El text en clar es TREBALL DE TFG. El text xifrat es TBLTR ADFEL EG. La clau es 3.

2.3. Algorisme clàssic per substitució

En el xifratge per substitució definiré únicament les tècniques més importants:

- Xifratge del Cèsar, aquest sistema de xifratge va ser utilitzada per Juli Cèsar al segle I a.C, l'algoritme consisteix en realitzar un desplaçament de 3 espais cap a la dreta dels caràcters del text en clar. A continuació, podem observar un quadre, on hi ha l'alfabet de 27 caràcters:

	0									10											20						
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6
m	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
x	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Figura 2.5: Alfabet de xifrat del Cèsar mod 27

En aquest xifratge la m significa el missatge en clar, la x el missatge xifrat o criptograma, la b el desplaçament i n es el numero de elements del alfabet. Tenim per tant, les següents expressions d'algoritmes per xifrar i desxifrar:

- x=m + b mod n
- m=x - b mod n

Un exemple de criptograma, es x= XVIFE OOHIX JK i el seu missatge en clar es m = TREBALL DE TFG

- Xifratge Afin, aquest sistema de xifratge es diferencia del xifratge del Cèsar en que en l'algoritme de xifratge es multiplica el text en clar per una variable, a, la resta es exactament igual. Tenim per tant, les següents expressions d'algoritmes per xifrar i desxifrar:

- x=a*m + b mod n
- m=x/a - b mod n

Un exemple d'aquest tipus de xifratge es el següent on a=2, b=3 i n=27:

Text en clar m	T	R	E	B	A	L	L	T	F	G
Alfabet	20	18	4	1	0	11	11	20	5	6
*2 mod 27	13	9	8	2	0	22	22	13	10	12
+3 mod 27	16	12	11	5	3	25	25	16	13	15

Text xifrat x (Criptograma)	P	M	L	F	D	Y	Y	P	N	O

Figura 2.6: Xifratge Afin

El criptograma es PMLFD YYPNO. El text en clar es TREBALL TFG.

- Xifrat de Vigenère, aquest sistema de xifratge consisteix en utilitzar més d'un alfabet de xifratge, es a dir, s'utilitza una clau per xifrar el missatge en clar. Tenim per tant, les següents expressions d'algoritmes per xifrar i desxifrar:

  - $x_i=m_i+k_i$ mod n, on m_i es el caràcter en una posició concreta en el text en clar, k_i es el caràcter en una posició concreta en la clau, x_i es el caràcter en una posició concreta xifrat i n es el numero de caràcters del alfabet.

  - $m_i=x_i-k_i$ mod n, on x_i es el caràcter en una posició concreta xifrat, k_i es el caràcter en una posició concreta en la clau, m_i es el caràcter en una posició concreta en el text en clar i n es el numero de caràcters del alfabet.

Un exemple d'aquest tipus de xifratge es el següent, on m=TREBALL TFG i k=XIFRAR:

Text en clar m	T	R	E	B	A	L	L	T	F	G
Alfabet	20	18	4	1	0	11	11	20	5	6
Clau k	X	I	F	R	A	R	X	I	F	R
Alfabet	24	8	5	18	0	18	24	8	5	18
m_i+k_i mod 27	17	26	9	19	0	2	8	1	10	24
Text xifrat x (Criptograma)	Q	Z	J	S	A	C	I	B	K	X

Figura 2.7: Xifratge de Vigenère

El criptograma es QZJSA CIBKX. El text en clar es TREBALL TFG. La clau es XIFRAR.

- Xifrat de Hill, aquest sistema de xifratge consisteix en utilitzar matrius. La matriu de la clau ha de ser quadrada, la seva dimensió ens indica el numero de caràcters que xifrem i el numero de caràcters que tindrà la clau. El determinant de la matriu de la clau no pot ser 0, ja que ha de tenir una matriu inversa per desprès poder desxifrar el missatge. Les expressions dels algoritmes per xifrar i desxifrar son les següents:

  - $x_j=k_{ij}*m_i$ mod n, on k_{ij} es el caràcter en una posició concreta en la matriu clau, m_i es el caràcter en una posició concreta en el text en clar,

x_j es el caràcter en una posició concreta xifrat i n es el numero de caràcters del alfabet.

- $m_i=x_j/k_{ij}$ mod n, on x_j es el caràcter en una posició concreta xifrat, on k_{ij} es el caràcter en una posició concreta en la matriu clau, m_i es el caràcter en una posició concreta en el text en clar i n es el numero de caràcters del alfabet.

Un exemple d'aquest tipus de xifratges es el següent:
El text en clar es TFG, per tant necessitem una matriu de dimensió 3*3 ja que el text en clar té 3 caràcters diferents. La clau haurà de tenir 9 caràcters, 3*3, on k=XIFRARMES.

$$\begin{bmatrix} M & E & S \\ X & I & F \\ R & A & R \end{bmatrix} * \begin{bmatrix} T \\ F \\ G \end{bmatrix} = \begin{bmatrix} 12 & 4 & 19 \\ 24 & 8 & 5 \\ 18 & 0 & 18 \end{bmatrix} * \begin{bmatrix} 20 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 240 & 20 & 114 \\ 480 & 40 & 30 \\ 360 & 0 & 108 \end{bmatrix}$$

Figura 2.8: Xifratge de Hill

x_1=(12*20+4*5+19*6) mod 27= (240+20+114) mod 27= 374 mod 27= 23= W
x_2=(24*20+8*5+5*6) mod 27= (480+40+30)= 550 mod 27= 10= K
x_3=(18*20+0*5+18*6) mod 27= (360+0+108)= 468 mod 27= 9= J
El criptograma es WKJ.

En definitiva, els sistemes de xifratge clàssics son sistemes molt senzills i lineals alhora de xifrar i desxifrar, resulta fàcil aplicar mètodes de criptoanàlisi, com per exemple fent ús de atacs per força bruta, atacs en utilitzar l'estadística amb la redundància del llenguatge o atacs amb text en clar conegut, tot això és per poder rompre el xifratge. Això els converteix en sistemes poc segurs i gens fiables. Un altre punt desfavorable de la criptografia clàssica, és que els seus algoritmes o tècniques de xifratge han de ser secrets i únicament coneguts per l'emissor i receptor del missatge, ja que si no fou així, la confidencialitat ja no estaria garantida.

3. Criptoanàlisi

3.1. Que és?

La criptoanàlisi consisteix a comprometre la seguretat d'un criptosistema. Això, és pot fer desxifrant un missatge sense conèixer la clau, o bé obtenint a partir d'un o més criptogrames la clau, que ha estat emprada en la seva codificació. En general, la criptoanàlisi se sol dur a terme estudiant grans quantitats de parells, missatge en clar – criptograma, generats amb la mateixa clau.

3.2. Alan Turing - Enigma

La maquina Enigma va ser inventada per un enginyer alemany, anomenat Arthur Scherbius, un expert en electromecànica que, després de la primera guerra mundial, va voler aplicar la tecnologia existent per a millorar els sistemes de criptografia dels exèrcits. La seva idea, patentada al febrer de 1918, consistia a aplicar xifratge de Vigenère. Com Scherbius no comptava amb recursos per a fabricar-la, es va associar amb Willie Korn que tenia una companyia anomenada Enigma Chiffiermaschinen AG a Berlin. Tots dos van millorar el disseny i en 1923 la van presentar en l'exhibició postal internacional de Berlín per al xifratge de secrets comercials. A continuació es pot veure una replica de la maquina Enigma i la seva estructura.

Figura 3.1: Maquina Enigma

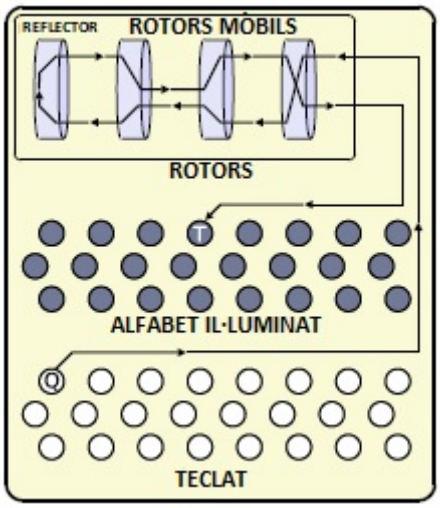


Figura 3.2: Estructura maquina Enigma

La màquina Enigma era un dispositiu electromecànic. El mecanisme consistia en una sèrie de tecles amb les lletres de l'alfabet, en realitat eren interruptors que accionaven els dispositius elèctrics i feien moure uns cilindres rotatoris. El funcionament cap a l'usuari era bastant senzill, ja que aquest havia de teclejar les lletres del seu missatge i anotar les lletres que retornava la màquina, a través d'un alfabet que s'anava il·luminant. El codi a usar es fixava

amb les posicions dels cilindres que constàvem, cadascun, de 26 cables que es connectaven al teclat però amb la particularitat, que el primer cilindre girava un vint-i-sisè de volta després de cada pulsació, de tal manera que la posició de les connexions anava canviant amb cada entrada del teclat i per tant s'obtenia un xifratge polialfabètic. A més, per a donar una major robustesa, el segon cilindre només donava un gir quan el primer havia fet 26 girs i el tercer quan el segon havia donat 26 girs i es va afegir la funcionalitat que els cilindres poguessin ser intercanviats de posició, de manera que el nombre de possibilitats augmentés fins a tenir 105.456 alfabets.

A més, el sistema comptava amb 6 cables de connexió que també permetien introduir modificacions atès que podrien endollar a 26 llocs, representat a les 16 lletres de l'alfabet d'Enigma, això produïa $101*10^9$ maneres diferents de connectar els cables que units als 105.456 alfabets donava $3,3*10^{113}$ possibilitats diferents de codificació

En 1933 l'Alemanya nazi va nacionalitzar la companyia Enigma Chiffiermaschinen AG i va passar a equipar a tot l'exèrcit amb aquestes maquines de xifratge, a les quals li van introduir un quart cilindre per complicar encara més el desxifratge dels missatges.

Durant la segona guerra mundial, l'Alemanya nazi comptava amb una gran avantatge, ja que el codi enigma era quasi indesxifrable i a més l'exèrcit alemany canviava cada dia el codi, de tal forma que els aliats comptaven amb un únic dia per a desxifrar-lo perquè a l'altre dia es tornava a canviar. Els alemanys en el missatge xifrat del dia, enviaven també la clau de l'endemà.

Alan Turing va ser un matemàtic i considerat un dels primers informàtics teòrics, va ser contractat perquè es dediques a la criptoanàlisi i va començar a investigar la maquina Enigma. Durant la segona guerra mundial la força aèria i de la marina alemanya van transmetre milers de missatges xifrats. Uns 30 analistes criptogràfics, entre els quals hi havia un grup dirigit per Turing van ajudar al fet que gran quantitat d'aquesta informació xifrada acabés en mans dels aliats. Això va ser possible, gracies a que Turing juntament amb Gordon Welchman van dissenyar una maquina que pogués contrarestar la potencia d'Enigma.

La maquina dissenyada per Turing juntament amb el seu col·lega, va començar a funcionar a la primavera de 1940 i ràpidament van començar a desxifrar els missatges de les forces aèries alemanyes. Però els missatges més importants per Anglaterra, eren els que enviaven els submarins alemanys, que operaven a l'atlàntic nord, i aquets utilitzaven un sistema Enigma amb major seguretat, disposaven de 8 rotors,això suposava que les possibles combinacions augmentessin exponencialment.

En aquest moment de la guerra, a causa de la major seguretat que oferien les maquines Enigma de la marina alemanya, Turing es va centrar en la variant naval; després de la captura d'uns documents alemanys, en els quals estaven anotades les claus per al més de febrer de 1941, el matemàtic i el seu equip van poder reconstruir els sistema usat pels alemanys, al desenvolupar una tècnica estadística, que permetia conèixer la identitat de cada rotor en la maquina Enigma, abans d'aplicar la seva pròpia màquina desxifradora al missatge.

La conseqüència dels desxifrats dels missatges alemanys, va permetre escurçar la guerra entre 2 i 4 anys. Cal considerar que durant la segona guerra mundial, és va produir una mitjana de 7 milions de morts a l'any, ens podem fer una idea de la magnitud de l'aportació de Turing alhora de salvar vides humanes.

4. Criptografia moderna

4.1. Que es un criptosistema?

Podem definir un criptosistema com un sistema xifrat, compost per:

- M, representa el missatge sense xifrar, es a adir, el text en clar.
- X, representa el missatge xifrat.
- K, representa la clau que s'utilitza.
- E, es la transformació a xifrat, representa la funció o funcions que s'apliquen a M per obtenir X.
- D, es la transformació de a desxifrat, representa la funció o funcions que s'apliquen a E per obtenir M.

Aquest sistema xifrat ha de complir sempre que el missatge xifrat, X, amb la clau, K, al tornar al missatge original, M, s'ha de desxifrar amb la clau K. On $D_k(E_k(m))=m$.

4.2. Orígens

La criptografia moderna neix a partir dels principis de Kerckhoffs, desenvolupats en el llibre *cryptographie militare* any 1883:

- El sistema ha de ser en la pràctica indesxifrable, en cas que no ho sigui matemàticament.
- El sistema no ha de ser secret i no ha de ser un problema que aquest caigui en mans de l'enemic.
- La clau del sistema ha de ser fàcil de memoritzar i comunicar a uns altres, sense necessitat d'haver d'escriure-la. Serà a més canviable i modificable per interlocutors vàlids.
- El sistema ha de poder aplicar-se a la correspondència telegràfica.
- El sistema ha de ser portables i el seu ús no haurà de requerir la intervenció de diverses persones.
- El sistema ha de ser de fàcil ús, no requerirà coneixements especials ni tindrà una llarga sèrie de regles o instruccions.

Com a síntesi d'aquets principis, podem concloure que la seguretat d'un criptosistema no ha de dependre de mantenir en secret l'algoritme de xifrat. La seguretat tan sols ha de dependre de mantenir la clau de xifrat en secret.

La criptografia moderna va agafar cos i consistència desprès de la publicació d'un parell de llibres i un article, on es van establir les bases de la nova criptografia. Hi ha dos moments clau en el segle XX on s'explica això:

- A finals de la dècada dels anys 40, Claude Shannon publica els llibres *A Mathematical Theory of Comunication* any 1948 i *Comunication Theory of Secrecy Systems* any 1949 on s'introduïa dos tècniques de xifrat de clau secreta o simètrica, per ocultar la redundància en un missatge en clar. Per una banda, hi ha el terme de la confusió que tracta d'ocultar la relació entre el missatge en clar i un criptosistema, i per altre banda, hi ha el terme de la difusió, que dilueix la redundància del text en clar repartint-la al llarg de tot el criptosistema. Avui dia, a l'utilitzar un xifrat de clau simètrica, trobem criptosistemes en bloc i de flux. El criptosistema en bloc es un sistema de xifratge que divideix el text en clar, en blocs de longitud fixa i realitza l'acció de xifratge sobre cada bloc, utilitzant la clau secreta compartida per emissor i receptor. El criptosistema de flux es un sistema de xifratge on el text en clar es combina amb un numero de caràcters pseudo-aleatoris.

- A finals de la dècada dels anys 70 i durant la dècada dels 80 es produiran els avanços conceptuals més importants, aquets en portaran fins avui dia. L'any 1976 Bailey Whitfield Diffie i Martin Hellman van publicar l'article *New directions in cryptography*, on es parlava per primer cop del concepte de criptografià asimètrica o clau publica, on cada usuari en una comunicació secreta disposava de dos claus, una publica i l'altre privada. Qualsevol usuari es podia comunicar amb el destinatari coneixent la clau publica d'aquest i el destinatari podia desxifrar la comunicació xifrada, ja que únicament ell coneixia la seva clau privada.

Aquets esdeveniments varen fer, que es comencessin a desenvolupar protocols criptogràfics en les xarxes de comunicació, com també per donar solució al problema de la distribució de claus. Tot això, ha permès i permet desenvolupar multitud de protocols, algoritmes i eines basats en aquets principis. En conseqüència, la criptografia s'ha convertit en un recurs obligatori per garantir la confidencialitat, integritat, autenticitat i no repudi.

En la criptografia moderna el xifratge es realitza damunt de bits o bytes, tant en el text en clar com en el criptograma, s'utilitzen els caràcters del codi ASCII per representar-los. El codi ASCII esta representat per 256 caràcters i cada caràcter esta constituït per 1 byte, o el que es el mateix per 8 bits.

Seguint els principis de Kerckhoffs, la fortalesa de la criptografia moderna ha de basar-se en la impossibilitat computacional de descobrir una clau secreta única, quan l'algoritme de xifratge es coneix i per tant es públic. Per impossibilitat computacional s'entén que matemàticament sí que és possible resoldre el problema, però que, a causa dels desorbitat

requeriments que ha de disposar la computadora, juntament al temps i els diners invertits, fan impossible abordar el descobriment d'aquesta clau secreta.

EL xifratge modern, el poden classificar d'acord a com és tracta l' informació abans de xifrar-la, distingint per una banda el xifrat de flux i per l'altre el xifrat en bloc. A més, podem dividir les distincions anteriors segons el tipus de clau utilitzada, diferenciant entre la criptografia simètrica o clau privada i la criptografia asimètrica o clau publica. A continuació mostro la classificació de la criptografia moderna:

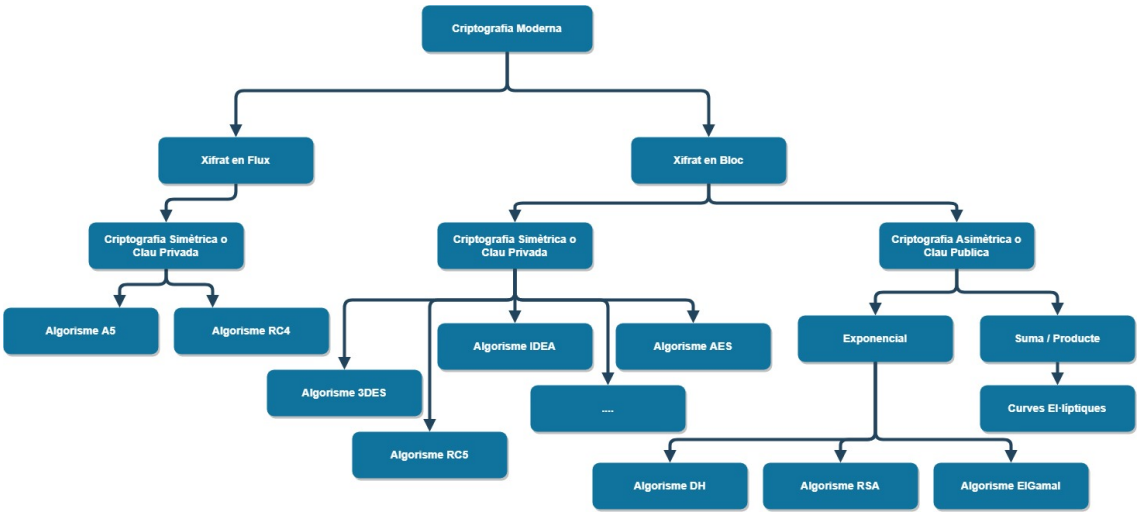


Figura 4.1: Classificació criptografia moderna

4.3. Criptografia simètrica

4.3.1. En bloc

El criptosistema de clau privada o simètric amb xifrat en bloc s'implementa com s'explica a continuació. S'usa una clau única que comparteixen emissor i receptor i que no canviarà durant el procés de xifrat. La informació a xifrar s'agrupa normalment en blocs de 128 bits i se li aplica operacions de permutació, substitució i de tipus XOR. Cada bloc de text en clar es converteix en un criptograma.

Perquè els bits del text en clar es barregin prou amb els bits de la clau, caldrà aplicar operacions de permutació, substitució i de tipus XOR, en aquets bits diverses vegades. En criptografia a aquestes operacions repetitives se'n diu voltes o rondes de l'algorisme. Amb això s'aconsegueix, entre altres característiques interessants, el denominat efecte d'allau, això es, que el canvi d'un sol bit en l'entrada o en la clau produeixi un canvi en aproximadament el 50% dels bits de sortida.

De la mateixa forma que els diferents blocs del missatge en clar estan concatenats formant el missatge, el criptograma final serà el resultat de la concatenació dels diferents criptogrames, aquets també anomenats subcriptogrames.

En la següent imatge, podem observar l'esquema d'un criptosistema simètric en bloc, on es visualitza les operacions que es realitzen en cada fase de l'algorisme. Cada bloc es xifra durant diverses voltes, en les quals s'usa una subclau, que es calcula a partir de la clau de xifrat K amb una funció que es coneix com a expansió de clau, de manera que en cada volta aquesta sigui diferent.

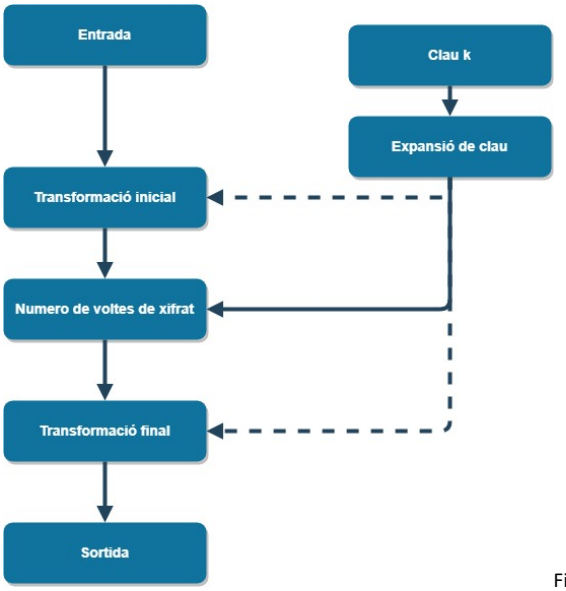


Figura 4.2: Criptosistema simètric en bloc

4.3.2. De flux

El criptosistema de clau privada o simètric amb xifrat de flux, s'implementa com s'explica a continuació. El missatge en clar es xifrat bit a bit o byte a byte, amb una seqüència de bits o bytes que formen la clau k, coneguda com seqüència de clau S_i, la clau k es intercanviada entre emissor i receptor, no es tracta d'una clau on el seu valor es únic sinó que va canviant a mida que es recorre el missatge en clar. Per això és recomana, que la seqüència de la clau tingui almenys tants bits com el missatge en clar, per evitar que sigui periòdica. Mitjançant un algorisme és genera la seqüència xifrant S_i i amb la funció XOR es xifren bit a bit el missatge en clar. Per realitzar el desxifrat del criptograma es torna a aplicar la funció XOR amb la mateixa seqüència S_i per obtenir el missatge en clar. Aquest tipus de xifrat el podem resumir en 3 característiques:

1. Un algorisme de xifrat i desxifrat basat en la funció XOR.

2. Una seqüència xifrant binaria i pseudo-aleatòria, anomenada S_i, que s'obté de la clau secreta k, compartida entre emissor i receptor. A més d'un algorisme generador de seqüència S_i.

3. L'operació de xifrat i desxifrat es realitza de la mateixa forma.

Per tant, la robustesa del criptosistema simètric de flux estarà en els trets de la seqüència de clau S_i. Aquesta seqüència serà pseudo-aleatòria si compleix les regles de Golomb:

- La probabilitat de trobar zeros o uns en qualsevol posició de la seqüència, sempre que sigui parell, haurà de ser un 50%. Si la seqüència és imparell, hi haurà un zero menys que uns.
- La probabilitat de trobar un zero o un, en un bit independent de la seqüència coneguda anteriorment ha de ser un 50%.
- L'autocorrecció fora de fase, haurà de tenir el mateix nombre d'encerts que d'errades per tots els desplaçaments aplicats a la seqüència, des de 1 fins n-1 bits. Això, significa que independentment del lloc en la que comencem a analitzar la seqüència xifrant, no s'obté una major informació o avantatge per la seva anàlisi, ni tampoc dades del seu comportament.

Al utilitzar un criptosistema simètric, obtenim els conceptes bàsics de seguretat de la informació:

- Confidencialitat, només aquells que estan autoritzats tindran accés a la informació.
- Integritat, les dades no poden ser alterades.

4.4. Criptografia asimètrica

El criptosistema de clau publica o asimètric amb xifrat en bloc s'implementa com s'explica a continuació. Aquest sistema criptogràfic està basat en el fet que la clau no es única i secreta, en estar composta per dues parts, una pública i coneguda per tot el món, i l'altre secreta i només coneguda pel seu propietari. Mitjançant propietats matemàtiques dels cossos finits, totes dues claus són inverses entre si i només el seu propietari pot calcular l'altre. Qualsevol usuari que conegui la clau publica d'un altre usuari, li serà computacionalment quasi impossible descobrir la clau privada corresponent.

Com cada usuari posseeix dues claus, una pública i l'altre privada, inverses entre si dins d'un cos, de manera que el que una xifra l'altre ho desxifra, és podrà realitzar les següents operacions:

- Xifrar amb la clau publica del destinatari: si és xifra un missatge m amb la clau publica del destí, llavors solament aquest destinatari serà capaç de desxifrar el criptograma enviat, perquè només el destinatari esta en possessió de la clau privada, inversa de la clau publica, que desfà la xifra i recupera així el missatge m. Per tant, en aquest cas el missatge m s'haurà enviat de manera secreta i confidencial. Això, és el que en la practica, és coneix com a intercanvi segur de clau.

- Xifrar amb la clau privada de l'emissor, firma digital: si l'emissor xifra un missatge m amb la seva clau privada, podrà desxifrar aquest criptograma i recuperar aquest missatge m, qualsevol que disposi de la clau publica de l'emissor. El missatge m en aquest cas no és un secret, perquè el poden recuperar tots els usuaris que coneguin la clau publica de l'emissor, però, no obstant això, els qui el desxifren poden estar segurs que aquest missatge es d'aquest emissor i no d'un altre persona, perquè ningú excepte aquesta persona, coneix la clau privada que s'ha usat en la xifra. En aquest cas s'obté autenticitat, integritat i no repudi.

Al utilitzar un criptosistema de clau publica, obtenim els conceptes bàsics de seguretat de la informació:

- Confidencialitat, només aquells que estan autoritzats tindran accés a la informació.
- Integritat, les dades no poden ser alterades.
- Autenticació, ningú pot suplantar l' identitat d'un altre.
- No repudi, es preserva que alguna de les parts negui algun o acció pres amb anterioritat.

A la imatge següent, realitzo una petita comparativa entre criptografia de clau simètrica i asimètrica.

	Gestió de claus	Intercanvi de clau i certificat digital	Grandària de Claus	Duració de la clau sessió	Taxa de Xifrat
Criptografia Simètrica	Igual al numero de connexions	NO	Entre 128 a 256 bits	Segons o minuts	Centenars de Mbytes per segon
Criptografia Asimètrica	1	SI	Entre 2.048 a 4.096 bits	Entre 2 i 4 anys	Centenars de Kbytes per segon

Figura 4.3: Comparació entre criptografia simètrica i asimètrica

5. Algorismes més utilitzats

5.1. Algorisme simètric en blocs: AES (Advanced Encryption Standard)

A l'any 2000 el NIST (National Institute for Standards and Technology) anuncia l' utilització de l'algorisme Rijndael com a estàndard avançat de xifratge simètric, més conegut com algorisme AES. Aquest algorisme utilitza la mida del bloc a 128 bits i la clau amb una longitud de 128, 192 o 256 bits. AES fixa el bloc en una matriu d'estat, de mida 4*4, on hi ha representades 16 celes d'un byte cadascuna, dins de cada cela trobem 2 dígits en hexadecimal, on és va modificant el seu valor durant el transcurs dels diferents processos que s'executen al algorisme, fins arribar al criptograma.

Per descriure el procés de funcionament de l'algorisme AES, utilitzaré una clau de longitud de 128 bits, ja que es l'opció mes utilitzada.

Durant el procés de xifrat podem diferenciar dos parts:

- Xifrat del missatge amb les voltes. El xifrat del bloc, es realitza aplicant 4 funcions diferents :
    - Funció SubBytes, consisteix en modificar els valors de cada cela de la matriu d'estat. És realitza una substitució del valor de la cela per el valor d'una taula de substitució, també coneguda com S-Box. A continuació podem observar la taula S-Box:

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figura 5.1: Taula S-Box

A continuació, podem observar la matriu d'estat original i la matriu d'estat una vegada aplicada la funció SubBytes:

Matriu d'estat

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

Matriu d'estat amb funció SubBytes

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

Figura 5.2: Aplicació funció SubBytes

- Funció ShiftRows, consisteix en realitzar en la segona fila una rotació cap a la esquerra de 1 byte, es a dir, una cela. També realitza una rotació cap a l'esquerra de 2 bytes, es a dir 2 celes, a la tercera fila. Finalment, realitza una rotació cap a l'esquerra de 3 bytes, 3 celes, a la quarta fila. A continuació, podem observar la matriu una vegada aplicada la funció ShiftRows:

Matriu d'estat

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

Matriu d'estat amb funció ShiftRows

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

Figura 5.3: Aplicació funció ShiftRows

- Funció MixColumns, consisteix en multiplicar cada columna de la matriu d'estat amb la matriu de camp Galois, obtenint una nova columna. Aquesta funció permet augmentar la difusió.

Matriu d'estat

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

Matriu de camp Galois

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Figura 5.4: Matriu d'estat i matriu de camp Galois

Les operacions que s'implementen son amb polinomis.

A continuació, podem observar l'aplicació d'aquesta funció amb la primera columna de la matriu d'estat:

Matriu de camp Galois					1ª Columna matriu d'estat		1ª Columna matriu d'estat amb funció MixColumns
02	03	01	01		d4		04
01	02	03	01		bf		66
01	01	02	03	*	5d	=	81
03	01	01	02		30		e5

Figura 5.5: Aplicació funció MixColumns 1ª columna

Per obtenir el resultat de la primera cela de la nova matriu, a mode d'exemple, hem de realitzar els següents passos:

02 * d4 XOR 03 * bf XOR 01 * 5d XOR 01 * 30

A continuació, convertim a binari la seqüència anterior:

10 * 11010100 XOR 11 * 10111111 XOR 01 * 01011101 XOR 01 * 00110000

A continuació, convertim a polinomi la seqüència anterior, aquest canvi el realitzem seguint unes normes bàsiques, on hi ha 1 posem una x i el seu exponent es la posició del bit. A més, hem de tenir en compte les següents regles:

1. Si $2x^{?}=0$
2. Si al polinomi hi ha un x^8 , cal realitzar una operació XOR amb el polinomi $x^8+x^4+x^3+x^1+1$ que es igual a 100011011

$x*(x^7+x^6+x^4+x^2)$ XOR $(x+1)*(x^7+x^5+x^4+x^3+x^2+x+1)$ XOR $1*(x^6+x^4+x^3+x^2+1)$ XOR $1*(x^5+x^4)$ = $x^8+x^7+x^5+x^3$ XOR $x^8+x^6+x^5+x^4+x^3+x^2+x+$ $x^7+x^5+x^4+x^3+x^2+x+1$ XOR $x^6+x^4+x^3+x^2+1$ XOR x^5+x^4= $x^8+x^7+x^5+x^3$ XOR $x^8+x^7+x^6+2x^5+2x^4+2x^3+2x^2+2x+$ 1 XOR $x^6+x^4+x^3+x^2+1$ XOR x^5+x^4= $x^8+x^7+x^5+x^3$ XOR $x^8+x^7+x^6+1$ XOR $x^6+x^4+x^3+x^2+1$ XOR x^5+x^4

A continuació, reduïm els polinomis que contenen grau 8, es a dir x^8:

$x^8+x^7+x^5+x^3$ XOR $x^8+x^4+x^3+x^1+1$= 110101000 XOR 100011011= 010110011

$x^8+x^7+x^6+1$ XOR $x^8+x^4+x^3+x^1+1$= 111000001 XOR 100011011= 011011010

Finalment, realitzem operació XOR amb la seqüència obtinguda, per aconseguir el resultat de la primer cela:

10110011 XOR 11011010 XOR 00101111 XOR 00110000= 00000100= 04

A la següent imatge, podem observar el resultat final de la matriu d'estat una vegada aplicada aquesta funció:

Matriu d'estat amb funció MixColumns

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

Figura 5.6: Matriu resultat aplicació funció MixColumns

- Funció AddRoundKey, consisteix en realitzar operació XOR entre les columnes del mateix ordre de la matriu d'estat i la matriu de clau o subclau de volta.

Matriu d'estat

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

Matriu subclau de volta

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

Figura 5.7: Matriu d'estat i matriu subclau de volta

A continuació, podem observar l'aplicació d'aquesta funció amb la primera columna de la matriu d'estat i subclau de volta:

1ª Columna matriu d'estat		1ª Columna matriu subclau de volta		1ª Columna matriu d'estat amb funció AddRoundKey
04		a0		a4
66		fa		9c
81	XOR	fe	=	7f
e5		17		f2

Figura 5.8: Aplicació funció AddRoundKey 1ª columna

04 XOR a0= 00000100 XOR 10100000= 10100100= a4
66 XOR fa= 01100110 XOR 11111010= 10011100= 9c

81 XOR fe= 10000001 XOR 11111110= 01111111= 7f
e5 XOR 17= 11100101 XOR 00010111= 11110010= f2

A la següent imatge podem observar el resultat final de la matriu d'estat, una vegada aplicada aquesta funció:

Matriu d'estat amb funció AddRoundKey

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

Figura 5.9: Matriu resultat aplicació funció AddRoundKey

En la següent imatge, podem observar les diferents funcions o operacions que es realitzen en les diferents voltes, amb l'objectiu de xifrar un bloc de text en clar.

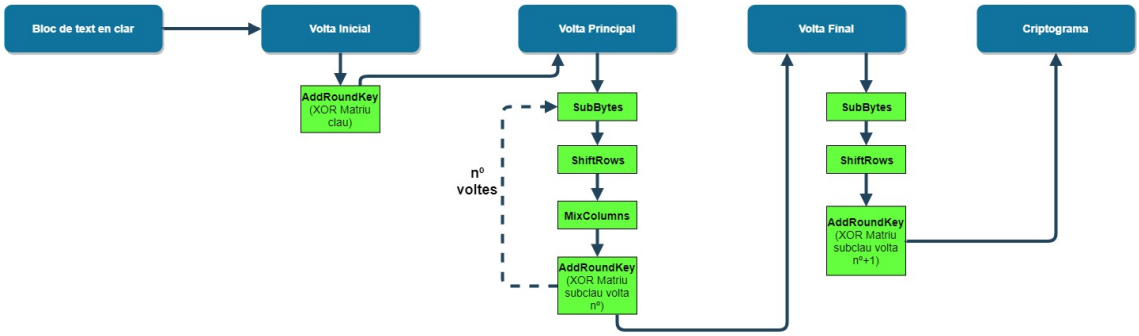


Figura 5.10: Criptosistema de xifrat AES

La mida de la clau utilitzada pot ser de:

- 128 bits, hi haurà 1 volta inicial, 9 voltes principals i 1 volta final. En la volta inicial, s'utilitzarà una matriu de clau de xifrat per realitzar l'operació AddRoundKey i en les voltes principals i volta final per realitzar aquesta mateixa funció, s'utilitzarà matriu de subclau de volta n, sent n el numero de volta en aquell moment. Per tant, hi haurà 10 matrius de subclau.

- 192 bits, hi haurà 1 volta inicial, 11 voltes principals i 1 volta final. En la volta inicial, s'utilitzarà una matriu de clau de xifrat per realitzar l'operació AddRoundKey i en les voltes principals i volta final per realitzar aquesta mateixa funció, s'utilitzarà matriu de subclau de volta n, sent n el numero de volta en aquell moment. Per tant, hi haurà 12 matrius de subclau.

- 256 bits, hi haurà 1 volta inicial, 13 voltes principals i 1 volta final. En la volta inicial, s'utilitzarà una matriu de clau de xifrat per realitzar l'operació AddRoundKey i en les voltes principals i volta final per realitzar aquesta

mateixa funció, s'utilitzarà matriu de subclau de volta n, sent n el numero de volta en aquell moment. Per tant, hi haurà 14 matrius de subclau.

- La codificació de la clau amb la funció expansió de clau.

  Al finalitzar la volta inicial, s'inicia la volta principal, en aquesta fase és té que trobar les matrius de subclau, per després poder aplicar la funció AddRoundKey. Per a trobar aquesta matriu, s'utilitza la funció d'expansió de clau.

  A continuació, podem observar la clau de 128 bits mitjançant la matriu de clau i la taula Rcon, aquesta darrera l'utilitzarem per fer operacions XOR.

Matriu de clau

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Taula Rcon

01	02	04	08	10	20	40	80	1b	36	6c	d8	ab	4d	9a
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figura 5.11: Matriu de clau i taula Rcon

La funció d'expansió de clau s'implementa de la següent forma:

- Per obtenir la primera columna de la nova matriu de subclau n, hem de realitzar les següents accions:

  1. Funció RotWord, consisteix en seleccionar la darrera columna de la matriu de clau o de subclau i realitzar un desplaçament vertical d'un bit cap a dalt. En la següent imatge, observem l'aplicació d'aquesta funció:

Darrera columna matriu de clau

09
cf
4f
3c

Darrera columna matriu clau amb funció RotWord

cf
4f
3c
09

Figura 5.12: Aplicació funció RotWord

  2. Funció SubBytes. A continuació, observem com s'aplica la funció SubBytes a la columna que se li aplicat anteriorment la funció RotWord:

Columna amb funció RotWord	Columna amb funció SubBytes
cf	8a
4f	84
3c	eb
09	01

Figura 5.13: Aplicació funció SubBytes

3. Seleccionem la quarta columna anterior i realitzem una operació XOR amb la columna que hem aplicat la funció SubBytes i tornem a realitzar operació XOR, amb la columna de la taula Rcon. Si volem implementar la matriu de subclau de volta 1, seleccionarem la columna 1 d'aquesta taula. Si volem implementar la matriu de subclau de volta 2, seleccionarem la columna 2 d'aquesta taula, i així successivament. A continuació, podem observar com obtenim la primera columna de la matriu de subclau de volta 1:

4ª columna Matriu de clau amb funció RotWord i SubByte		1ª Columna Matriu de clau		1ª Columna Taula Rcon		1ª Columna Matriu subclau volta 1
8a		2b		01		a0
84		7e		00		fa
eb	XOR	15	XOR	00	=	fe
01		16		00		17

Figura 5.14: 1ª Columna matriu subclau volta 1

8a XOR 2b XOR 01= 10001010 XOR 00101011 XOR 00000001= 10100000= a0
84 XOR 7e XOR 00= 10000100 XOR 01111110 XOR 00000000= 11111010= fa
Eb XOR 15 XOR 00= 11101011 XOR 00010101 XOR 00000000= 11111010= fe
01 XOR 16 XOR 00= 00000001 XOR 00010110 XOR 00000000= 00010111= 17

- Per obtenir la segona, tercera i quarta columna de la nova matriu de subclau n, hem de realitzar la següent acció:

    1. Cal realitzar operació XOR amb la darrera columna i la quarta columna anterior. A Continuació, visualitzem la segona columna de la matriu de subclau de volta 1:

1ª Columna anterior		4ª Columna anterior		2ª Columna Matriu subclau volta 1
a0		28		88
fa	XOR	ae	=	54
fe		d2		2c
17		a6		b1

Figura 5.15: 2ª Columna matriu subclau volta 1

a0 XOR 28= 10100000 XOR 00101000= 10001000= 88
fa XOR ae= 11111010 XOR 10101110= 01010100= 54
fe XOR d2= 11111110 XOR 11010010= 00101100= 2c
17 XOR a6= 00010111 XOR 10100110= 10110001= b1

L'estat final de la matriu subclau de volta 1 es:

Matriu subclau de volta 1

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

Figura 5.16: Matriu subclau de volta 1

El procés de desxifrat, consisteix en substituir les funcions de xifrat per les seves inverses i girar l'ordre en que s'implementen, a continuació s'observa l' implementació del criptograma AES per desxifrar:

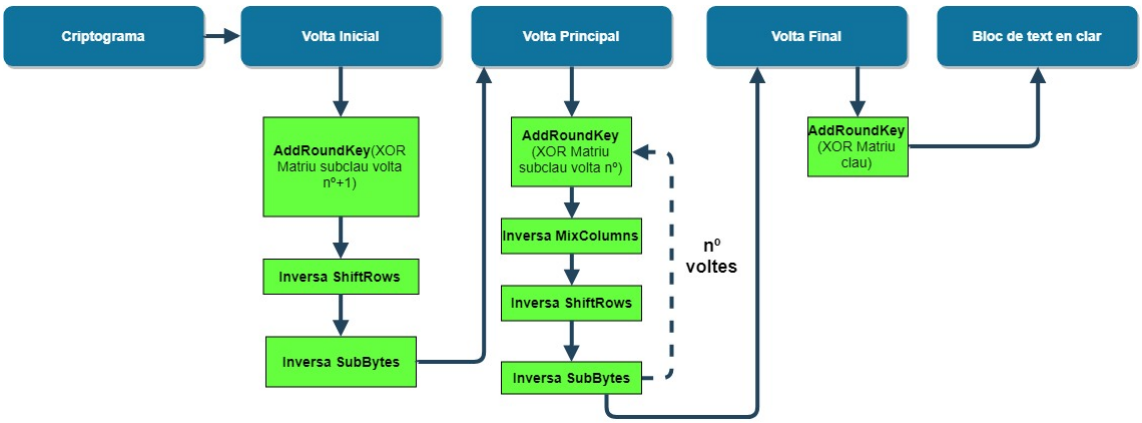


Figura 5.17: Criptosistema de desxifrat AES

Les operacions que s'implementen en el procés de desxifrat son:

1. Funció Inversa AddRoundKey, es idèntica a la funció AddRoundKey i per tant, consisteix en realitzar operació XOR entre les columnes del mateix ordre de la matriu d'estat i la matriu de clau o subclau de volta.

2. Funció Inversa MixColumn, consisteix en multiplicar cadascuna de les columnes per la inversa de la matriu de camp de Galois. La matriu inversa es la següent:

Matriu inversa de camp de Galois

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

Figura 5.18: Matriu inversa de camp de Galois

3. Funció Inversa ShiftRows, consisteix en realitzar una rotació cap a la dreta de 1 byte, es a dir, una cela, a la segona fila. Realitzar una rotació cap a la dreta de 2 bytes, es a dir 2 celes, a la tercera fila. I finalment realitzar una rotació cap a la dreta de 3 bytes, 3 celes, a la quarta fila.

4. Funció Inversa SubBytes, consisteix en modificar els valors de cada cela de la matriu. És realitza una substitució del valor de la cela, pel valor d'una taula inversa de substitució, també coneguda com Inversa S-Box. A continuació, podem observar la taula Inversa S-Box:

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figura 5.19: Taula inversa S-Box

Les diferents subclaus que s'utilitzen en el procés de desxifratge, són les mateixes que les del procés de xifrat però amb la diferencia que s'han d'utilitzar en ordre invers.

Al utilitzar AES, la seva seguretat s'implementa en la longitud de la clau, con més gran és la grandària de la clau, més voltes s'han de realitzar al xifrar. En definitiva, no existeixen claus

dèbils, degut a que es cerca eliminar la simetria en les subclaus, amb la qual cosa fa que sigui resistent a criptoanàlisi. En conseqüència, AES es l'algorisme simètric més segur que hi ha actualment.

5.2. Algorisme simètric de flux: CHACHA20

Ens trobem amb l'algorisme simètric per flux més importat i segur dels pròxims anys, Google ho porta implementat al seu navegador Chrome, al realitzar connexions HTTPS, des de l'any 2014. Va ser definit en articles científics, pel professor Dan Bernstein a l'any 2008. Amb una primera pinzellada, es podria dir que a partir d'una seqüència de claus, generada prèviament amb la funció quart de volta, es podria xifrar amb una operació XOR el text en clar.

La matriu de xifrat que s'utilitzarà per obtenir la seqüència de claus, té una mida de 4x4, per tant té 16 elements i cadascun d'aquets té una mida de 32 bits. L'estat inicial de la matriu de xifrat la podrien representar amb la següent imatge:

Constant 1	Constant 2	Constant 3	Constant 4
k_0	k_1	k_2	k_3
k_4	k_5	k_6	k_7
Comptador Bloc	Nonce	Nonce	Nonce

Figura 5.20: Matriu de xifrat

En aquesta matriu podem diferenciar:

1. Els primers 4 elements, corresponen a les posicions 0 a 3, són constants fixes on:
   - Constant 1= 61707865h
   - Constant 2= 3320646eh
   - Constant 3= 79622d32h
   - Constant 4= 6b206574h

61707865	3320646e	79622d32	6b206574
k_0	k_1	k_2	k_3
k_4	k_5	k_6	k_7
Comptador Bloc	Nonce	Nonce	Nonce

Figura 5.21: Matriu de xifrat

2. Els següents 8 elements, corresponen a les posicions 4 a 11, son les subclaus de la clau K, on K={K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7} . La clau K té una mida de 256 bits, es divideix aquesta per subclaus de 32 bits de mida. L'ordre de col·locació a cada element de la matriu, és en ordre little-endyan. Tenint per exemple la següent clau k : 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f, a continuació podem observar com queda l'estat inicial de la matriu de xifrat:

61707865	3320646e	79622d32	6b206574
03020100	07060504	0b0a0908	0f0e0d0c
13121110	17161514	1b1a1918	1f1e1d1c
Comptador Bloc	Nonce	Nonce	Nonce

Figura 5.22: Matriu de xifrat

3. L'element que està a la posició 12, es el comptador de bloc. Aquest element és el numero de bloc de la matriu de xifrat. Si tenim una única matriu de xifrat aquest numero sol ser 1, però és podria establir en qualsevol numero. En el cas que tinguéssim la necessitat d'emprar dos matrius de xifrat, perquè el text en clar tingues més bits que una única matriu de xifrat, es a dir, més gran de 512 bits, la primera tindria el comptador de bloc a 1 i la segona a 2. En el nostre exemple, només haurem de necessitar una única matriu i per tant, el comptador de bloc serà 1. L'estat inicial de la matriu de xifrat, encara incomplert és:

61707865	3320646e	79622d32	6b206574
03020100	07060504	0b0a0908	0f0e0d0c
13121110	17161514	1b1a1918	1f1e1d1c
00000001	Nonce	Nonce	Nonce

Figura 5.23: Matriu de xifrat

4. Els elements de les posicions 13 a 15, son anomenats Nonce o vectors d'inicialització. L'ordre de col·locació a cada element de la matriu, és en ordre little-endyan. En el cas que una mateixa clau K, és utilitzada per diferents remitents, és importantissim per la seguretat, no utilitzar els mateixos vectors d'inicialització. En el exemple, el vector d'inicialització es 000000090000004a00000000. L'estat complert inicial de la matriu xifrat es:

61707865	3320646e	79622d32	6b206574
03020100	07060504	0b0a0908	0f0e0d0c
13121110	17161514	1b1a1918	1f1e1d1c

00000001	09000000	4a000000	00000000

Figura 5.24: Matriu de xifrat

Per a poder arribar a la seqüència de claus, hem d'aplicar la funció quart de volta a la matriu de xifrat, aquesta funció es l'operació bàsica de Chacha20. La funció quart de volta està definida en quatre parts per les següents operacions:

1. a= a+b ; d= d XOR a; d= rotació de 16 bits a l'esquerra de l'element d.
2. c= c+d ; b= b XOR c; b= rotació de 12 bits a l'esquerra de l'element b.
3. a= a+b ; d= d XOR a; d= rotació de 8 bits a l'esquerra de l'element d.
4. c= c+d ; b= b XOR c; b= rotació de 7 bits a l'esquerra de l'element b.

On (a,b,c,d) son 4 del 16 elements de la matriu de xifrat. Al seleccionar aquets 4 elements, se li aplica la funció quart de volta i un cop obtinguts els nous valors, és substitueixen pels valors antics de la matriu de xifrat. La funció quart de volta implementa l'estructura ARX (add roll XOR), això són les operacions que es realitzen.

Hi ha que aplicar 4 vegades la funció quart de volta a cadascuna de les 20 voltes. Hi trobem 10 voltes anomenades voltes de columna, aquestes són les voltes imparells i 10 voltes anomenades voltes en diagonal, aquestes son les voltes parells.

A les voltes de columna cal aplicar:

- 1ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (0, 4, 8, 12).
- 2ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (1, 5, 9, 13).
- 3ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (2, 6, 10, 14).
- 4ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (3, 7, 11, 15).

A les voltes diagonals cal aplicar:

- 1ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (0, 5, 10, 15).
- 2ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (1, 6, 11, 12).
- 3ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (2, 7, 8, 13).

- 4ª Funció quart de volta, amb les posicions dels elements de la matriu de xifrat (3, 4, 9, 14).

Al acabar les 20 voltes, 10 voltes en columna i 10 voltes diagonal, cal treure de la matriu de xifrat el valor dels 16 elements de 32 bits que conté, abans de realitzar aquesta acció, cal treure els valors dels elements en ordre little-endian. Finalment, al treure el valors dels elements de la matriu de xifrat, s'obté la seqüència de claus o també anomenada keystream, aquesta al aplicar una operació XOR amb el text en clar s'obté el criptograma.

El xifratge del text en clar es pot realitzar esperant que tota la seqüència de claus estigui disponible o xifrant per blocs o elements, a mida que aquets van sortint de la matriu de xifratge, abans que sortí el següent bloc, estalviant una mica de memòria i per tant xifrant en temps real.

En la següent imatge, s'observa les diferents funcions o operacions que es realitzen en Chacha20, amb l'objectiu de xifrar un bloc de text en clar.

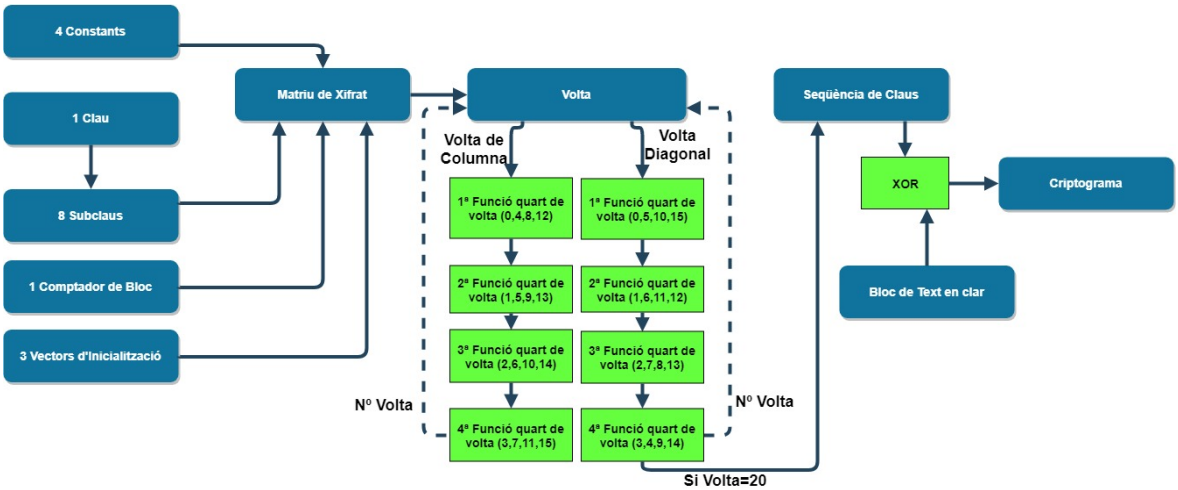


Figura 5.25: Criptosistema de xifrat Chacha20

El procés de desxifrat, consisteix en aplicar operació XOR entre el criptograma i la seqüència de claus. A continuació, s'observa l' implementació de l'estructura per desxifrar:

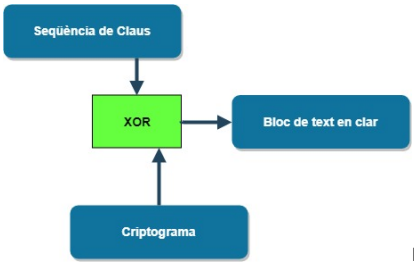


Figura: 5.26: Criptosistema encongit de xifrat Chacha20

Trets substancials de Chacha20 respecte AES:

- És més eficient i rapit que AES, en software, ja que aconsegueix un rendiment 3 vegades superior, a més aprofita millor la capacitat del CPU reduint un 16% l'ample de banda utilitzat.
- És més simple i fàcil d'implementar que AES.
- S'utilitza majoritàriament en connexions HTTPS, SSH, entre altres.
- No utilitza taula S-Box, sinó que té una estructura ARX. I per tant, tant sols utilitza mètode de transposició per aconseguir difusió.
- Necessita de 4 constants, 8 subclaus, 1 comptador de bloc i 3 vectors d'inicialització per implementar inicialment la matriu de xifrat.
- Realitza 20 voltes.
- Utilització funció quart de volta, per obtenir seqüència de claus.
- Utilització de seqüència de claus, keystream, per xifrar text en clar.

5.3. Algorisme asimètric: RSA

En el febrer de 1978, tres investigadors anomenats Ron Rivest, Adi Shamir i Leonard Adleman de l'institut tecnològic de Massachusetts (MIT) van patentar l'algorisme RSA.

La seva fortalesa es basa, en la dificultat computacional de factoritzar un numero compost molt gran, producte de dos nombres primers grans i trobar aquets. Es a dir, per posar un exemple, es molt fàcil realitzar la multiplicació de dos nombres primers, però trobar aquets tenint tan sols el producte, es converteix encara avui en quasi impossible quan aquets nombres primers son molt grans. Això, requereix molt de temps i molts recursos computacionals i econòmics per tenir èxit, en conseqüència fa que sigui quasi impossible trobar-los.

Gràcies a la seva versatilitat i seguretat, s'ha convertit en l'estàndard de la criptografia asimètrica, essent utilitzat en les comunicacions actuals en Internet. Avui dia, s'ha d'utilitzar una mida com a mínim per cadascun dels nombres primers de 1024 bits, perquè l'algorisme sigui segur i en conseqüència tenir un producte de 2048 bits.

Per implementar RSA, cal realitzar les següents passes:

1. Seleccionar dos nombres primers, **p** i **q**. On p i q han de tenir una mida mínima de 1024 bits.
2. Realitzar l'operació **n**=**p***q**.

3. Realitzar l'operació **z**=(**p**-1)* (**q**-1)

4. Seleccionar un petit numero **k** que sigui nombre primer, a més també ha de complir que **z** no sigui divisible per k. On 1<**k**<**z**.

5. Trobar **j** on (**kj** mod **z**) = 1.

6. Publicar **n** i **k** com la clau pública.

7. Guardar **j** com la clau privada.

Al realitzar els passos anteriors, poden utilitzar les següents expressions per xifrar i desxifrar:

- Criptograma= (text en clar)^k mod n
- Text en clar= (Criptograma)^j mod n

A continuació implementaré RSA com exemple, on utilitzaré nombres primers baixos per fer la demostració, ja que ajudarà a l'explicació i per tant la seguretat serà quasi nul·la.

- p =3 i q=11.
- n=p*q; n=3*11; n=33.
- z=(p-1)*(q-1); z=(3-1)*(11-1); z=2*10; z=20.
- k=13. Seleccionen aquest per dos raons:
    - es nombre primer .
    - z no es divisible per k.
- k*j=1 (mod z) on 13*j=1(mod 20);13*j/20 amb rest 1; on 221/20 te rest 1; on 13*j=221; j=17 .
- La clau pública serà un conjunt de números (n, k), per tant (33, 13).
- Guardem clau privada j=17.

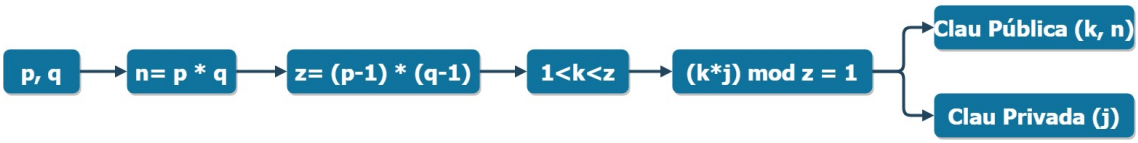


Figura 5.27: Generació de claus RSA

Ara ja podem aplicar l'expressió per xifrar:

- Criptograma= (text en clar)^k mod n; suposem que té el valor del text en clar es 18; on Criptograma=(18)^{13} mod 33; Criptograma=20822964865671168 mod 33; Criptograma=(630998935323368,727272-630998935323368)*33; Criptograma=0,727272 * 33 = 24. Per tant, 24 es el numero 18 xifrat per RSA.

Ara ja podem aplicar l'expressió per desxifrar:

- Text en clar= (Criptograma)^j mod n; Text en clar=(24)^{17} mod 33; Text en clar=29079779498268255741542 4 mod 33; Text en clar=(881205439341462295198 2,545454-8812054393414622951982)*33; Text en clar=0,545454 * 33 = 18.

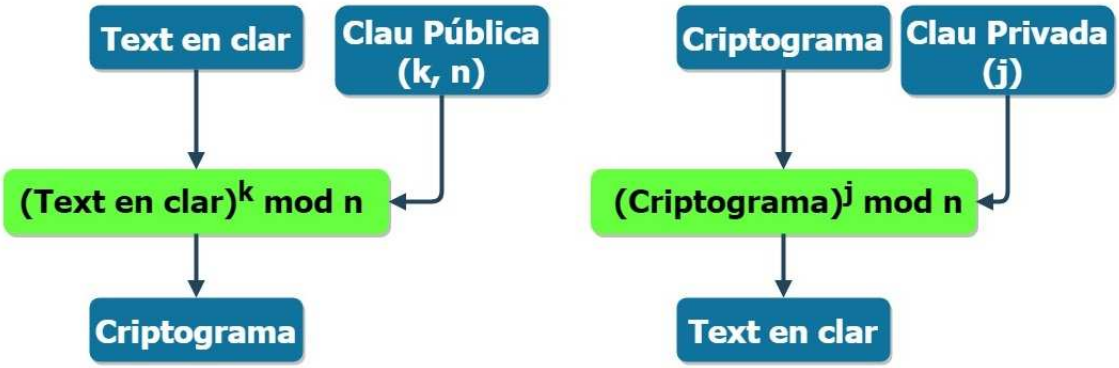


Figura 5.28: Criptosistema RSA per xifrar i desxifrar

Els Trets substancials de criptografia asimètrica respecte la simètrica:

- Elimina un dels principals problemes de la criptografia simètrica, l'intercanvi de claus.
- Consumeix més recursos de càlcul.
- Necessita molt més temps per realitzar el xifrat i el desxifrat

6. Aplicacions criptogràfiques

6.1. Protocols de comunicació segura

Els Protocols TCP/IP, realitzen l' implementació de la xarxa de Internet. S'organitzen en 5 capes apilades, on cadascuna d'aquestes capes es comuniquen amb la seva capa més propera inferior i superior amb la finalitat d'intercanviar informació. Es defineixen aquestes capes com:

1. Capa física, descriu les característiques físiques de la comunicació.
2. Capa d'enllaç, descriu com els paquets viatgen per el medi físic.
3. Capa de xarxa, descriu com fer arribar els paquets al seu destí.
4. Capa de transport, descriu com els paquets han arribat al seu destí i en l'ordre correcte.
5. Capa d'aplicació, les aplicacions que utilitzen Internet, accedeixen a aquesta capa per rebre i enviar les dades que necessiten.

Trobem en qualsevol capa del esquema TCP/IP, protocols que tenen com a finalitat obtenir comunicacions segures. Aquets protocols són utilitzats per crear una xarxa privada virtual, VPN. La VPN esta constituïda sobre la xarxa publica, es a dir, utilitza els seus recursos de forma privada. En la VPN es xifra utilitzant diversos algorismes, per obtenir confidencialitat i integritat a les dades transmeses entre dos llocs.

6.1.1. Protocol SSL (Secure Socket Layer)

Permet establir connexions segures a través d'Internet, la seva implementació es fàcil i transparent. El trobem entre la capa d'aplicació i transport donant seguretat a protocols com HTTP, SMTP, entre altres. La comunicació a través de SSL té 3 fases:

1. S'estableix connexió i negociació dels algorismes criptogràfics que s'utilitzaran en la comunicació.
2. Hi ha autenticació i intercanvi de claus entre els interessats, utilitzant mecanisme de clau publica, certificats digitals i firmes digitals.
3. Es xifra simètricament el tràfic.

La seva darrera versió es la 3.0, suporta gran quantitat d'algorismes criptogràfics, entre els que es troben RSA, AES i SHA1. Hi ha implementacions de SSL que permeten construir túnels SSL, que permeten dirigir qualsevol connexió a un port TCP mitjançant una connexió SSL prèvia, de manera transparent per les aplicacions que es connecten.

6.1.2. Protocol TLS (Transport Layer Security)

Esta basat en la versió 3.0 de SSL i a més incorpora millores, en conseqüència TLS es la següent generació de SSL. Aquestes millores són:

- Fa ús de funcions de resum MAC en lloc de funcions MDC.
- Numeració seqüencial de tots els camps que componen la comunicació.
- Protecció davant atacs que volen forçar l' utilització de versions més antigues o xifrats més dèbils.
- El missatge que finalitza la fase d'establiment de la comunicació, incorpora una signatura hash o també anomenada funció de resum de totes les dades entre els interessats.

Un dels grans avantatges de TLS respecte SSL, es que a partir d'una connexió TCP ja existent, pot ser inicialitzat, això permet continuar treballant amb els mateixos ports que els protocols no xifrats.

Aquest inclement de seguretat, té com a desavantatge que la velocitat de la connexió és redueix, ja que puja el nivell d'exigència computacional.

Entre les aplicacions més importants que utilitzen TLS, trobem OpenSSL, GnuTLS, BoringSSL, JSSE, S2n, entre altres.

6.1.3. Protocol IPSec

És un estàndard que proporciona confidencialitat i autenticació, en segons quins casos també proporciona integritat, als paquets IP. S'implementa a la capa de xarxa del model TCP/IP. Aquest estàndard no es tracta d'un únic protocol, sinó de varis.

És poden distingir dos tipus de modes d'operació:

1. Mode Transport, tan sols és xifra les dades del paquet IP. I per tant l'estàndard IPSec, intervé únicament en cada extrem de la comunicació.
2. Mode Túnel, és xifra tot el paquet IP. I per tant és crea una ruta segura entre dos equips de xarxa, es a dir, una VPN.

   Els túnels VPN IPSec en xarxes LAN, s'utilitzen per permetre la transmissió segura de dades, entre dos llocs remots. S'utilitza el protocol ISAKMP, Protocol d'administració de claus i associació de seguretat d'Internet també conegut com IKE, Internet Key Exchange. És el protocol encarregat de la negociació, permet a 2 equips posar-se d'acord alhora d'implementar una associació de seguretat IPSec.

Aquesta negociació té 2 fases:

- Fase 1. Es crea el primer túnel, que protegeix els missatges de la negociació ISAKMP posterior.
- Fase 2. Es crea el segon túnel, que protegeix les dades. IPSec s'activa per xifrar les dades utilitzant algorismes i proporcionar autenticació, confidencialitat i integritat.

Per obtenir autenticació d'usuaris i dispositius, es poden utilitzar:

- Claus pre-compartides entre els extrems. On s'introdueix manualment la clau a compartir, aquest mètode no és escalable, ja que cada IPSec es té que configurar punt a punt.
- Certificats Digitals, utilitzant algorisme RSA.

Per obtenir confidencialitat, s'utilitza un algorisme simètric.

Per obtenir integritat, s'utilitza una funció de resum, per garantir que els paquets no s'han modificat entre origen i destí.

També s'utilitza l'algorisme DH, Diffie-Hellman, per proporcionar un mètode d'intercanvi amb clau pública perquè els dos extrems determinin una clau secreta compartida.

IPSec utilitza els següents protocols per encapsular els paquets:

- IP AH, IP Authentication Header. Quan la confidencialitat no es requerida. Proporciona autenticació i integritat de dades.
- IP ESP, IP Encapsulating Security Payload. Proporciona confidencialitat, autenticació i integritat de dades.

6.2. Autenticació

L'autenticació consisteix en verificar l' identitat de qui o que hi ha a l'altre banda de la connexió, com també de saber si els missatges que rebem son integres. La verificació de l' identitat la podem obtenir a partir del certificat digital i la integritat del missatge amb la firma digital.

6.2.1. Certificat digital

Es un document digital que vincula una determinada clau pública a un usuari. Permet evitar l'atac de l'home en mig camí, on es suplanta l' identitat i s'envia la clau publica de l'atacant.

La informació que conté el certificat es:

- Numero de sèrie del certificat
- Identificació d'algorisme criptogràfic de signatura.
- Nom de l'entitat emissora.
- Període de validesa del certificat.
- Clau Pública.
- Identitat i dades de la persona o entitat propietària.
- Signatura digital per l'entitat emissora , CA.

El procés de certificació pot ser:

- Centralitzat. Es denota amb l'estàndard X.509, el que s'utilitza en la gran majoria d'aplicacions comercials.
- Descentralitzat. Utilitza el model PGP, Pretty Good Privacy. Es basa en la confiança que tenen els usuaris entre ells. Cada usuari genera el seu propi certificat i aquest el firmen les persones més properes.

6.2.2. Signatura digital

S'origina gràcies a l'aparició de la clau pública. Es pot considerar l'invers d'un esquema de xifratge de clau pública, es a dir, si l'emissor xifra un missatge m amb la seva clau privada, podrà desxifrar aquest criptograma i recuperar aquest missatge m, qualsevol que disposi de la clau publica de l'emissor. Els qui el desxifren poden estar segurs que aquest missatge es d'aquest emissor i no d'un altre persona, perquè ningú excepte aquesta persona coneix la clau privada que s'ha usat en la xifra.

Avui dia, la signatura digital es realitza mitjançant la funció de resum del missatge, juntament amb el propi missatge. Aquesta operació es de l'ordre de 1.000 cops més rapida, que xifrar tot el missatge amb la clau privada.

Les accions que cal fer per verificar una signatura digital són:

1. Llegir el certificat digital que acompanya la signatura, per conèixer les dades del propietari de la clau pública.
2. Verificar la signatura del certificat, feta per l'autoritat de certificació,CA.
3. Verificar el període de validesa del certificat, cal assegurar-se que no ha caducat.

4. Verificar que el certificat no ha sigut revocat a la llista de revocació de certificats, CRL.
5. Extreure la clau pública del certificat.
6. Verificar la signatura del missatge feta per l'emissor, ja que tenim la clau pública que s'ha extret del certificat digital.

7. Part practica

7.1. Packet Tracer

Implemento una VPN site-to-site, amb la finalitat de realitzar una comunicació segura basada en IPSec en mode túnel que compleixi els següents requisits:

- Algorisme de xifrat: AES256
- Verificador d'integritat amb funció Hash: SHA1
- Autenticació mitjançant clau compartida.

A la següent imatge podem observar l'esquema visual de la implementació.

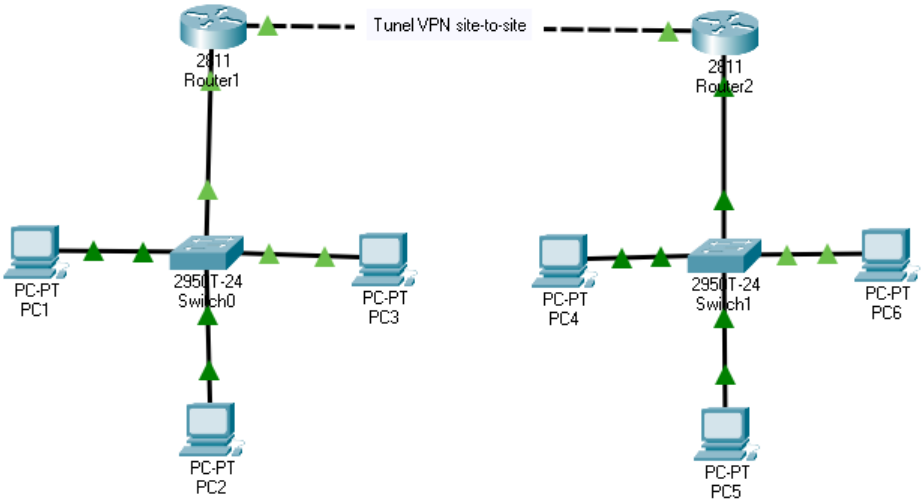


Figura 7.1: VPN site-to-site

Realitzo la configuració de les interfícies i l'encaminament estàtic del Router1 i , amb les següents comandes:

```
Router>ena
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Router1
Router1(config)#interface f0/0
Router1(config-if)#ip address 192.168.1.1 255.255.255.0
Router1(config-if)#no shutdown

Router1(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

Router1(config-if)#interface f0/1
Router1(config-if)#ip address 10.0.0.1 255.0.0.0
Router1(config-if)#no shutdown

Router1(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

Router1(config-if)#exit
Router1(config)#ip route 192.168.2.0 255.255.255.0 10.0.0.2
```

A continuació podem observar com queda la taula d'encaminament del Router 1:

```
Router1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/8 is directly connected, FastEthernet0/1
L       10.0.0.1/32 is directly connected, FastEthernet0/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet0/0
L       192.168.1.1/32 is directly connected, FastEthernet0/0
S    192.168.2.0/24 [1/0] via 10.0.0.2
```

Realitzo la configuració de les interfícies i l'encaminament estàtic del Router2 i , amb les següents comandes:

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Router2
Router2(config)#interface f0/0
Router2(config-if)#ip address 192.168.2.1 255.255.255.0
Router2(config-if)#interface f0/1
Router2(config-if)#ip address 10.0.0.2 255.0.0.0
Router2(config-if)#exit
Router2(config)#ip route 192.168.1.0 255.255.255.0 10.0.0.1
```

A continuació podem observar com queda la taula d'encaminament del Router 2:

```
Router2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/8 is directly connected, FastEthernet0/1
L       10.0.0.2/32 is directly connected, FastEthernet0/1
S    192.168.1.0/24 [1/0] via 10.0.0.1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, FastEthernet0/0
L       192.168.2.1/32 is directly connected, FastEthernet0/0
```

He de dir que m'ha hagués agradat poder utilitzar l'algorisme d'integritat SHA-256 i el grup 21 de DH però el Packet Tracer no els porta implementats en la simulació.

En la següent imatge podem observar les comandes que s'utilitzen al Router 1 per començar a configurar la negociació de la fase 1, on l'objectiu es configurar un canal de xifrat. Utilitzo el protocol d'administració de claus i associació de seguretat d'Internet (ISAKMP), que permetrà posteriorment als dos routers posar-se d'acord sobre com construir una associació de seguretat (SA) IPSec per a la fase 2. També es pot observar les comandes per configurar l'autenticació mitjançant clau compartida, la integritat utilitzant algorisme SHA, la confidencialitat utilitzant algorisme de xifrat AES 256 bits i finalment el grup 5 de Diffie-Hellman (DH) per proporcionar un mètode d'intercanvi de clau publica perquè els 2 pars estableixin una clau secreta compartida.

```
Router1(config)#crypto isakmp policy 3
Router1(config-isakmp)#?
  authentication  Set authentication method for protection suite
  encryption      Set encryption algorithm for protection suite
  exit            Exit from ISAKMP protection suite configuration mode
  group           Set the Diffie-Hellman group
  hash            Set hash algorithm for protection suite
  lifetime        Set lifetime for ISAKMP security association
  no              Negate a command or set its defaults
Router1(config-isakmp)#authentication pre-share
Router1(config-isakmp)#encription ?
% Unrecognized command
Router1(config-isakmp)#encryption ?
  3des  Three key triple DES
  aes   AES - Advanced Encryption Standard
  des   DES - Data Encryption Standard (56 bit keys).
Router1(config-isakmp)#encryption aes ?
  128  128 bit keys.
  192  192 bit keys.
  256  256 bit keys.
  <cr>
Router1(config-isakmp)#encryption aes 256
Router1(config-isakmp)#group ?
  1  Diffie-Hellman group 1
  2  Diffie-Hellman group 2
  5  Diffie-Hellman group 5
Router1(config-isakmp)#group 5
Router1(config-isakmp)#hash ?
  md5  Message Digest 5
  sha  Secure Hash Standard
Router1(config-isakmp)#hash sha
Router1(config-isakmp)#exit
```

A la següent imatge, configurem la clau d'autenticació compartida amb la IP 10.0.0.2, la clau es dissenyxarxes. En aquesta comanda finalitza la Fase1.

```
Router1(config)#crypto isakmp key dissenyxarxes address 10.0.0.2
```

A partir de la següent imatge comença la fase 2, aquesta fase té com objectiu establir la associació de seguretat (SA) IPSec, aquesta es un conjunt d'especificacions de tràfic, que informen al dispositiu que tràfic enviar per la VPN i com xifrar-lo i autenticar-lo, es a dir, els dos routers acorden un conjunt de paràmetres de comunicació. En aquesta imatge defineixo un conjunt de transformacions anomenades enc/dec per poder utilitzar els algorismes de xifrat i integritat:

```
Router1(config)#crypto ipsec transform-set enc/dec ?
  ah-md5-hmac   AH-HMAC-MD5 transform
  ah-sha-hmac   AH-HMAC-SHA transform
  esp-3des      ESP transform using 3DES(EDE) cipher (168 bits)
  esp-aes       ESP transform using AES cipher
  esp-des       ESP transform using DES cipher (56 bits)
  esp-md5-hmac  ESP transform using HMAC-MD5 auth
  esp-sha-hmac  ESP transform using HMAC-SHA auth
Router1(config)#crypto ipsec transform-set enc/dec esp-aes esp-sha-hmac
```

A continuació podem observar com realitzo l'assignació criptogràfica anomenada tunelVPN, que vincula tots els paràmetres de la fase 2 que son la IP de la interfície del router destí, el conjunt de transformacions enc/dec i la ACL VPN-STS. A més utilitzo el numero de seqüència 10, per realitzar l'assignació amb ipsec-isakmp.

```
Router1(config)#crypto map tunelVPN 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
Router1(config-crypto-map)#set peer 10.0.0.2
Router1(config-crypto-map)#set transform-set enc/dec
Router1(config-crypto-map)#match address VPN-STS
Router1(config-crypto-map)#exit
```

Finalment, en la següent imatge podem observar com realitzo vinculació entre l'assignació criptogràfica cmap i la interfície de sortida f0/1 del router 1.

```
Router1(config)#interface f0/1
Router1(config-if)#crypto map tunelVPN
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
Router1(config-if)#
```

En la següent imatge podem observar les comandes que s'utilitzen al Router 2 per començar a configurar la negociació de la fase 1. També es pot observar les comandes per configurar l'autenticació mitjançant clau compartida, la integritat utilitzant algorisme SHA, la confidencialitat utilitzant algorisme de xifrat AES 256 bits i finalment el grup 5 de Diffie-Hellman (DH).

```
Router2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router2(config)#crypto isakmp policy 3
Router2(config-isakmp)#?
  authentication  Set authentication method for protection suite
  encryption      Set encryption algorithm for protection suite
  exit            Exit from ISAKMP protection suite configuration mode
  group           Set the Diffie-Hellman group
  hash            Set hash algorithm for protection suite
  lifetime        Set lifetime for ISAKMP security association
  no              Negate a command or set its defaults
Router2(config-isakmp)#authentication pre-share
Router2(config-isakmp)#hash ?
  md5  Message Digest 5
  sha  Secure Hash Standard
Router2(config-isakmp)#hash sha
Router2(config-isakmp)#encryption ?
  3des  Three key triple DES
  aes   AES - Advanced Encryption Standard
  des   DES - Data Encryption Standard (56 bit keys).
Router2(config-isakmp)#encryption aes ?
  128  128 bit keys.
  192  192 bit keys.
  256  256 bit keys.
  <cr>
Router2(config-isakmp)#encryption aes 256
Router2(config-isakmp)#group ?
  1  Diffie-Hellman group 1
  2  Diffie-Hellman group 2
  5  Diffie-Hellman group 5
Router2(config-isakmp)#group 5
Router2(config-isakmp)#exit
```

A la següent imatge, configurem la clau d'autenticació compartida amb la IP 10.0.0.1, la clau es dissenyxarxes. En aquesta comanda finalitza la Fase1.

```
Router2(config)#crypto isakmp key dissenyxarxes address 10.0.0.1
```

A partir de la següent imatge comença la fase 2. En la següent comanda configuro una ACL anomenada VPN-STS, que accepta el tràfic IP que prové de la LAN del Router 2 cap a la LAN del router 1. La resta de tràfic que s'origina en les LAN's no es xifrarà.

```
Router2(config)#ip access-list extended VPN-STS
Router2(config-ext-nacl)#permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
Router2(config-ext-nacl)#exit
```

En aquesta imatge defineixo un conjunt de transformacions anomenades enc/dec per poder utilitzar els algorismes de xifrat i integritat:

```
Router2(config)#crypto ipsec transform-set enc/dec ?
  ah-md5-hmac   AH-HMAC-MD5 transform
  ah-sha-hmac   AH-HMAC-SHA transform
  esp-3des      ESP transform using 3DES(EDE) cipher (168 bits)
  esp-aes       ESP transform using AES cipher
  esp-des       ESP transform using DES cipher (56 bits)
  esp-md5-hmac  ESP transform using HMAC-MD5 auth
  esp-sha-hmac  ESP transform using HMAC-SHA auth
Router2(config)#crypto ipsec transform-set enc/dec esp-aes esp-sha-hmac
```

A continuació podem observar com realitzo l'assignació criptogràfica anomenada tunelVPN, que vincula tots els paràmetres de la fase 2 que son la IP de la interfície del router destí, el conjunt de transformacions enc/dec i la ACL VPN-STS. A més utilitzo el numero de seqüència 10, per realitzar l'assignació amb ipsec-isakmp.

```
Router2(config)#crypto map tunelVPN 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
Router2(config-crypto-map)#set peer 10.0.0.1
Router2(config-crypto-map)#set transform-set enc/dec
Router2(config-crypto-map)#match address VPN-STS
Router2(config-crypto-map)#exit
```

Finalment, en la següent imatge podem observar com realitzo vinculació entre l'assignació criptogràfica cmap i la interfície de sortida f0/1 del router 2.

```
Router2(config)#interface f0/1
Router2(config-if)#crypto map tunelVPN
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
Router2(config-if)#exit
```

A continuació podem observar l'esquema del packet tracer juntament amb una simulació d'enviament de paquets ICMP.

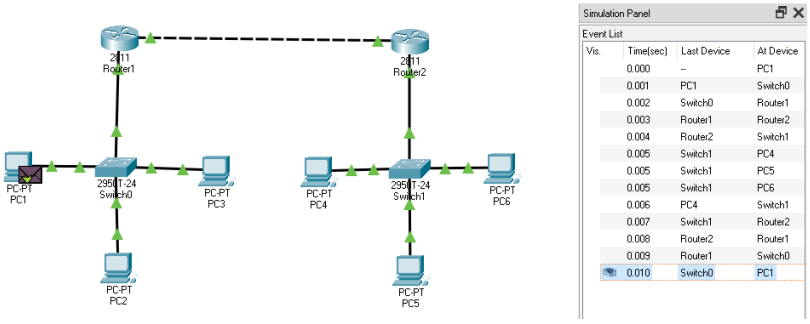


Figura 7.2: Simulació VPN site-to-site

A continuació verifico des del router 1 que s'ha creat un túnel VPN IPSec, es a dir, verifico les associacions de seguretat (SA), del router 1 a router 2.

```
Router1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
10.0.0.2        10.0.0.1        QM_IDLE           1016    0 ACTIVE


IPv6 Crypto ISAKMP SA
```

A la següent imatge podem observar com estan configurat tots els paràmetres de la fase 2, al router 1.

```
Router1#show crypto map
Crypto Map tunelVPN 10 ipsec-isakmp
        Peer = 10.0.0.2
        Extended IP access list VPN-STS
            access-list VPN-STS permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
        Current peer: 10.0.0.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                enc/dec,
        }
        Interfaces using crypto map tunelVPN:
                FastEthernet0/1
```

Amb la següent comanda, podem observar a la imatge com s'han xifrat 4 paquets i s'han desxifrat 2 paquets des del router 1.

```
Router1#show crypto ipsec sa

interface: FastEthernet0/1
    Crypto map tag: tunelVPN, local addr 10.0.0.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   remote  ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
   current_peer 10.0.0.2 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 0
   #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 0
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 1, #recv errors 0

     local crypto endpt.: 10.0.0.1, remote crypto endpt.:10.0.0.2
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/1
     current outbound spi: 0xA4467D78(2756083064)

     inbound esp sas:
      spi: 0xE800A1D0(3892355536)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2004, flow_id: FPGA:1, crypto map: tunelVPN
        sa timing: remaining key lifetime (k/sec): (4525504/2659)
        IV size: 16 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xA4467D78(2756083064)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2005, flow_id: FPGA:1, crypto map: tunelVPN
        sa timing: remaining key lifetime (k/sec): (4525504/2659)
        IV size: 16 bytes
        replay detection support: N
        Status: ACTIVE
```

A continuació verifico des del router 2 que s'ha creat un túnel VPN IPSec, es a dir, verifico les associacions de seguretat (SA), del router 2 a router 1.

```
Router2#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
10.0.0.1        10.0.0.2        QM_IDLE           1006    0 ACTIVE


IPv6 Crypto ISAKMP SA
```

A la següent imatge podem observar com estan configurat tots els paràmetres de la fase 2, al router 2.

```
Router2#show crypto map
Crypto Map tunelVPN 10 ipsec-isakmp
        Peer = 10.0.0.1
        Extended IP access list VPN-STS
            access-list VPN-STS permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
        Current peer: 10.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                enc/dec,
        }
        Interfaces using crypto map tunelVPN:
                FastEthernet0/1
```

Amb la següent comanda, podem observar a la imatge com s'han xifrat 2 paquets i s'han desxifrat 4 paquets des del router 2.

```
Router2#show crypto ipsec sa

interface: FastEthernet0/1
    Crypto map tag: tunelVPN, local addr 10.0.0.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
   remote  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   current_peer 10.0.0.1 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 2, #pkts encrypt: 2, #pkts digest: 0
   #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 0
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 0, #recv errors 0

     local crypto endpt.: 10.0.0.2, remote crypto endpt.:10.0.0.1
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/1
     current outbound spi: 0xE800A1D0(3892355536)

     inbound esp sas:
      spi: 0xA4467D78(2756083064)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2004, flow_id: FPGA:1, crypto map: tunelVPN
        sa timing: remaining key lifetime (k/sec): (4525504/2659)
        IV size: 16 bytes
        replay detection support: N
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xE800A1D0(3892355536)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2005, flow_id: FPGA:1, crypto map: tunelVPN
        sa timing: remaining key lifetime (k/sec): (4525504/2659)
        IV size: 16 bytes
        replay detection support: N
        Status: ACTIVE
```

7.2. OPENSSL

Amb l'aplicació OpenSSL realitzaré:

- Xifrat amb clau simètrica del document anomenat ArxiuTFG.
- Xifrat amb clau publica del document anomenat ArxiuTFG.
- Signatura del document anomenat ArxiuTFG amb clau privada.
- Verificació del signant, al document anomenat Signatura-ArxiuTFG amb clau publica.

En la següent imatge podem observar el contingut del document l'ArxiuTFG.

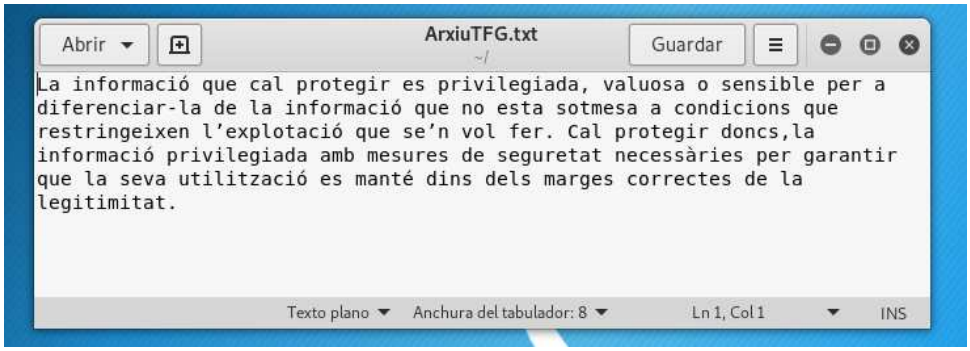


Figura 7.3: ArxiuTFG.txt

A continuació realitzo el xifrat amb l'algorisme AES amb una clau de 256bits. El document resultat serà ArxiuTFG-XifratAES.

```
root@kali:~#
root@kali:~# openssl enc -aes-256-cbc -in ArxiuTFG.txt -out ArxiuTFG-XifratAES.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
```

En la següent imatge es pot observar el contingut del document xifrat de l'ArxiuTFG, ara anomenat ArxiuTFG-XifratAES. La contrasenya que he utilitzat es treballfinalgrau.

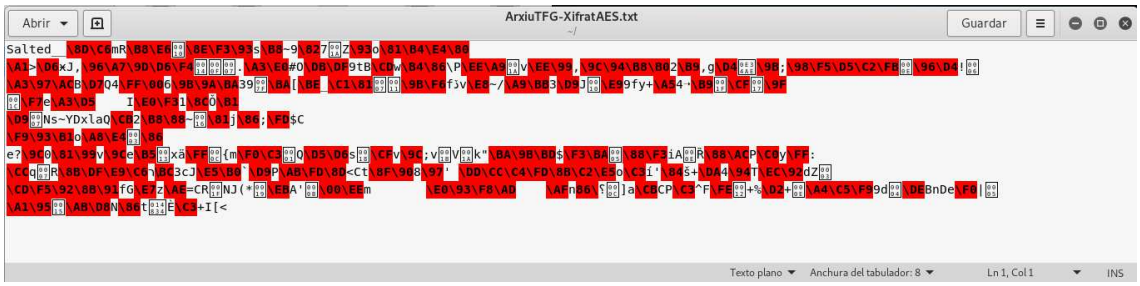


Figura 7.4: ArxiuTFG-XifratAES.txt

Ara procedeixo a desxifrar el document ArxiuTFG-XifratAES. El document resultant es gravarà amb el nom Desxifrar_ArxiuTFG-XifratAES. La contrasenya utilitzada per desxifrar es treballfinalgrau, es la mateixa que hem utilitzat per xifrar.

```
root@kali:~# openssl enc -aes-256-cbc -d -in ArxiuTFG-XifratAES.txt -out Desxifrar_ArxiuTFG-XifratAES.txt -pass pass:treballfinalgrau
```

En la següent imatge, podem observar el contingut del document resultant anomenat Desxifrar_ArxiuTFG-XifratAES, aquest contingut es idèntic al contingut del document ArxiuTFG i en conseqüència el xifratge i desxifratge s'ha produït correctament .

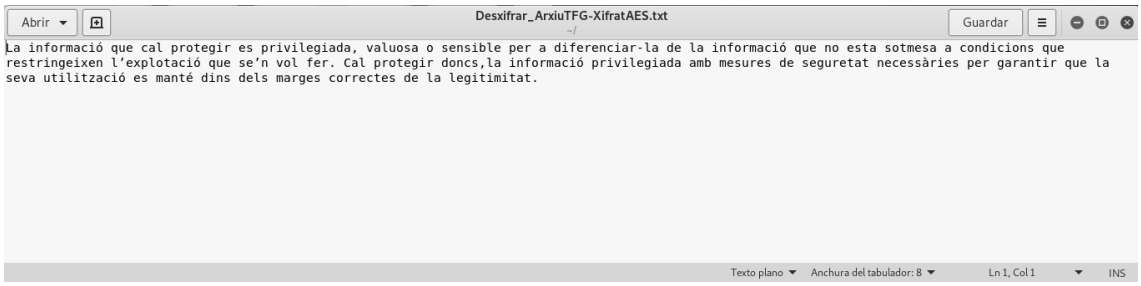
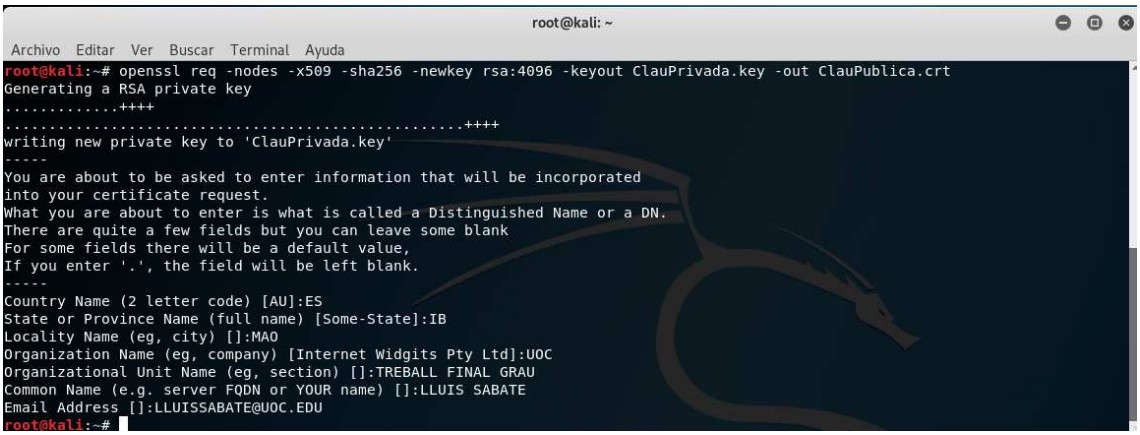


Figura 7.5: Desxifrar_ArxiuTFG-XifratAES.txt

Per poder realitzar xifratge amb RSA, abans he de generar les claus, tant la pública com privada. A continuació es pot visualitzar, com genero aquestes claus amb una grandària de 4096 bits i empleno unes dades que es demanen.



Una vegada generades les claus, aquestes son les que apareixen en la següent imatge.

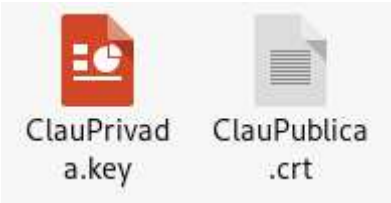


Figura 7.6: Clau Privada i Pública

El següent pas, es realitzar el xifratge del document ArxiuTFG amb el certificat que he generat anteriorment, ClauPublica.crt, el contingut es gravarà en un nou document anomenat ArxiuTFG-Xifrat.

```
root@kali:~# openssl rsautl -encrypt -inkey <(openssl x509 -in ClauPublica.crt -pubkey -noout) -pubin -in ArxiuTFG.txt -out
ArxiuTFG-Xifrat.txt.enc
root@kali:~#
```

En la següent imatge, es pot observar el contingut del document ArxiuTFG-Xifrat.

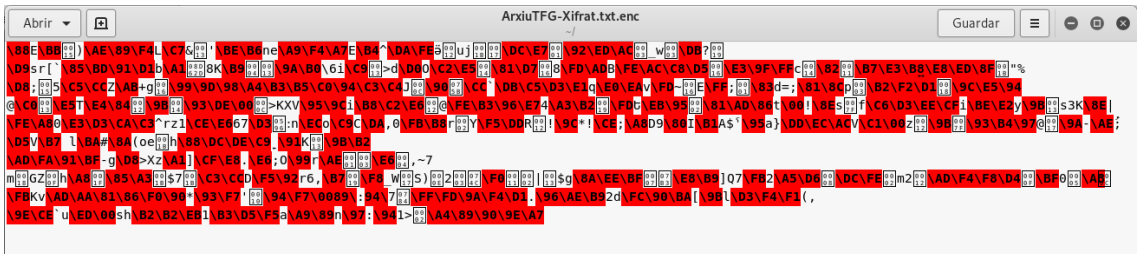


Figura 7.7: ArxiuTFG-Xifrat

Ara procedeixo a desxifrar el document ArxiuTFG-Xifrat amb la clau privada, ClauPrivada.key, que he generat anteriorment. A continuació es visualitza la comanda que he utilitzat per realitzar aquesta acció. El document resultant es gravarà amb el nom Desxifrar_ArxiuTFG-Xifrat.

```
root@kali:~# openssl rsautl -decrypt -inkey ClauPrivada.key -in ArxiuTFG-Xifrat.txt.enc -
out Desxifrar_ArxiuTFG-Xifrat.txt
root@kali:~#
```

A continuació es pot observar el contingut de l'arxiu Desxifrar_ArxiuTFG-Xifrat.

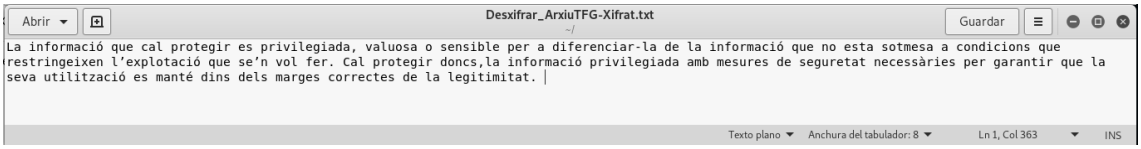


Figura 7.8: Desxifrar_ArxiuTFG-Xifrat

Per signar un document, s'ha d'utilitzar la clau privada. Acontinuació, es pot observar la comanda que utilitzo per realitzar signatura del document anomenat ArxiuTFG amb la clau privada que he generat anteriorment, ClauPrivada.key i llavors es genera un nou document anomenat Signatura-ArxiuTFG.

```
root@kali:~#
root@kali:~# openssl dgst -sha256 -sign ClauPrivada.key -out Signatura-ArxiuTFG.signature ArxiuTFG.txt
root@kali:~#
```

En la següent imatge, es pot observar el contingut del document Signatura-ArxiuTFG.

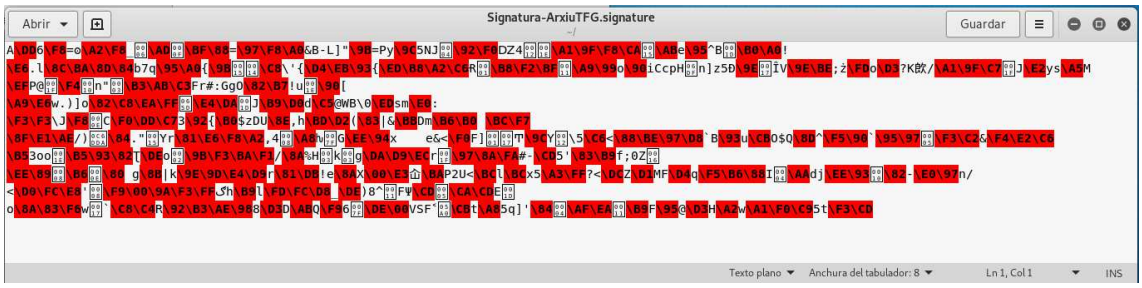


Figura 7.9: Signatura-ArxiuTFG

Per verificar la signatura d'un document, s'ha d'utilitzar la clau publica. A continuació, es pot observar per una banda, la comanda que utilitzo per realitzar la verificació de la signatura del document anomenat Signatura-ArxiuTFG amb la clau publica que he generat anteriorment, ClauPublica.crt. I per altre banda, la resposta que obtinc de la comada que he generat , i aquesta es " Verified OK", per tant puc afirmar amb total seguretat que la clau Privada, ClauPrivada.key, es parella de la clau publica, ClauPublica.crt i en conseqüència verifico l'autenticitat i l'integritat de l'arxiu.

```
root@kali:~#
root@kali:~# openssl dgst -sha256 -verify <(openssl x509 -in ClauPublica.crt -pubkey -noout) -signature Signatura-ArxiuTFG.signature ArxiuTFG.txt
Verified OK
root@kali:~#
```

Si volgués intentar verificar aquesta mateixa signatura amb un altre certificat, la resposta seria que la clau Privada i la clau publica utilitzades no serien parells i per tant no es por verificar, en conseqüència es pot dir que no hi ha autenticitat i per tant hi hagut suplantació d'identitat.

A continuació es pot corroborar aquesta afirmació. Realitzo una nova generació de claus, amb una grandària de 4096 bits, la clau privada serà Estudiant.key i la clau publica serà Estudiant.crt

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:~# openssl req -nodes -x509 -sha256 -newkey rsa:4096 -keyout Estudiant.key
-out Estudiant.crt
Generating a RSA private key
.................................++++
.......................................................................................
.................................++++
writing new private key to 'Estudiant.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:CAT
Locality Name (eg, city) []:BCN
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UOC
Organizational Unit Name (eg, section) []:TFG
Common Name (e.g. server FQDN or YOUR name) []:ESTUDIANT GENERIC
Email Address []:ESTUDIANT@UOC.EDU
root@kali:~#
```

Per concloure, realitzo verificació de signatura del document Signatura-ArxiuTFG amb la clau publica, Estudiant.crt i el document original ArxiuTFG. La resposta de la comanda es “Verification Failure”, en conseqüència la verificació ha fallat, ja que ClauPrivada.key i Estudiant.crt no son parells, es a dir, no s’han generat en el mateix moment.

```
root@kali:~# openssl dgst -sha256 -verify <(openssl x509 -in Estudiant.crt -pubkey -noout) -signature Signatura-ArxiuTFG.signature ArxiuTFG.txt
Verification Failure
root@kali:~#
```

8. Conclusions

La realització d'aquest projecte, ve donat per la problemàtica que genera la informació avui dia, i per tant tenir com a objectiu prioritari aconseguir la seguretat d'aquesta mitjançant la criptografia.

Fent un resum dels inicis de la criptografia, trobem la criptografia clàssica que implementa algorismes que realitzen transposició i substitució dels caràcters del text en clar. Aquests sistemes de xifratge son senzills, pocs segurs i gens fiables on resulta fàcil aplicar mètodes de criptoanàlisi per rompre el xifratge.

La criptografia moderna es diferencia clarament de la clàssica, partint de la premissa que l'únic que s'ha de protegir és la clau d'un criptosistema i no el seu algorisme.

En la realització d'aquest projecte, he seleccionat els tres algorismes que actualment son els més utilitzats en la criptografia moderna i els he estudiat detingudament. L' implementació dels algorismes simètrics es basa en:

- AES: fa ús de les voltes per aplicar funcions que realitzen operacions de transposició, substitució i de tipus XOR. Aconseguint que els bits del text en clar es barregin amb els bits de la clau, amb un mínim de 11 voltes. Cadascuna d'aquestes voltes genera, a partir de la funció expansió de clau, una nova clau, anomenada subclau.

- CHACHA20: abans de xifrar el text en clar, s'ha d'obtenir la seqüència de claus, a partir d'aplicar voltes i quarts de volta a la clau inicial, utilitzant operacions de transposició, substitució i de tipus XOR. Una vegada obtinguda la seqüència de claus, es podrà aplicar una operació XOR entre aquesta i el text en clar per obtenir el criptograma.

L' implementació de l'algorisme asimètric es basa en:

- RSA: a partir de la selecció de nombres primers molt grans, genera les claus pública (k, n) i privada (j). El criptosistema de xifrat utilitza la clau publica (k) com a exponent del text en clar amb l'operació mod de la clau pública (n). I el criptosistema de desxifrat utilitza la clau privada (j) com a exponent del criptograma amb l'operació mod de la clau pública (n).

Finalment, puc concloure que:

- Els algorismes simètrics utilitzen una sola clau, mentre que els asimètrics utilitzen dues, una pública i una privada.

- Els algorismes simètrics necessiten de l'enviament de la clau per cada sessió, en canvi pels asimètrics no es necessari.

- El xifratge simètric és més ràpid però menys segur que el xifratge asimètric.

- La criptografia simètrica utilitza algorismes que implementen voltes i operacions XOR, mentre que, l'asimètrica utilitza exponenciació modular.

La criptografia és un camp que ha d'evolucionar constantment, per seguir sent eficient, amb l'ajuda de la tecnologia. Sembla que el seu futur passa per l'anomenada criptografia quàntica, que es regeix principalment per les lleis físiques i no tant per les matemàtiques.

La criptografia quàntica fa us de partícules lluminoses, fotons, i les seves característiques pròpies, per desenvolupar criptosistemes segurs, ja que és impossible mesurar l'estat quàntic de qualsevol criptosistema sense pertorbar-lo.

9. Annex

9.1. Operació Mod

S'anomena també operació mòdul, té com a finalitat obtenir el residu d'una divisió entre dos nombres. Per exemple, l'expressió 7 mod 2 té com a resultat 1.

9.2. Operació XOR

Es una operació lògica, ja que XOR es un operador lògic. Al utilitzar sistema binari, es a dir,uns i zeros, consisteix a realitzar comparació dels bits de la mateixa posició de 2 nombres, si els 2 nombres son iguals, el bit resultant serà 0; en canvi si els 2 nombres son diferents, el bit resultant serà igual a 1.

9.3. Funció de Resum o hash

Consisteix en acceptar com a entrada un missatge de longitud variable, missatge m, i al aplicar un algorisme hash, es retorna una cadena de bits de longitud fixa. Aquesta cadena s'anomena hash i té com a tret principal que el seu valor és únic. Existeixen dos tipus de funcions de resum:

- MDC (Modification Detection Codes), aquest tipus d'algorisme es calcula directament sobre el missatge, amb l'objectiu de garantir la seva integritat. En l'actualitat trobem algorismes hash MDC com SHA-2, SHA-3, entre altres.

- MAC, (Message Authentication Codes), aquest tipus d'algorisme calcula directament sobre el missatge i es xifra amb una clau, per garantir no només la seva integritat sinó que es garanteix l'origen del missatge i per tant la seva autenticitat. En l'actualitat trobem algorismes hash MAC com Poly1305.

10. Referències

UOC. Política de seguridad criptográfica de la Universidad Oberta de Catalunya [en línia]. https://www.uoc.edu/portal/_resources/ES/documents/seu-electronica/Politica_Seguretat_Criptogrxfica_UOC-cat_ES.pdf

UPM. Class4crypt: Introducción Videoclases de criptografia aplicada [en línia]. http://www.criptored.upm.es/descarga/Class4cryptc4c1.1_Ciberseguridad_y_criptografia.pdf

Youtube. Class4crypt:Canal videoclases de criptografia aplicada [en línia]. https://www.youtube.com/channel/UC5t0X0VGOT2P8sj7gXPeHIg

UPM. Proyecto CLCript: Cuadernos de laboratio de criptografia [en línia]. http://www.criptored.upm.es/software/sw_m001s.htm

UPM. Interpedia:Information Security Encyclopedia [en línia]. http://www.criptored.upm.es/intypedia/index.php?lang=es

UPM. Proyecto Thoth [en línia]. http://www.criptored.upm.es/thoth/index.php

UNAM. Criptografia [en línia]. https://criptografia.webnode.es/

Binance Academy. ¿Qué es la criptografia de clave pública? [en línia]. https://academy.binance.com/es/articles/what-is-public-key-cryptography

Mediacloud. Para qué sirve la criptografía de clave pública y privada [en línia]. https://blog.mdcloud.es/para-que-sirve-la-criptografia-de-clave-publica-y-privada/

Grupo Atico. Qué es la criptografía asimétricra y cómo funciona [en línia]. https://protecciondatos-lopd.com/empresas/criptografia-asimetrica/

MDN Web Docs. Algoritmo criptográfico [en línia]. https://developer.mozilla.org/es/docs/Glossary/Cipher#saber_m%C3%A1s

CiberSeguridad. ¿Qué son los algoritmos de cifrado? Tipos y caracteristicas [en línia]. https://ciberseguridad.com/servicios/algoritmos-cifrado/#%C2%BFPara_que_sirve_la_criptografia_o_cifrado_de_datos

HiperTextual. La màquina Enigma, el sistema de cifrado que puso en jaque a Europa [en línia]. https://hipertextual.com/2011/07/la-maquina-enigma-el-sistema-de-cifrado-que-puso-en-jaque-a-europa

Eldiariopuntoes. Codigo Enigma, descifrado: el papel de Turing en la Segunda Guerra Mundial [en línia]. https://www.eldiario.es/turing/criptografia/alan-turing-enigma-codigo_1_5038272.html

Redes Zone. Todo sobre criptografía:Algoritmos de clave simétrica y asimétrica [en línia]. https://www.redeszone.net/tutoriales/seguridad/criptografia-algoritmos-clave-simetrica-asimetrica/

YouTube. Algoritmos de cifrado AES y DES [en línia]. https://www.youtube.com/watch?app=desktop&v=BctlBJ2NdHk

Error509. Vpn IPSEC site-to-site básico [en línia]. https://www.error509.com/2019/09/vpn-ipsec-site-to-site-basico/

WatchGuard. Acerca de las negociaciones VPN de IPSec [en línia]. https://www.watchguard.com/help/docs/help-center/es-419/Content/es-419/fireware/mvpn/general/ipsec_vpn_negotiations_c.html

Itesa. Packet Tracer: Configuración de VPN [en línia]. https://www.itesa.edu.mx/netacad/networks/course/files/7.1.2.4%20Packet%20Tracer%20-%20Configuring%20VPNs%20(Optional)%20Instructions.pdf

WatchGuard. Terminología VPN para Sucursales [en línia]. https://www.watchguard.com/help/docs/help-center/es-419/Content/es-419/fireware/bovpn/manual/bovpn_terms_c.html

Wikipedia. Salsa20 [en línia]. https://en.wikipedia.org/wiki/Salsa20#ChaCha20_adoption

Asecurity Site. AES is a great but we need a fall-back:meet chacha and poly1305 [en línia]. https://en.wikipedia.org/wiki/Salsa20#ChaCha20_adoption

Asecurity Site. Salsa20 and Chacha20 stream chipers [en línia]. https://asecuritysite.com/encryption/salsa20

Check point Software technologies. The Chacha20 Encryption Algoritm [en línia]. https://datatracker.ietf.org/doc/html/rfc7539#section-2.4

Segu-info. Protocols chacha20 y Poli1305 para reforzar conexiones HTTPS. https://blog.segu-info.com.ar/2014/04/protocolos-chacha20-y-poly1305-para.html

Junco TIC. RSA: ¿Cómo funciona este algoritmo de cifrado? [en línia]. https://juncotic.com/rsa-como-funciona-este-algoritmo/

Fluproject. Generación de claves RSA y selección de los números primos [en línia]. https://www.flu-project.com/2019/10/generacion-claves-rsa-Parte-2.html

Pleets Blog. Criptografía simétrica en OpenSSL-Algoritmos de cifrado. https://blog.pleets.org/article/intro-y-algoritmos-openssl

LN. Criptografía: Que son los algoritmos hash y para que se utilizan [en línia]. https://leonardonetwork.es/blog/criptografia-que-son-los-algoritmos-hash-y-para-que-se-utilizan/

LN. Todo sobre criptografía: Algoritmos de clave simétrica y asimétrica. https://leonardonetwork.es/blog/todo-sobre-criptografia-algoritmos-de-clave-simetrica-y-asimetrica/

Criptografia para principiantes. José de Jesús Angel Angel. Creative Commons, Attribution Noncomercial No derivate works. 2002.

Libro Electrónico de Seguridad Informática y Criptografía. Jorge Ramió Aguirre. Creative Commons, Attribution Noncomercial No derivate works. 2006.

Curso de Criptografía Aplicada. Jorge Ramió Aguirre. Creative Commons, Attribution Noncomercial No derivate works. 2020.