UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

Treball final del grau de Matemàtiques

Estudi algebraic de l'AES i d'un atac teòric

David Alegre Alarza

Director: Dr. Artur Travesa i Grau
Barcelona, 13 de juny de 2022

Abstract

Since the birth of the AES, ways have been sought to try to break it. It seems that the most common cryptoanalytic methods do not succeed. For this reason the XSL is invented, an algebraic attack against the AES. The goal of this thesis is to study the algebra of this attack.

Resum

Des del naixement de l'AES, s'han buscat maneres d'intentar trencar-lo. Sembla que els mètodes criptoanalítics més comuns no ho aconsegueixen. És per això que s'inventa l'XSL, un atac algebraic contra l'AES. L'objectiu d'aquest treball és estudiar algebraicament aquest atac.

Agraïments

Vull donar les gràcies a diverses persones:

En primer lloc al Dr. Artur Travesa, per la seva gran ajuda.

En segon lloc a la meva família, per la seva comprensió i ànims.

2020 Mathematics Subject Classification: 11T06, 11T71, 13M10, 13P15, 68W40.

Contingut

Capítol 1
Introducció

En la criptografia moderna es consideren dos grans grups de criptosistemes, els de clau pública o asimètrica i els de clau privada o simètrica. En els criptosistemes de clau pública la clau hi és dividida en dues parts. Una de les parts només la coneix el destinatari del missatge, és a dir a qui va dirigit el missatge. L'altra part és del coneixement de tothom i és utilitzada per xifrar el missatge que més tard desxifrarà el destinatari amb la part privada de la clau. En els criptosistemes simètrics hi ha una clau, i només la coneixen l'emissor i el destinatari.

Una altra classificació dels criptosistemes és expressada per la quantitat d'unitats de missatge que es xifren. Un d'aquests grups és el dels xifratges en bloc. En aquests criptosistemes el missatge es descompon en blocs d'una longitud determinada pel criptosistema i cada bloc es xifra independentment dels altres. Un exemple d'aquests xifratges en bloc és l'AES.

L'AES neix a partir de la iniciativa de l'US National Institute of Standards and Technology (NIST) per a reemplaçar el que fins aquell moment era l'estàndard d'encriptació, el DES. Al gener del 1997, el NIST va anunciar que el procés de selecció per a l'AES seria obert, és a dir, que qualsevol podria presentar un criptosistema. Un dels criptosistemes presentats va ser el Rijndael, proposat per Joan Daemen i Vincent Rijmen. A l'octubre de l'any 2000 el NIST va anunciar que el guanyador del concurs era el Rijndael, que és un xifratge en bloc de clau simètrica.

Des del naixement de l'AES els criptoalanistes han buscat maneres de trencar el sistema. Pel seu disseny, l'AES sembla resistent als algoritmes criptoanalítics tradicionals, fent que s'hagin de buscar algoritmes nous.

Claude E. Shannon en el seu article [Sh-49] de l'any 1949 va plantejar la pregunta següent: Com podem estar segurs que un sistema que no és ideal, per tant, ha de tenir una solució única, necessitarà una gran quantitat de treball per a trencar-lo amb cada mètode d'anàlisi? Ell dona dues maneres de dirigir-se al problema. Una manera és estudiar els mètodes criptoanalítics i dissenyar el sistema perquè no sigui possible utilitzar-los. Una altra manera és construir el criptosistema de manera que trencar-lo sigui equivalent a resoldre un problema que se sàpiga que porta molt treball. Així que si podem veure que trencar un criptosistema necessita com a molt tant treball com resoldre un sistema d'equacions amb un nombre prou gran d'incògnites de tipus complex, aleshores podríem obtenir una fita inferior per al treball que fa falta per a poder trencar el sistema.

Com a resposta a aquesta pregunta, Nicolas Courtois i Josef Pieprzyk van dissenyar l'algoritme XSL. Aquest algoritme és un atac algebraic que es basa en la resolució d'un sistema d'equacions en diverses incògnites.

Aquesta memòria consta de tres parts. En la primera part s'estudia l'AES algebraicament. En la segona part es descriuen els dos atacs XSL i s'estudia la matemàtica que hi ha darrere de cada pas de l'algoritme. En la tercera i última part s'exposen les conclusions obtingudes després de l'estudi dels dos atacs XSL.

Capítol 2
Advanced Encryption Standard

L'única diferència entre l'AES i el Rijndael és en el nombre de bits de cada bloc i de la clau. En el Rijndael varia tant la longitud del bloc com la de la clau i poden ser especificades independentment com a qualsevol múltiple de 32 bits, amb un mínim de 128 bits i un màxim de 256 bits. L'AES, en canvi, fixa la longitud dels blocs en 128 bits i el que varia és la longitud de la clau, que només pot ser de 128, 192 o 256 bits. Això és perquè en el procés de selecció del sistema criptogràfic només es va testar la seguretat de les composicions que utilitza l'AES.

En aquest capítol s'especifica l'algoritme Rijndael, xifratge i desxifratge, i s'estudia l'àlgebra que hi ha darrere de cada pas. Per a poder parlar còmodament i amb profunditat de l'algoritme convé veure la seva base matemàtica.

2.1 L'àritmètica bàsica del criptosistema

Anomenem $\mathbb{F}_2$ al cos de dos elements, i sigui $\mathbb{F}_2[x]$ l'anell de polinomis en una indeterminada sobre $\mathbb{F}_2$. Per a qualsevol polinomi no constant $m(x)$, podem considerar l'anell quocient $\mathcal{B} = \dfrac{\mathbb{F}_2[x]}{(m(x))}$. Aquest anell és finit i és un espai vectorial de dimensió n sobre $\mathbb{F}_2$, on n és el grau del polinomi $m(x)$. Els seus elements són les classes dels polinomis de grau com a molt $n-1$ i de coeficients en $\mathbb{F}_2$. En particular les classes de x^{n-1}, x^{n-2},..., x, 1 formen una $\mathbb{F}_2$-base. Per al cas de l'AES, ens fixarem en el cas $n = 8$.

Siguin $a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ i $b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ dues classes qualssevol de polinomis. La suma és

$$a(x) + b(x) = \sum_{i=0}^{7} (a_i + b_i)x^i,$$

on $(a_i + b_i)$ és la suma a $\mathbb{F}_2$.

Tenim que qualsevol classe de representant $a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0$ la podem representar com $(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ que alhora es pot representar com a bytes de la manera següent, $a_7a_6a_5a_4a_3a_2a_1a_0$. Aquesta última representació ens permet

veure que la suma en $\mathcal{B}$ és equivalent a l'operació informàtica XOR.

Siguin $a(x) = a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0$ i $b(x) = b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x+b_0$ dues classes qualssevol de polinomis. Aleshores el producte és $a(x)\cdot b(x)$ en $\mathbb{F}_2[x]$ mòdul el polinomi $m(x)$. Aquesta operació depèn de com s'ha descrit $\mathcal{B}$, ja que un polinomi representa una classe diferent segons quin sigui el polinomi $m(x)$. Per exemple, siguin $a(x) = x^7+1$ i $b(x) = x^2+x$. Aleshores $a(x)\cdot b(x) = x^9+x^8+x^2+x$ en $\mathbb{F}_2[x]$. Si definim $\mathcal{B}$ amb $m(x) = x^8+x^4+x^3+x+1$, aleshores tenim que

$$a(x)\cdot b(x) = x^5+x^3+x+1.$$

Si definim $\mathcal{B}$ amb $m(x) = x^8+x^4+x^3+x^2+1$ tenim que

$$a(x)\cdot b(x) = x^5+1.$$

Proposició 2.1.1. *Els polinomis irreductibles de* $\mathbb{F}_2[x]$ *són*

$1+x+x^3+x^4+x^8$,	$1+x^3+x^5+x^6+x^8$,	$1+x^4+x^5+x^7+x^8$,
$1+x^2+x^3+x^4+x^8$,	$1+x^4+x^5+x^6+x^8$,	$1+x^2+x^3+x^4+x^5+x^7+x^8$,
$1+x+x^3+x^5+x^8$,	$1+x+x^2+x^4+x^5+x^6+x^8$,	$1+x+x^6+x^7+x^8$,
$1+x^2+x^3+x^5+x^8$,	$1+x+x^3+x^4+x^5+x^6+x^8$,	$1+x+x^2+x^3+x^6+x^7+x^8$,
$1+x^3+x^4+x^5+x^8$,	$1+x+x^2+x^7+x^8$,	$1+x+x^2+x^4+x^6+x^7+x^8$,
$1+x+x^2+x^3+x^4+x^5+x^8$,	$1+x+x^3+x^7+x^8$,	$1+x^2+x^3+x^4+x^6+x^7+x^8$,
$1+x^2+x^3+x^6+x^8$,	$1+x^2+x^3+x^7+x^8$,	$1+x+x^2+x^5+x^6+x^7+x^8$,
$1+x+x^2+x^3+x^4+x^6+x^8$,	$1+x+x^2+x^3+x^4+x^7+x^8$,	$1+x+x^4+x^5+x^6+x^7+x^8$,
$1+x+x^5+x^6+x^8$,	$1+x+x^5+x^7+x^8$,	$1+x^2+x^4+x^5+x^6+x^7+x^8$,
$1+x^2+x^5+x^6+x^8$,	$1+x^3+x^5+x^7+x^8$,	$1+x^3+x^4+x^5+x^6+x^7+x^8$.

Demostració. Sabem que $x^{2^8}-x$ és el producte de $x^{2^4}-x$ per tots els polinomis irreductibles mònics de grau 8 sobre $\mathbb{F}_2$. Aleshores només cal comprovar que $\dfrac{x^{2^8}-x}{x^{2^4}-x}$ és el producte dels polinomis donats i tenir en compte que $\mathbb{F}_2$ és de factorització única. □

Per al cas de l'AES, $\mathcal{B}$ es defineix amb el polinomi $m(x) = x^8+x^4+x^3+x+1$. Com que $m(x)$ és irreductible, aleshores $\mathcal{B}$ és un cos.

A partir de $\mathcal{B}$, definim l'anell $\mathcal{A} = \dfrac{\mathcal{B}[x]}{(x^4+1)}$. Aquest anell és un espai vectorial de grau 4 sobre $\mathcal{B}$. Fixem-nos que en aquest cas $x^4+1 = (x+1)^4$. Els elements de $\mathcal{A}$ són les classes de polinomis de grau com a molt 3 i de coeficients en $\mathcal{B}$.

Siguin $a(x) = a_3x^3+a_2x^2+a_1x+a_0$ i $b(x) = b_3x^3+b_2x^2+b_1x+b_0$ dues classes qualssevol de $\mathcal{A}$. Aleshores la suma és

$$a(x)+b(x) = \sum_{i=0}^{3}(a_i+b_i)x^i,$$

on a_i+b_i és la suma en $\mathcal{B}$.

Siguin $a(x) = a_3x^3+a_2x^2+a_1x+a_0$ i $b(x) = b_3x^3+b_2x^2+b_1x+b_0$ dues classes qualssevol de $\mathcal{A}$. Aleshores el producte en $\mathcal{A}$ és el producte $a(x)\cdot b(x)$ en $\mathcal{B}[x]$ mòdul x^4+1. Com

que el polinomi x^4+1 no és irreductible, aleshores no tots els elements de $\mathcal{A}$ tenen invers per al producte.

Proposició 2.1.2. *L'anell quocient $\mathcal{A}$ és un $\mathcal{B}$-mòdul lliure. Una base ordenada és*

$$\{x^3, x^2, x, 1\}.$$

□

Per a poder descriure l'AES, convé definir unes funcions. Definim $g : \mathcal{B} \to \mathcal{B}$ tal que $g(0) = 0$ i $g(a(x)) = a^{-1}(x)$, $\forall a(x) \in \mathcal{B}$, $a(x) \neq 0$.

Proposició 2.1.3. *La funció g és ben definida, bijectiva i la seva inversa és ella mateixa.*

□

Definim $f : \mathcal{B} \to \mathcal{B}$ tal que $\forall a(x) \in \mathcal{B}$, representant $a(x)$ com a element en la base ordenada $\{x^7, x^6, x^5, x^4, x^3, x^2, x, 1\}$, li fa correspondre un element $b(x) \in \mathcal{B}$ tal que

$$\begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

□

Proposició 2.1.4. *L'aplicació afí f és invertible i la seva inversa és*

$$\begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

□

Proposició 2.1.5. *$f \circ g$ no és lineal.*

Demostració. Considerem la classe de representant $x+1$. Aleshores volem veure que

$$f(g((x+1)+(x+1))) \neq f(g(x+1)) + f(g(x+1)).$$

Calculem $(x+1)+(x+1) = 0$, així que $g(0) = 0$ i $f(0)$ és la classe de representant x^6+x^5+x+1. Com la inversa de la suma en $\mathcal{B}$ és ella mateixa, aleshores tenim que $f(g(x+1)) + f(g(x+1)) = 0$. Per tant, hem vist que la composició no és lineal. □

Definim $p : \mathcal{A} \to \mathcal{A}$ tal que $\forall a(x) \in \mathcal{A}$, li fa correspondre un element $b(x) \in \mathcal{A}$ tal que

$$b(x) = a(x) \cdot c(x),$$

on $c(x) = 00000011 \cdot x^3 + 00000001 \cdot x^2 + 00000001 \cdot x + 00000010$ i el producte és el d'$\mathcal{A}$. Com que $\mathcal{A}$ és una $\mathcal{B}$-àlgebra, aleshores el producte és $\mathcal{B}$-lineal.

Proposició 2.1.6. *La funció p es pot descriure matricialment com*

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 00000010 & 00000011 & 00000001 & 00000001 \\ 00000001 & 00000010 & 00000011 & 00000001 \\ 00000001 & 00000001 & 00000010 & 00000011 \\ 00000011 & 00000001 & 00000001 & 00000010 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}.$$

□

Com que el producte es pot descriure com a una funció $\mathcal{B}$-lineal, i tenim que $c(x)$ és invertible en $\mathcal{A}$ amb invers

$$d(x) = 00001011 \cdot x^3 + 00001101 \cdot x^2 + 00001001 \cdot x + 00001110,$$

resulta que

Proposició 2.1.7. *la funció inversa de p es pot descriure matricialment com*

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 00001110 & 00001011 & 00001101 & 00001001 \\ 00001001 & 00001110 & 00001011 & 00001101 \\ 00001101 & 00001001 & 00001110 & 00001011 \\ 00001011 & 00001101 & 00001001 & 00001110 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}.$$

□

Definim la permutació τ tal que $(a_1, ..., a_{16}) = (a_{\tau(1)}, ..., a_{\tau(16)})$, on τ és la permutació

$$\tau = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 1 & 14 & 11 & 8 & 5 & 2 & 15 & 12 & 9 & 6 & 3 & 16 & 13 & 10 & 7 & 4 \end{pmatrix}$$

i $a_i \in \mathcal{B}$, per a $i = 1, ..., 16$.

2.2 Algoritme Rijndael

L'algoritme necessita dues entrades per a poder-se dur a terme. Necessita un text per xifrar, que anomenarem text pla. El text pla és format per bits, i ha de tenir almenys 128 bits, ja que és el necessari per a poder aplicar l'algoritme. Si en té més, fa falta que aquest nombre sigui un múltiple de 128. El més segur és que al passar el missatge que volem xifrar a bits no tinguem aquest nombre de bits desitjat. Per tant, s'han de buscar altres maneres de complementar el missatge original sense perdre informació. Una manera podria ser afegir unitats de missatge amb un cert patró per tal de completar el text. Aquest mètode té un problema. Aquest patró podria facilitar a qualsevol persona que interceptés el missatge trobar la clau. Una manera per a solucionar aquest punt podria ser afegir uns bits al final que indiquin la llargària del missatge original. Entremig, encara ens poden quedar bits per completar o no. Una manera per omplir aquest espai és completar amb

bits aleatoris. Si tenim el cas en què l'arxiu ja fos múltiple del nombre de bits per bloc, podem aplicar el mateix procediment per afegir-li una mica més de seguretat, fent que sigui més difícil descobrir un patró de xifratge. Si en afegir la longitud del text ens dona un múltiple del nombre de bytes per bloc, aleshores afegim un bloc sencer aleatori per la mateixa raó que en el cas anterior, per afegir un grau més de seguretat. Ara ja es té el text pla.

L'altra dada necessària per a poder xifrar és la clau. Mentre que l'AES fixa la longitud per bloc, no passa el mateix amb la longitud de la clau. Aquesta varia entre 128, 192 o 256 bits. Anomenem N_k al nombre de bytes entre 4. Per exemple, si $N_k = 4$, aleshores hi ha 16 bytes, que són 128 bits. Per a l'AES-192 $N_k = 6$ i per a l'AES-256, $N_k = 8$.

Claude E. Shannon en [Sh-49] explica que hi ha dos mètodes per a intentar frustrar un atac estadístic sobre un criptosistema, la confusió i la difusió. La confusió fa la relació entre la clau i el criptosistema complexa. La difusió fa que l'estructura estadística del missatge xifrat, que porta a redundàncies, no sigui tan evident, és a dir, dissipa les redundàncies del missatge. Amb aquest mètode qui intercepti el missatge haurà d'interceptar una quantitat considerable de missatges diferents per a poder trobar aquesta estructura. L'AES utilitza els dos mètodes.

2.2.1 Expansió de la clau

L'objectiu d'aquesta secció és explicar com es prepara la clau per a poder xifrar el text pla.

L'algoritme diferencia entre $N_k \leq 6$ i $N_k > 6$. Sigui $i = 1, 2, 3, 4$ i sigui $j = 0, ..., 43$. El valor j compta el nombre de grups de 4 bytes i el valor i recorre aquests bytes. L'algoritme genera els valors K_{i+4j} de la clau. El primer pas és escriure en les primeres posicions la clau donada. En el cas $N_k = 4$, j prendria els valors des de 4 fins a 43. L'algoritme és el següent:

1. Si j és múltiple de N_k o si $N_k > 6$ i $j \equiv 4 \pmod{N_k}$,
   (a) Si $i = 1$,
   $$k_{i+4j} = k_{1+4(j-N_k)} + f(g(k_{2+4(j-1)})) + R_{\frac{j}{N_k}} \in \mathcal{B},$$
   on les funcions f i g són les descrites en la secció anterior i $R_l = x^{l-1} \in \mathcal{B}$. La constant s'afegeix per a treure simetries i la caixa S per a la part de confusió.
   (b) Si $i \neq 1$,
   $$k_{i+4j} = k_{i+4(j-N_k)} + f(g(k_{(i \ (mod \ N_k))+1+4(j-1)})) \in \mathcal{B}.$$
2. Si j no és múltiple de N_k,
   $$k_{i+4j} = k_{i+4(j-N_k)} + k_{i+4(j-1)}.$$

2.2.2 Xifratge i desxifratge

Anomenem $p_1p_2p_3p_4p_5p_6p_7p_8p_9p_{10}p_{11}p_{12}p_{13}p_{14}p_{15}p_{16}$ els elements d'un bloc de xifratge, on p_i és un byte. Un cop tenim la clau expandida, tindrem les anomenades claus de ronda, que són segments de la clau expandida. Aquestes claus les anomenarem $k_{j,i}$, on $j = 0, ..., 10$ indica la ronda on s'utilitza aquell segment i $i = 1, ..., 16$ indica la posició dins

la clau del byte.

El primer pas de l'algoritme és sumar la clau de ronda 0, així que tindrem $q_{1,i} = p_i + k_{0,i}$, per a $i = 1, ..., 16$. Aquesta suma és la suma en $\mathcal{B}$, amb els elements expressats en la base ordenada $\{x^7, x^6, x^5, x^4, x^3, x^2, x, 1\}$. Ara tindrem el bloc de la forma

$$q_{1,1}q_{1,2}q_{1,3}q_{1,4}q_{1,5}q_{1,6}q_{1,7}q_{1,8}q_{1,9}q_{1,10}q_{1,11}q_{1,12}q_{1,13}q_{1,14}q_{1,15}q_{1,16}.$$

Els següents passos s'apliquen un total de 9 vegades en aquest ordre. A aquesta sèrie de passos se'ls anomena ronda. El primer pas és aplicar la composició $(f \circ g)(q_{j,i})$ a tots els bytes $q_{j,i}$ del bloc. Cada byte pot ser representat a partir de dues paraules hexadecimals, on els 4 primers bits del byte formen una paraula i els 4 restants formen l'altra paraula. Per exemple el byte 00010001 forma la paraula 11. Amb aquesta notació es pot construir una taula per a fer més eficient la implementació d'aquesta composició. Per aquest motiu aquesta operació és anomenada caixa S. En l'apèndix A hi és aquesta caixa tabulada. Aquest pas correspon a la part de confusió de l'algoritme.

A continuació s'aplica la permutació τ a tot el bloc, donant lloc al bloc $r_{j,i}$. Tot seguit s'aplica l'operació p de la manera següent. Cada quatre bytes $r_{j,i}$ formen un element de $\mathcal{A}$, ja que cadascun dels grups forma un element de $\mathcal{B}$, així que es fan les operacions $p(r_{j,1}r_{j,2}r_{j,3}r_{j,4}) = s_{j,1}s_{j,2}s_{j,3}s_{j,4}$, $p(r_{j,5}r_{j,6}r_{j,7}r_{j,8}) = s_{j,5}s_{j,6}s_{j,7}s_{j,8}$, $p(r_{j,9}r_{j,10}r_{j,11}r_{j,12}) = s_{j,9}s_{j,10}s_{j,11}s_{j,12}$ i $p(r_{j,13}r_{j,14}r_{j,15}r_{j,16}) = s_{j,13}s_{j,14}s_{j,15}s_{j,16}$, donant lloc al bloc

$$s_{j,1}s_{j,2}s_{j,3}s_{j,4}s_{j,5}s_{j,6}s_{j,7}s_{j,8}s_{j,9}s_{j,10}s_{j,11}s_{j,12}s_{j,13}s_{j,14}s_{j,15}s_{j,16}.$$

Finalment, se suma la clau de ronda $k_{j,i}$, $i = 1, ..., 16$ que correspon. Aquestes tres últimes operacions corresponen a la part de difusió.

Un cop ja ha fet les 9 rondes, executa una última ronda però sense utilitzar la funció p.

A l'hora de desxifrar hi ha dues maneres. La primera manera és la més intuïtiva, que correspon a fer el procés invers utilitzant les funcions inverses que hem calculat anteriorment. L'altra manera de desxifrar usa la linealitat d'algunes de les funcions. El primer que veiem és que τ' és una permutació i $g \circ f^{-1}$ actua sobre cada grup sense importar el seu ordre, així que aquestes dues poden permutar. A més a més, la funció p^{-1} és lineal i estem aplicant aquesta funció a una suma, així que és equivalent a aplicar p^{-1} a cadascun dels sumands i després sumar, permetent-nos intercanviar l'ordre de la suma i p^{-1}. A partir d'aquesta última observació obtenim que ens farà falta una altra clau, que és la clau equivalent. Aquesta clau és la clau expandida, però on a cada bloc s'ha aplicat la funció p^{-1}. Amb aquesta clau i les observacions aconseguim una nova manera de desxifrar, que és fer el mateix algoritme que per a xifrar en el mateix ordre, però utilitzant les inverses de cada funció i on la clau és la clau equivalent.

Capítol 3
Algoritme XSL

L'algoritme XSL és un algoritme criptoanalític que intenta trencar l'AES reduint-lo a resoldre un sistema d'equacions quadràtiques en diverses incògnites. Aquest problema també s'anomena problema MQ per les seves sigles en anglès (*Multivariate Quadratic equations*). Aquest tipus de problema és en el que es basen alguns sistemes de clau pública, com el criptosistema HFE. Adi Shamir en el seu article [Co,Kl,Pa,Sh-] mostra que encara que el problema MQ és NP-difícil la seva complexitat baixa en augmentar el nombre d'equacions, fent-lo sobredeterminat. A més a més, l'algoritme XSL intenta mostrar que si el sistema és escàs (un baix nombre de monomis diferents) i té una estructura regular, aleshores és encara més fàcil resoldre'l.

Els dissenyadors de l'algoritme, Nicolas T. Courtois i Josef Pieprzyk, van idear aquest algoritme per a criptosistemes que van anomenar XSL. Aquests algoritmes es diuen així per la seva composició.

X La ronda comença amb un XOR del text pla amb la clau.

S A continuació s'aplica un nombre finit de Caixes S bijectives en paral·lel, una per a cada bloc.

L Després s'aplica una capa lineal de difusió.

Es repeteix aquest esquema fins que s'arriba al nombre de rondes i llavors s'aplica la ronda final.

És clar que com és definit, l'AES és un criptosistema XSL. La part que correspon a la X és la suma amb la clau, la S és l'aplicació de la caixa S i la L és aplicar la permutació i el producte, a excepció de la ronda final en la qual només s'aplica la permutació. Courtois i Pieprzyk donen dos atacs XSL, on la diferència és en considerar l'algoritme de la clau en un, i en l'altre no.

3.1 Equacions del sistema

El primer pas de l'algoritme és trobar les equacions que descriuen el sistema. Primer trobem les equacions de la caixa S. En la secció anterior hem descrit la caixa S com una composició de dues funcions: g que és la inversa revisada, és a dir, la inversa en la qual $g(0) = 0$, i f que és una funció afí. Anomenem x a la unitat de missatge d'entrada de la caixa S. Sigui $y = g(x)$ l'invers de x en $\mathcal{B}$ i sigui $z = f(y)$ la unitat de missatge de sortida

de la caixa S. Primer aconseguim les equacions que defineixen la funció g. Per com és definida aquesta funció tenim que

$$1 = x \cdot y$$

en quasi tots els casos excepte si $x = 00$, on hem utilitzat la notació hexadecimal. Per tant, tindrem un sistema de la forma

$$\begin{cases} 0 &= (xy)_7, \\ 0 &= (xy)_6, \\ 0 &= (xy)_5, \\ 0 &= (xy)_4, \\ 0 &= (xy)_3, \\ 0 &= (xy)_2, \\ 0 &= (xy)_1, \\ 1 &= (xy)_0. \end{cases}$$

Les primeres set equacions sempre són certes sigui quin sigui el byte x. L'última equació, en canvi, és falsa si el byte x és 00, fent que, per descripció de la funció g, $y = 00$ i el producte resultant sigui el byte 00 en comptes del 01, on 00 i 01 representen una classe descrita amb dues paraules hexadecimals. Per tant, només hi ha un byte dels 256 possibles que fa que sigui falsa l'última equació. Per tant, si considerem que tots els bytes són equifreqüents, la freqüència amb la qual són certes les equacions és 1 per a les set primeres i $\frac{255}{256}$ per a l'última. Ens queda calcular els termes de la dreta de la igualtat en cada equació. Un cop fet el producte ens queda el sistema.

$$\begin{cases} 0 = x_7y_0 + x_6y_1 + x_5y_2 + x_4y_3 + x_3y_4 + x_2y_5 + x_1y_6 + x_0y_7 + x_7y_7 + x_7y_5 + x_6y_6 + \\ \quad + x_5y_7 + x_7y_4 + x_6y_5 + x_5y_6 + x_4y_7, \\ 0 = x_6y_0 + x_5y_1 + x_4y_2 + x_3y_3 + x_2y_4 + x_1y_5 + x_0y_6 + x_7y_6 + x_6y_7 + x_7y_4 + x_6y_5 + \\ \quad + x_5y_6 + x_4y_7 + x_7y_3 + x_6y_4 + x_5y_5 + x_4y_6 + x_3y_7, \\ 0 = x_5y_0 + x_4y_1 + x_3y_2 + x_2y_3 + x_1y_4 + x_0y_5 + x_7y_5 + x_6y_6 + x_5y_7 + x_7y_3 + x_6y_4 + \\ \quad + x_5y_5 + x_4y_6 + x_3y_7 + x_7y_2 + x_6y_3 + x_5y_4 + x_4y_5 + x_3y_6 + x_2y_7, \\ 0 = x_4y_0 + x_3y_1 + x_2y_2 + x_1y_3 + x_0y_4 + x_7y_4 + x_6y_5 + x_5y_6 + x_4y_7 + x_7y_2 + x_6y_3 + \\ \quad + x_5y_4 + x_4y_5 + x_3y_6 + x_2y_7 + x_7y_7 + x_7y_1 + x_6y_2 + x_5y_3 + x_4y_4 + x_3y_5 + x_2y_6 + \\ \quad + x_1y_7, \\ 0 = x_3y_0 + x_2y_1 + x_1y_2 + x_0y_3 + x_7y_4 + x_6y_5 + x_5y_6 + x_4y_7 + x_7y_3 + x_6y_4 + x_5y_5 + \\ \quad + x_4y_6 + x_3y_7 + x_7y_7 + x_7y_1 + x_6y_2 + x_5y_3 + x_4y_4 + x_3y_5 + x_2y_6 + x_1y_7 + x_7y_6 + \\ \quad + x_6y_7 + x_7y_5 + x_6y_6 + x_5y_7, \\ 0 = x_2y_0 + x_1y_1 + x_0y_2 + x_7y_3 + x_6y_4 + x_5y_5 + x_4y_6 + x_3y_7 + x_7y_2 + x_6y_3 + x_5y_4 + \\ \quad + x_4y_5 + x_3y_6 + x_2y_7 + x_7y_6 + x_6y_7, \\ 0 = x_1y_0 + x_0y_1 + x_7y_2 + x_6y_3 + x_5y_4 + x_4y_5 + x_3y_6 + x_2y_7 + x_7y_7 + x_7y_1 + x_6y_2 + \\ \quad + x_5y_3 + x_4y_4 + x_3y_5 + x_2y_6 + x_1y_7 + x_7y_5 + x_6y_6 + x_5y_7, \\ 1 = x_0y_0 + x_7y_1 + x_6y_2 + x_5y_3 + x_4y_4 + x_3y_5 + x_2y_6 + x_1y_7 + x_7y_6 + x_6y_7 + x_7y_5 + \\ \quad + x_6y_6 + x_5y_7. \end{cases}$$

El que volem és trobar les equacions que ens relacionen l'entrada amb la sortida de la caixa S, així que hem de trobar la relació entre y i z. Com que $z = f(y)$, aleshores $y = f^{-1}(z)$,

donant les relacions

$$\begin{cases} y_7 = z_6 + z_4 + z_1, \\ y_6 = z_5 + z_3 + z_0, \\ y_5 = z_7 + z_4 + z_2, \\ y_4 = z_6 + z_3 + z_1, \\ y_3 = z_5 + z_2 + z_0, \\ y_2 = z_7 + z_4 + z_1 + 1, \\ y_1 = z_6 + z_3 + z_0, \\ y_0 = z_7 + z_5 + z_2 + 1. \end{cases}$$

Substituint les y_i en el sistema que hem trobat abans per les relacions que acabem de trobar, obtenim el següent sistema.

$$\begin{cases} 0 = x_4z_0 + x_5z_0 + x_1z_0 + x_0z_6 + x_0z_4 + x_0z_1 + x_2z_7 + x_2z_4 + x_2z_2 + x_3z_6 + x_3z_3 + \\ \quad + x_3z_1x_4z_6 + x_4z_5 + x_4z_4 + x_4z_2 + x_4z_1 + x_5z_6 + x_5z_7 + x_5z_5 + x_5z_3 + x_6z_6 + \\ \quad + x_6z_7 + x_6z_5 + x_6z_4 + x_6z_2 + x_7z_5 + x_7z_3 + x_1z_5 + x_1z_3 + x_5 + x_7, \\ 0 = x_0z_0 + x_3z_0 + x_4z_0 + x_0z_5 + x_0z_3 + x_2z_6 + x_2z_3 + x_2z_1 + x_3z_6 + x_3z_5 + x_3z_4 + \\ \quad + x_3z_2 + x_3z_1 + x_4z_6 + x_4z_7 + x_4z_5 + x_4z_3 + x_5z_6 + x_5z_7 + x_5z_5 + x_5z_4 + x_5z_2 + \\ \quad + x_6z_5 + x_6z_3 + x_7z_6 + x_7z_2 + x_7z_1 + x_1z_7 + x_1z_4 + x_1z_2 + x_4 + x_6, \\ 0 = x_2z_0 + x_3z_0 + x_7z_0 + x_0z_7 + x_0z_4 + x_0z_2 + x_2z_6 + x_2z_5 + x_2z_4 + x_2z_2 + x_2z_1 + \\ \quad + x_3z_6 + x_3z_7 + x_3z_5 + x_3z_3 + x_4z_6 + x_4z_7 + x_4z_5 + x_4z_4 + x_4z_2 + x_5z_5 + x_5z_3 + \\ \quad + x_6z_6 + x_6z_2 + x_6z_1 + x_7z_5 + x_7z_1 + x_1z_6 + x_1z_3 + x_1z_1 + x_3 + x_5 + x_7, \\ 0 = z_0x_2 + z_0x_6 + z_0x_7 + z_0x_1 + x_0z_6 + x_0z_3 + x_0z_1 + x_2z_6 + x_2z_7 + x_2z_5 + x_2z_3 + \\ \quad + x_3z_6 + x_3z_7 + x_3z_5 + x_3z_4 + x_3z_2 + x_4z_5 + x_4z_3 + x_5z_6 + x_5z_2 + x_5z_1 + x_6z_5 + \\ \quad + x_6z_1 + x_7z_6 + x_7z_7 + x_7z_1 + x_1z_6 + x_1z_5 + x_1z_4 + x_1z_2 + x_1z_1 + x_2 + x_4 + \\ \quad + x_6 + x_7, \\ 0 = x_0z_0 + x_4z_0 + x_6z_0 + x_7z_0 + x_0z_5 + x_0z_2 + x_2z_6 + x_2z_5 + x_3z_6 + x_3z_5 + x_3z_1 + \\ \quad + x_4z_5 + x_4z_4 + x_5z_6 + x_5z_7 + x_5z_3 + x_5z_1 + x_6z_5 + x_6z_4 + x_6z_2 + x_6z_1 + x_7z_6 + \\ \quad + x_7z_7 + x_7z_3 + x_1z_6 + x_1z_7 + x_3 + x_6 + x_1, \\ 0 = x_3z_0 + x_4z_0 + x_6z_0 + x_1z_0 + x_0z_7 + x_0z_4 + x_0z_1 + x_2z_6 + x_2z_7 + x_2z_5 + x_2z_4 + \\ \quad + x_2z_2 + x_2z_1 + x_3z_6 + x_3z_5 + x_3z_4 + x_3z_3 + x_3z_1 + x_4z_7 + x_4z_5 + x_4z_4 + x_4z_3 + \\ \quad + x_4z_2 + x_5z_6 + x_5z_7 + x_5z_4 + x_5z_3 + x_5z_2 + x_5z_1 + x_6z_5 + x_6z_4 + x_6z_3 + x_6z_2 + \\ \quad + x_7z_7 + x_7z_4 + x_7z_3 + x_7z_2 + x_7z_1 + x_1z_6 + x_1z_3 + x_0 + x_2 + x_7, \\ 0 = x_0z_0 + x_2z_0 + x_3z_0 + x_5z_0 + x_7z_0 + x_0z_6 + x_0z_3 + x_2z_6 + x_2z_5 + x_2z_4 + x_2z_3 + \\ \quad + x_2z_1 + x_3z_7 + x_3z_5 + x_3z_4 + x_3z_3 + x_3z_2 + x_4z_6 + x_4z_7 + x_4z_4 + x_4z_3 + x_4z_2 + \\ \quad + x_4z_1 + x_5z_5 + x_5z_4 + x_5z_3 + x_5z_2 + x_6z_7 + x_6z_4 + x_6z_3 + x_6z_2 + x_6z_1 + x_7z_4 + \\ \quad + x_7z_3 + x_7z_2 + x_1z_6 + x_1z_7 + x_1z_5 + x_1z_4 + x_1z_2 + x_1z_1 + x_6 + x_7 + x_1, \\ 1 = x_0 + x_6 + x_2z_0 + x_5z_0 + x_6z_0 + x_0z_7 + x_0z_5 + x_0z_2 + x_2z_5 + x_2z_3 + x_3z_7 + x_3z_4 + \\ \quad + x_3z_2 + x_4z_6 + x_4z_3 + x_4z_1 + x_5z_6 + x_5z_5 + x_5z_4 + x_5z_2 + x_5z_1 + x_6z_6 + x_6z_7 + \\ \quad + x_6z_5 + x_6z_3 + x_7z_6 + x_7z_7 + x_7z_5 + x_7z_4 + x_7z_2 + x_1z_6 + x_1z_4 + x_1z_1. \end{cases}$$

Encara existeixen més equacions. Observem que $\forall x$ tenim que $x = x^2 * y$. Seguint el mateix raonament tenim que $x^2 = x^4 * y^2$, $x^4 = x^8 * y^4$, fins a arribar a $x^{128} = x * y^{128}$. Cadascuna és el quadrat de l'equació anterior i, com que elevar al quadrat és $\mathbb{F}_2$-lineal com a funció en diverses variables, aleshores aquestes 8 equacions generen el mateix conjunt d'equacions.

Aleshores s'escull una d'aquestes equacions (Courtois i Pieprzyk escullen l'última) i, com que és simètrica si canviem x per y, tenim les dues equacions $x^{128} = x*y^{128}$ i $y^{128} = x^{128}*y$. Cadascuna d'aquestes equacions ens dona un conjunt de 8 equacions en diverses variables.

$$
\begin{cases}
\begin{aligned}
0 = {} & x_3 + x_5 + x_6 + x_1 + x_2z_2 + x_5z_7 + x_7z_4 + x_7z_1 + x_7z_3 + x_0z_1 + x_6z_5 + x_6z_3 + \\
& + x_7z_7 + x_4z_6 + x_4z_1 + x_4z_5 + x_4z_0 + x_4z_2 + x_1z_5 + x_1z_3 + x_5z_5 + x_5z_3 + x_5z_0 + \\
& + x_3z_1 + x_3z_3 + x_6z_6 + x_3z_4 + x_2z_3 + x_2z_6 + x_4z_7 + x_0z_5 + x_0z_3 + x_1z_4 + x_1z_7 + \\
& + x_6z_1 + x_3z_0 + x_4z_3 + x_0z_7 + x_1z_6 + x_2z_5, \\
0 = {} & x_3 + x_6 + x_1 + x_2z_4 + x_5z_1 + x_7z_1 + x_5z_6 + x_0z_6 + x_0z_4 + x_6z_3 + x_6z_4 + \\
& + x_7z_5 + x_7z_2 + x_4z_5 + x_4z_0 + x_1z_5 + x_1z_3 + x_5z_5 + x_5z_3 + x_3z_1 + x_3z_3 + x_3z_6 + \\
& + x_4z_7 + x_0z_5 + x_0z_3 + x_1z_2 + x_6z_1 + x_3z_5 + x_3z_0 + x_3z_2 + x_4z_3 + x_0z_7 + x_3z_7 + \\
& + x_6z_7 + x_7z_7 + x_2z_1 + x_2z_3 + x_1z_6 + x_2z_0, \\
0 = {} & x_3 + x_4 + x_5 + x_1 + x_2z_2 + x_2z_7 + x_5z_1 + x_5z_4 + x_5z_7 + x_7z_6 + x_7z_4 + x_7z_1 + \\
& + x_6z_2 + x_6z_7 + x_7z_7 + x_4z_6 + x_4z_1 + x_4z_5 + x_1z_3 + x_1z_0 + x_5z_3 + x_3z_3 + \\
& + x_2z_6 + x_0z_5 + x_0z_3 + x_1z_4 + x_6z_1 + x_3z_5 + x_3z_0 + x_4z_3 + x_0z_2 + x_3z_7 + x_1z_1 + \\
& + x_0z_6 + x_6z_5 + x_2z_1 + x_2z_3 + x_2z_5 + x_2z_0, \\
0 = {} & x_3 + x_4 + x_1 + x_2z_7 + x_5z_1 + x_5z_7 + x_7z_4 + x_0z_4 + x_0z_1 + x_6z_4 + x_6z_7 + \\
& + x_7z_2 + x_4z_4 + x_4z_1 + x_1z_5 + x_1z_3 + x_1z_0 + x_5z_5 + x_3z_1 + x_3z_3 + x_3z_6 + \\
& + x_2z_3 + x_4z_7 + x_0z_3 + x_0z_0 + x_1z_2 + x_1z_7 + x_6z_1 + x_3z_5 + x_4z_3 + x_1z_1 + \\
& + x_1z_6 + x_7z_7 + x_7z_5 + x_6z_6 + x_5z_2 + x_2z_5 + x_2z_0, \\
0 = {} & x_2 + x_6 + x_7 + x_1 + x_2z_2 + x_5z_1 + x_5z_4 + x_7z_4 + x_7z_1 + x_5z_6 + x_7z_3 + \\
& + x_6z_2 + x_6z_4 + x_6z_7 + x_7z_7 + x_7z_2 + x_4z_6 + x_4z_0 + x_1z_0 + x_5z_5 + x_5z_3 + \\
& + x_2z_1 + x_0z_0 + x_1z_4 + x_6z_1 + x_3z_0 + x_4z_3 + x_0z_2 + \\
& + x_0z_6 + x_6z_3 + x_5z_0 + x_6z_6 + x_3z_7 + x_1z_6, \\
0 = {} & x_2 + x_3 + x_4 + x_5 + x_1 + x_2z_2 + x_2z_7 + x_5z_1 + x_5z_4 + x_7z_6 + x_7z_1 + \\
& + x_0z_4 + x_0z_1 + x_6z_5 + x_6z_2 + x_6z_4 + x_6z_7 + x_7z_2 + x_4z_4 + x_4z_2 + x_1z_5 + \\
& + x_5z_0 + x_3z_1 + x_3z_6 + x_6z_6 + x_5z_2 + x_3z_4 + x_2z_3 + x_2z_6 + x_4z_7 + x_0z_5 + \\
& + x_1z_2 + x_1z_4 + x_1z_7 + x_0z_7 + x_1z_1 + x_1z_6 + x_2z_5 + x_2z_0 + x_5z_6 + x_0z_6 + \\
& + x_1z_3 + x_5z_5 + x_0z_3 + x_0z_0, \\
0 = {} & x_0 + x_2 + x_3 + x_7 + x_2z_4 + x_5z_4 + x_5z_7 + x_7z_6 + x_7z_1 + x_5z_6 + x_0z_6 + \\
& + x_6z_2 + x_7z_7 + x_4z_6 + x_4z_4 + x_4z_1 + x_4z_5 + x_4z_0 + x_4z_2 + x_1z_5 + x_1z_3 + \\
& + x_6z_6 + x_5z_2 + x_3z_4 + x_2z_1 + x_2z_6 + x_7z_0 + x_0z_5 + x_0z_3 + x_1z_2 + x_1z_7 + \\
& + x_0z_2 + x_0z_7 + x_3z_7 + x_1z_6 + x_6z_1 + x_3z_2 + x_0z_4 + x_0z_1 + x_1z_0 + x_5z_5, \\
0 = {} & x_3 + x_5 + x_2z_4 + x_2z_7 + x_5z_1 + x_5z_7 + x_7z_6 + x_7z_1 + x_5z_6 + x_7z_3 + x_0z_6 + \\
& + x_6z_3 + x_6z_0 + x_6z_7 + x_7z_5 + x_4z_4 + x_4z_1 + x_4z_0 + x_1z_5 + x_1z_3 + x_5z_5 + \\
& + x_3z_3 + x_3z_6 + x_5z_2 + x_2z_3 + x_2z_6 + x_0z_0 + x_1z_7 + x_3z_5 + x_3z_2 + x_4z_3 + \\
& + x_0z_1 + x_6z_5 + x_5z_3 + x_5z_0 + x_1z_1 + x_2z_5 + x_0z_2, \\
0 = {} & x_5 + x_7 + z_7 + z_5 + z_3 + z_1 + x_5z_1 + x_5z_4 + x_7z_3 + x_0z_6 + x_0z_4 + x_0z_1 + \\
& + x_4z_2 + x_1z_5 + x_1z_0 + x_5z_3 + x_6z_6 + x_3z_4 + x_2z_3 + x_4z_7 + x_7z_0 + x_6z_1 + \\
& + x_6z_3 + x_7z_2 + x_4z_4 + x_2z_5 + x_2z_0 + x_3z_7,
\end{aligned}
\end{cases}
$$

$$
\begin{cases}
0 = x_3 + x_5 + x_7 + z_6 + z_7 + z_5 + z_4 + z_3 + x_2z_2 + x_2z_4 + x_2z_7 + x_7z_1 + \\
\quad + x_6z_2 + x_6z_4 + x_7z_7 + x_7z_2 + x_4z_6 + x_4z_1 + x_5z_3 + x_5z_0 + x_3z_1 + x_3z_3 + \\
\quad + x_3z_4 + x_0z_5 + x_0z_3 + x_0z_0 + x_1z_4 + x_1z_7 + x_6z_1 + x_4z_3 + x_6z_5 + x_6z_0 + \\
\quad + x_6z_6 + x_5z_2, \\
0 = x_3 + x_5 + x_6 + x_7 + x_1 + z_6 + z_5 + z_3 + z_2 + x_5z_1 + x_5z_7 + x_7z_6 + x_7z_1 + \\
\quad + x_6z_3 + x_6z_0 + x_6z_7 + x_4z_6 + x_4z_4 + x_4z_1 + x_4z_5 + x_4z_0 + x_4z_2 + x_1z_3 + \\
\quad + x_5z_2 + x_2z_1 + x_2z_3 + x_2z_6 + x_7z_0 + x_1z_4 + x_3z_0 + x_3z_2 + x_0z_2 + \\
\quad + x_0z_4 + x_6z_5 + x_3z_3 + x_6z_6 + x_0z_7 + x_1z_1, \\
0 = x_3 + x_4 + x_5 + x_1 + z_4 + z_3 + z_1 + z_0 + x_2z_2 + x_2z_4 + x_5z_1 + x_5z_6 + \\
\quad + x_6z_2 + x_6z_4 + x_6z_7 + x_7z_7 + x_7z_5 + x_4z_6 + x_4z_5 + x_4z_0 + x_1z_3 + x_1z_0 + \\
\quad + x_6z_6 + x_2z_1 + x_2z_6 + x_4z_7 + x_7z_0 + x_0z_3 + x_1z_2 + x_3z_2 + x_4z_3 + x_3z_7 + \\
\quad + x_0z_1 + x_6z_5 + x_5z_0 + x_3z_1 + x_2z_5 + x_2z_0 + x_0z_6, \\
0 = x_2 + x_3 + x_5 + x_6 + x_1 + z_6 + z_2 + z_0 + x_2z_7 + x_5z_1 + x_5z_4 + x_5z_7 + \\
\quad + x_6z_5 + x_7z_7 + x_7z_2 + x_4z_6 + x_4z_5 + x_1z_5 + x_1z_0 + x_5z_5 + x_5z_3 + x_5z_0 + \\
\quad + x_6z_6 + x_3z_4 + x_2z_6 + x_7z_0 + x_0z_5 + x_0z_0 + x_1z_2 + x_1z_7 + x_6z_1 + x_3z_0 + \\
\quad + x_7z_6 + x_7z_4 + x_7z_3 + x_3z_1 + x_3z_6 + x_3z_7 + x_1z_1 + x_0z_2, \\
0 = x_0 + x_3 + x_4 + x_5 + x_1 + z_6 + z_7 + z_5 + z_4 + z_3 + z_1 + z_0 + x_5z_1 + x_5z_7 + \\
\quad + x_0z_4 + x_0z_1 + x_6z_5 + x_6z_3 + x_6z_0 + x_6z_4 + x_7z_7 + x_7z_5 + x_4z_6 + x_4z_4 + \\
\quad + x_4z_2 + x_1z_5 + x_5z_0 + x_3z_1 + x_3z_3 + x_6z_6 + x_5z_2 + x_2z_3 + x_2z_6 + x_4z_7 + \\
\quad + x_1z_7 + x_6z_1 + x_3z_5 + x_3z_0 + x_0z_7 + x_1z_1 + x_1z_6 + x_2z_0 + x_7z_4 + x_5z_6 + \\
\quad + x_4z_1 + x_4z_5 + x_7z_0 + x_1z_4, \\
0 = x_2 + x_3 + x_7 + x_1 + z_6 + z_7 + z_5 + z_4 + z_3 + z_2 + z_1 + 1 + x_2z_2 + x_2z_4 + \\
\quad + x_5z_7 + x_7z_1 + x_7z_3 + x_0z_6 + x_6z_5 + x_6z_3 + x_6z_0 + x_6z_2 + x_6z_4 + x_6z_7 + \\
\quad + x_4z_4 + x_4z_1 + x_4z_5 + x_4z_0 + x_1z_5 + x_1z_3 + x_1z_0 + x_5z_5 + x_5z_0 + x_3z_1 + \\
\quad + x_2z_6 + x_0z_3 + x_0z_0 + x_3z_0 + x_3z_2 + x_4z_3 + x_3z_7 + x_1z_1 + x_2z_5 + x_2z_1 + \\
\quad + x_2z_7 + x_5z_4 + x_7z_7 + x_4z_6 + x_3z_6, \\
0 = x_0 + x_7 + x_1 + z_6 + z_2 + z_1 + z_0 + 1 + x_2z_4 + x_5z_4 + x_5z_7 + x_7z_4 + \\
\quad + x_6z_4 + x_6z_7 + x_4z_5 + x_4z_0 + x_1z_5 + x_2z_1 + x_2z_6 + x_0z_5 + x_1z_2 + x_1z_7 + \\
\quad + x_4z_3 + x_0z_2 + x_0z_7 + x_1z_1 + x_1z_6 + x_7z_3 + x_6z_2 + x_3z_5 + x_3z_0 + x_7z_1.
\end{cases}
$$

Combinant aquests conjunts d'equacions amb el conjunt trobat anteriorment, obtenim un conjunt de 24 equacions, de les quals 23 són certes amb freqüència 1 i una amb freqüència $\frac{255}{256}$.

3.2 Estructura algebraica dels atacs XSL

Courtois y Pieprzyk agafen la següent estructura. Anomenem $x_1, x_2, ...$ les incògnites que representen les unitats de missatge en una certa ronda abans de ser passades per la caixa S i anomenem $z_1, z_2, ...$ les incògnites que representen les unitats de missatge en la mateixa ronda que les x_i després de passar les x_i per la caixa S, és a dir, $z_8z_7z_6z_5z_4z_3z_2z_1 = f(g(x_8x_7x_6x_5x_4x_3x_2x_1))$, etcètera. Anomenem a l'anell quocient

$$\mathcal{C} = \frac{\mathbb{F}_2[x_1, x_2, ..., x_{1280}, z_1, z_2, ..., z_{1280}]}{(x_i^2 - x_i, z_j^2 - z_j,\ \forall i,\ \forall j)}.$$

La dimensió com a espai vestorial de $\mathcal{C}$ és 2^{2560}.

Per a cada byte tenim l'estructura següent. Sigui i la posició dins de la cadena bytes del byte sobre el qual estem treballant. Sigui $j = 1, ..., 8$ el bit j-èsim del byte, així que $x_{i_j} = x_{j+4i}$ i $z_{i_j} = z_{j+4i}$. Anomenem

$$\mathcal{C}_i = \frac{\mathbb{F}_2[x_{i_j}, z_{i_j} | j = i, ..., 8]}{(x_{i_j}^2 - x_{i_j}, z_{i_j}^2 - z_{i_j} | j = 1, ..., 8)}.$$

Com que $\mathcal{C}_i$ és un anell finit generat, aleshores admet un nombre finit de generadors $B' = \{x_{i_j}, z_{i_j}, x_{i_j} z_{i_l}, 1\}$. El nombre d'elements de B_i' és $\#B_i' = 81$. A més a més, per la secció anterior sabem que ha de complir les 24 equacions que hem trobat.

Proposició 3.2.1. *Les equacions trobades anteriorment de la caixa S són linealment independents en $\mathcal{C}$.*

□

Com que les equacions són linealment independents, aleshores podem escriure 24 monomis que apareixen en la base com a combinació de la resta. Així que una base de $\mathcal{C}_i$ serà $B_i = \{x_{i_j} z_{i_l} | j \neq l\}$.

3.3 Nucli de l'atac

En aquest pas es fa ús de les bases trobades anteriorment. Anomenarem caixa S activa a la caixa que estem aconseguint més equacions per descriure-la. Anomenarem caixes S passives a la resta de caixes de l'algoritme de xifratge.

Per a cada ronda tindrem per unitat de missatge una equació del tipus

$$z_i + k_i = x_{i+1},$$

on z_i és el valor de la unitat de missatge després d'aplicar la caixa S, x_i és el valor de la unitat de missatge abans d'utilitzar la caixa S i k_i és la unitat de clau que li correspon per posició i ronda.

En el primer atac, k_i és considerada una constant, així que se substitueixen k_i i x_{i+1} per l'expressió que li pertoqui com a element de $\mathcal{C}_i$ en la base definida anteriorment.

En el cas del segon atac, les equacions de la secció anterior també descriuen les caixes S utilitzades en la clau, així que de la mateixa manera es pot descriure un $\mathcal{C}'$ a partir de $\mathcal{C}$, però amb les variables k_i i w_i, on k_i és el valor de la unitat de clau abans de passar per la caixa S si és que hi passa i en cas contrari el valor de la unitat de clau en la posició i, i w_i és la unitat de clau després d'haver passat per la caixa S la unitat de clau k_i. Per exemple, suposem que tenim un sistema amb una addició de la clau de la següent manera:

$$\begin{cases} p_0 + w_{10} + k_{00} = 0, \\ p_1 + w_{11} + k_{01} = 0, \\ p_2 + w_{12} + k_{02} = 0, \\ p_3 + w_{13} + k_{03} = 0, \end{cases}$$

on p_i és el text pla. Suposem que les caixes S són definides per les equacions

$$
\begin{aligned}
0 =& w_{10} + w_{10}x_{11} + w_{11}x_{10} + w_{11}x_{12} + w_{12}x_{10} + w_{13}x_{11} + 1,\\
0 =& w_{11} + w_{10}x_{11} + w_{10}x_{13} + w_{11}x_{13} + w_{12}x_{10} + w_{12}x_{13} + w_{13}x_{10} + w_{13}x_{11},\\
0 =& w_{12} + w_{10}x_{11} + w_{10}x_{12} + w_{12}x_{11} + w_{12}x_{13} + w_{13}x_{10} + w_{13}x_{11},\\
0 =& w_{13} + w_{10}x_{11} + w_{10}x_{12} + w_{10}x_{13} + w_{11}x_{10} + w_{11}x_{13} + w_{12}x_{10} + w_{12}x_{13} + w_{13}x_{10},\\
0 =& x_{10} + w_{10}x_{11} + w_{10}x_{12} + w_{11}x_{10} + w_{11}x_{13} + w_{12}x_{11} + 1,\\
0 =& x_{11} + w_{10}x_{12} + w_{10}x_{13} + w_{11}x_{10} + w_{11}x_{13} + w_{13}x_{10} + w_{13}x_{11} + w_{13}x_{12},\\
0 =& x_{12} + w_{10}x_{13} + w_{11}x_{10} + w_{11}x_{12} + w_{11}x_{13} + w_{12}x_{10} + w_{13}x_{12},\\
0 =& x_{13} + w_{10}x_{11} + w_{10}x_{12} + w_{10}x_{13} + w_{11}x_{10} + w_{12}x_{10} + w_{13}x_{10} + w_{13}x_{11} + w_{13}x_{12},\\
0 =& w_{10}x_{10} + w_{10}x_{11} + w_{11}x_{10} + w_{12}x_{13} + w_{13}x_{12} + 1,\\
0 =& w_{11}x_{11} + w_{10}x_{12} + w_{10}x_{13} + w_{11}x_{12} + w_{12}x_{10} + w_{12}x_{11} + w_{12}x_{13} + w_{13}x_{10} + w_{13}x_{12},\\
0 =& w_{12}x_{12} + w_{10}x_{11} + w_{11}x_{10} + w_{11}x_{13} + w_{12}x_{13} + w_{13}x_{11} + w_{13}x_{12},\\
0 =& w_{13}x_{13} + w_{10}x_{13} + w_{11}x_{12} + w_{12}x_{11} + w_{13}x_{10},
\end{aligned}
$$

amb w_{ij} i x_{ij} el j-èsim bit d'entrada i de sortida de la i-èsima caixa S respectivament. Suposem que la caixa S de la clau és la mateixa, amb $k_{i,j}$ i s_{ij} el j-èsim bit d'entrada i de sortida de la i-èsima caixa S respectivament. Aleshores fent les substitucions en el sistema d'equacions del criptosistema obtenim el sistema d'equacions

$$
\begin{cases}
p_0 + w_{10}x_{11} + w_{11}x_{10} + w_{11}x_{12} + w_{12}x_{10} + w_{13}x_{11} + k_{00}s_{01} + k_{01}s_{00} + k_{01}s_{02} \\
\quad + k_{02}s_{00} + k_{03}s_{01} = 0,\\
p_1 + w_{10}x_{11} + w_{10}x_{13} + w_{11}x_{13} + w_{12}x_{10} + w_{12}x_{13} + w_{13}x_{10} + w_{13}x_{11} + k_{00}s_{01} \\
\quad + k_{00}s_{03} + k_{01}s_{03} + k_{02}s_{00} + k_{02}s_{03} + k_{03}s_{00} + k_{03}s_{01} = 0,\\
p_2 + w_{10}x_{11} + w_{10}x_{12} + w_{12}x_{11} + w_{12}x_{13} + w_{13}x_{10} + w_{13}x_{11} + k_{00}s_{01} + k_{00}s_{02} \\
\quad + k_{02}s_{01} + k_{02}s_{03} + k_{03}s_{00} + k_{03}s_{01} = 0,\\
p_3 + w_{10}x_{11} + w_{10}x_{12} + w_{10}x_{13} + w_{11}x_{10} + w_{11}x_{13} + w_{12}x_{10} + w_{12}x_{13} + w_{13}x_{10} \\
\quad + k_{00}s_{01} + k_{00}s_{02} + k_{00}s_{03} + k_{01}s_{00} + k_{01}s_{03} + k_{02}s_{00} + k_{02}s_{03} + k_{03}s_{00} = 0.
\end{cases}
$$

El primer atac proposat per Pieprzyk i Courtois no té en compte l'estructura de la clau. Per aquest motiu per a poder portar-lo a terme fa falta conèixer $N_r + 1$ parelles de text pla/text xifrat. Per tant, el nombre total de caixes S serà $S = B * N_r * (N_r + 1)$, on B és el nombre de caixes S per ronda. Si anomenem s el nombre de bits per caixa S, aleshores tenim que el nombre total d'equacions serà $s * S$.

El segon atac té en compte l'estructura de la clau. Sigui Λ el nombre de textos necessaris per a poder determinar la clau. Anomenem D el nombre de caixes S de la clau i E el nombre de caixes S artificials addicionals. El nombre total de caixes S en aquest atac és de

$$S = \Lambda \cdot B \cdot N_r + D + E.$$

Notem que en l'algoritme de la clau no totes les variables passen per una caixa S, així que per a tenir-les totes, s'introdueix l'anomenada caixa S artificial. Aquesta caixa està formada per les variables que falten, però no conté cap equació. Per tant, en aquesta caixa, $r = 0$ i els monomis són les variables que ens feien falta afegir.

Per a poder dur a terme l'algoritme ens fa falta introduir un paràmetre que anomenarem $P \in \mathbb{N}$. Considerem el conjunt de totes les caixes S passives. Anomenem S_i a la i-èsima caixa S passiva. Aleshores se seleccionen $P - 1$ caixes S passives, que anomenem

$S_{i_1}, ..., S_{i_{P-1}}$, tals que les equacions que descriuen aquestes caixes no tinguin cap incògnita en comú amb les equacions que descriuen la caixa S activa. Abans hem vist que les equacions de les caixes S ens donen una base per a $\mathcal{C}_i$, per a un cert i.

La caixa S activa actua sobre un byte en concret sobre una ronda en específic. Cadascuna de les equacions que descriuen com actua el criptosistema sobre aquest byte en aquesta ronda es multiplica per cada element de la base de $\mathcal{C}_{i_j}$, on $\mathcal{C}_{i_j}$ és el $\mathcal{C}_i$ que té per indeterminades les mateixes incògnites que la caixa S_{i_j}, per a $j = 1, ..., P-1$.

Proposició 3.3.1. *El nombre total d'equacions generades per aquest pas és*

$$R = s \cdot S \cdot (t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix},$$

on t és el nombre de monomis diferents en $\mathcal{C}_i$ i r el nombre d'equacions.

Demostració. Primer mirem el nombre d'equacions que ens surten en descriure el sistema. Per a cada bit del bloc tindrem una equació per ronda. En el cas del segon atac, a més a més tenim una equació per a cada bit de la clau. Per tant, la quantitat d'equacions inicials és $s \cdot S$, on s és el nombre de bits que necessita cada caixa S i S el nombre de caixes S del criptosistema.

En el pas següent per a cada equació de les anteriors, se seleccionen $P-1$ caixes S del total de les caixes S passives, que són totes excepte una, i es fan les multiplicacions per la base de $\mathcal{C}_i$ per a la i que correspongui segons les caixes seleccionades. Com que totes les caixes són iguals, aleshores totes tenen el mateix nombre d'elements en la base de $\mathcal{C}_i$, que és $t-r$. Com que a cada equació la multipliquem per $t-r$ monomis de cadascuna de les bases de les $P-1$ caixes S seleccionades, obtenim que

$$R = s \cdot S \cdot (t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix}.$$

□

En el cas de l'AES, $t = 81$ i $r = 24$. r és el nombre d'equacions trobades en la secció 3.1. El valor t bé de la suma dels monomis per a cada caixa S. Tenim els monomis x_i, amb $i = 1, ..., 8$, z_i, $i = 1, ..., 8$, $x_i z_j$, $i = 1, ..., 8$ i $j = 1, ..., 8$ i el monomi 1, així que hi ha un total de $t = 8 + 8 + 8 \cdot 8 + 1 = 81$.

Proposició 3.3.2. *El nombre total de termes en les equacions serà de*

$$T \approx (t-r)^p \cdot \begin{pmatrix} S \\ P \end{pmatrix}.$$

Demostració. Per a cada caixa S tenim una base de monomis de $t-r$ monomis. Per tant, en total tenim $(t-r) \cdot \begin{pmatrix} S \\ 1 \end{pmatrix}$monomis diferents. Com hem vist abans, en fer els productes obtenim que cada monomi el multipliquem per $(t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix}$. Per tant, la quantitat de termes en les equacions serà de

$$T = \cdot(t-r)^P \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix} \cdot \begin{pmatrix} S \\ 1 \end{pmatrix}.$$

Ara, tenim que

$$\begin{pmatrix} S-1 \\ P-1 \end{pmatrix} = \frac{(S-1)!}{(S-P)!(P-1)!},$$

$$\begin{pmatrix} S \\ 1 \end{pmatrix} = \frac{S!}{(S-1)!},$$

així que

$$\begin{pmatrix} S-1 \\ P-1 \end{pmatrix} \cdot \begin{pmatrix} S \\ 1 \end{pmatrix} = \frac{S!}{(S-P)!(P-1)!} = \frac{S!P}{(S-P)!P!} = P \cdot \begin{pmatrix} S \\ P \end{pmatrix}.$$

Com volem un P enter petit, anomenem petit a un valor < 10, aleshores podem aproximar T per

$$T \approx (t-r)^p \cdot \begin{pmatrix} S \\ P \end{pmatrix}.$$

□

Proposició 3.3.3. *Anomenem $\mathcal{T}_i'$ al conjunt de monomis del sistema tal que $x_i \cdot \mathcal{T}_i' \subset \mathcal{T}$, on $\mathcal{T}$ és el conjunt de tots els monomis del sistema. Anomenem T' al cardinal de $\mathcal{T}_i'$. Amb aquesta notació tenim que*

$$T' = t' \cdot (t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix},$$

sent $t' < t$ el nombre de termes en la base d'una caixa S tal que quan són multiplicats per una variable fixada, encara hi són a la base.

Demostració. Inicialment, per a cada incògnita x_i o z_i, o k_i en el cas del segon atac, tenim que hi ha en les equacions inicials t' monomis que pertanyen a $\mathcal{T}_i'$. Ara, al fer els productes, cadascun d'aquests termes és multiplicat per $(t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix}$ monomis. Aquests nous termes pertanyen a $\mathcal{T}_i'$, així que

$$T' = t' \cdot (t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix}.$$

□

Ens queda calcular el valor de P. Fixem-nos que l'objectiu de l'atac és aconseguir un sistema sobredeterminat amb molt poques incògnites. Per tant, el que es vol assolir és que $R > T - T'$. Substituint per les expressions trobades abans tenim que

$$s \cdot S \cdot (t-r)^{P-1} \cdot \begin{pmatrix} S \\ P-1 \end{pmatrix} > (t-r)^P \cdot \begin{pmatrix} S \\ P \end{pmatrix} - t' \cdot (t-r)^{P-1} \cdot \begin{pmatrix} S-1 \\ P-1 \end{pmatrix}.$$

Dividint per $(t-r)^{P-1} \cdot \begin{pmatrix} S \\ P-1 \end{pmatrix}$ i multiplicant per S en ambdos costats de la desigualtat tenim que

$$\frac{S^2 s}{S-P+1} > \frac{(t-r)S}{P} - t'.$$

Per al cas de l'AES, $t = 81$, $r = 24$, $t' = 17$ i $s = 8$, així que P ha de complir que

$$\frac{8S^2}{S - P + 1} > \frac{57S}{P} - 17.$$

En el cas del primer atac, si suposem que tenim $N_r + 1$ parelles de text pla/ text xifrat,

$$S = B \cdot N_r \cdot (N_r + 1) = 16 \cdot 10 \cdot 11 = 1760.$$

En el cas del segon atac, si suposem que tenim $\Lambda = 1$ parella de text pla/ text xifrat,

$$S = \Lambda \cdot B \cdot N_r + D + E = 1 \cdot 16 \cdot 10 + D + 1,$$

on D depèn del tipus d'AES escollit.

Proposició 3.3.4. *Per al primer atac, $P \geq 8$. Per al segon atac, $P \geq 7$.*

3.4 Mètode T'

Aquest algoritme s'aplica just abans d'aplicar linealització al sistema d'equacions. Anomenem $Free$ al nombre d'equacions del sistema linealment independents. L'objectiu d'aquest mètode és que $Free \approx T$ sense crear cap monomi diferent dels que ja tenim. S'espera que després d'unes quantes iteracions $Free = T - 1$ o $Free = T$.

Utilitzant la notació de l'apartat anterior, suposem que $Free \geq T - T'_i + C$, on T'_i és definida a partir de x_i per a qualsevol i i $C \geq 1$. L'algoritme és el següent.

1. Fer eliminació gaussiana fins a tenir el sistema de la forma $x_l x_k = eq_1$, on $x_l x_k$ és un terme de T i eq_1 és una expressió de termes de T', amb T' definida per a x_i.
2. Es torna a fer eliminació gaussiana del sistema original l'únic que ara amb T' definida per x_j, amb $j \neq i$.

Ara tenim dos sistemes d'equacions on en cadascun hi ha C equacions que només tenen termes de T'.

3. En el primer sistema de C equacions multipliquem cada equació per x_i i les afegim al sistema original si no la teníem ja.
4. Si no tenim equacions noves i $Free \neq T$ i $Free \neq T-1$, aleshores agafem unes noves variables i tornem al punt 1.
5. Si no tenim equacions noves i $Free = T$ o $Free = T - 1$, ja hem acabat.
6. Utilitzant el sistema del punt 2, es fan substitucions sobre el sistema del punt 3 i afegim les C equacions del segon sistema.
7. Es multiplica aquest últim sistema d'equacions per x_j.
8. Es descarten les equacions que queden invariants i la resta s'afegeixen al sistema.
9. Si no tenim equacions noves i $Free \neq T$ i $Free \neq T-1$, aleshores agafem unes noves variables i tornem al punt 1.
10. Si no tenim equacions noves i $Free = T$ o $Free = T - 1$, ja hem acabat.

11. Les equacions que queden es reescriuen utilitzant el sistema del punt 1.
12. Es multipliquen les equacions per x_i.
13. Es descarten les equacions que no són linealment independents i la resta s'afegeixen al sistema.
14. Si no tenim equacions noves i $Free \neq T$ i $Free \neq T-1$, aleshores agafem unes noves variables i tornem al punt 1.
15. Si no tenim equacions noves i $Free = T$ o $Free = T-1$, ja hem acabat.
16. Les equacions que queden es reescriuen utilitzant el sistema del punt 2.
17. Es torna al pas 7.

Si no traiem cap equació nova amb aquest mètode, aleshores el sistema d'equacions es queda invariant i el mètode T' no és útil.

En aquest algoritme s'utilitza un conjunt de monomis que hem descrit anteriorment. En la secció del nucli de l'atac, 3.3, hem vist que hi ha un conjunt de monomis que en ser multiplicats per una de les variables dona elements d'aquest conjunt. Aquest conjunt anteriorment s'ha anomenat $\mathcal{T}_i$, on i indica la variable per la qual el producte per la variable és tancat.

En aquest algoritme en específic s'utilitzen dos d'aquests conjunts, que anomenarem $\mathcal{T}_i$ i $\mathcal{T}_j$, seguint la notació utilitzada en aquesta secció. El primer pas és descriure els elements de $\mathcal{T} - \mathcal{T}_i$ en funció dels elements de $\mathcal{T}_i$ usant les equacions (respectivament amb $\mathcal{T}_j$). Ara s'obtenen expressions dels elements de la base de $\mathcal{C}$ en funció dels monomis de $\mathcal{T}_i$ o $\mathcal{T}_j$. En fer els productes i les substitucions estem buscant elements perquè formin part de la base de $\mathcal{C}$, aconseguint noves equacions que descriuen el sistema.

Anem a veure un exemple d'aquest algoritme. Comencem amb un sistema amb $n = 5$ variables, així que $T = 16$ i $T' = 10$, amb una única solució i $Free = T - T' + 2$. Suposem que el sistema amb T' definit a partir de x_1 és

$$\begin{cases} x_3x_2 &= x_1x_3 + x_2 \\ x_3x_4 &= x_1x_4 + x_1x_5 + x_5 \\ x_3x_5 &= x_1x_5 + x_4 + 1 \\ x_2x_4 &= x_1x_3 + x_1x_5 + 1 \\ x_2x_5 &= x_1x_3 + x_1x_2 + x_3 + x_4 \\ x_4x_5 &= x_1x_2 + x_1x_5 + x_2 + 1 \\ 0 &= x_1x_3 + x_1x_4 + x_1 + x_5 \\ 1 &= x_1x_4 + x_1x_5 + x_1 + x_5 \end{cases},$$

i el sistema amb T' definit a partir de x_2 és

$$\begin{cases} x_1x_3 &= x_3x_2 + x_2 \\ x_1x_4 &= x_3x_2 + x_2 + x_1 + x_5 \\ x_1x_5 &= x_2x_4 + x_3x_2 + x_2 + 1 \\ x_3x_5 &= x_2x_4 + x_3x_2 + x_2 + x_4 \\ x_3x_4 &= x_2x_4 + x_1 + 1 \\ x_4x_5 &= x_1x_2 + x_2x_4 + x_3x_2 \\ 0 &= x_1x_2 + x_2x_5 + x_3x_2 + x_2 + x_3 + x_4 \\ 0 &= x_2x_4 \end{cases}.$$

Tenim que el rang és 8. Multipliquem les dues últimes equacions del primer sistema per x_1 i obtenim que

$$\begin{cases} 0 &= x_1x_3 + x_1x_4 + x_1 + x_1x_5 \\ 0 &= x_1x_4 \end{cases}.$$

Aquestes equacions són linealment independents de les que ja teníem, així que el rang passa a ser 10. Utilitzant el segon sistema fem substitucions sobre aquestes dues equacions i afegim les dues últimes equacions del segon sistema, obtenint el sistema

$$\begin{cases} 0 &= x_1x_2 + x_2x_5 + x_3x_2 + x_2 + x_3 + x_4 \\ 0 &= x_2x_4 \\ 0 &= x_2x_4 + x_3x_2 + x_5 + x_2 + 1 \\ 0 &= x_3x_2 + x_2 + x_1 + x_5 \end{cases}.$$

Multipliquem les quatre equacions per x_2, obtenint el sistema

$$\begin{cases} 0 &= x_1x_2 + x_2x_5 + x_2x_4 + x_2 \\ 0 &= x_2x_4 \\ 0 &= x_2x_4 + x_3x_2 + x_5x_2 \\ 0 &= x_3x_2 + x_2 + x_1x_2 + x_2x_5 \end{cases},$$

on la segona equació és linealment dependent i la resta linealment independents, així que el rang és 13. Reescrivim les tres equacions linealment independents utilitzant el primer sistema, obtenint el sistema

$$\begin{cases} 1 &= x_1x_5 + x_2 + x_3 + x_4 \\ 1 &= x_1x_2 + x_1x_3 + x_1x_5 + x_2 + x_3 + x_4 \\ 0 &= x_3 + x_4 \end{cases}.$$

El multipliquem per x_1 i tenim que

$$\begin{cases} 0 &= x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_1 \\ 0 &= x_1x_4 + x_1x_5 + x_1 \\ 0 &= x_1x_3 + x_1x_4 \end{cases},$$

on la segona i la tercera equació són linealment dependents, així que el rang és 14. Finalment, substituïm les equacions del segon sistema sobre la primera equació i la multipliquem per x_2, obtenint l'equació

$$0 = x_1x_2 + x_2x_4 + x_3x_2 + x_2x_5,$$

que és linealment independent, així que el rang és 15. Com que hem arribat a $Free = 15 = 16 - 1 = T - 1$, aleshores hem acabat.

Teorema 3.4.1. *L'algoritme T' acaba.*

Demostració. El mètode comença escollint el nombre d'incògnites sobre les quals treballa. Aquest nombre és finit, ja que cada incògnita representa una unitat de missatge en una ronda en concret. Com que tenim un nombre finit de rondes i un nombre finit d'unitats de missatge, no es pot escollir més incògnites de les que hi ha. Per tant, aquest pas de l'algoritme és finit.

Un cop ha seleccionat el nombre d'incògnites sobre les quals treballa ha de seleccionar quines. De la mateixa manera que abans hi ha un nombre finit d'incògnites, així que hi ha un nombre finit d'eleccions.

En el següent pas aplica eliminació gaussiana sobre el sistema d'equacions. Aquest sistema d'equacions és finit, ja que per a cada unitat de missatge que hi ha en un bloc, tenim un total d'11 equacions que el descriuen, així que tindrem un nombre finit d'equacions. A més a més les equacions són finites com hem vist en seccions anteriors. De la mateixa manera ocorre amb les equacions de la clau. Per tant, aquest pas és finit.

Després selecciona les equacions del sistema reduït extres. Hi ha una quantitat finita d'aquestes equacions. Per tant, aquest pas acaba. També es fan un nombre finit de substitucions i multiplicacions, per ser la quantitat d'equacions del sistema finita.

L'últim pas que ens queda mirar és a l'hora d'afegir les equacions en el sistema original.

Si el nombre d'equacions sobre el que estem operant va decreixent eventualment arribarem a no tenir-ne cap i, per tant, aquest pas acabaria.

Si el nombre d'equacions sobre el que estem operant arriba un punt en el qual no baixa en nombre d'equacions, aleshores s'aniran afegint equacions fins que el nombre d'equacions del sistema sigui T o $T-1$. En qualsevol dels dos casos l'algoritme acaba.

Si no hi afegim equacions noves, aleshores l'algoritme les descarta i el sistema d'equacions sobre el que treballem no té equacions, el que significa que acaba.

Com que tots els passos de l'algoritme són finits, aleshores l'algoritme acaba. □

Un cop ja tenim el sistema s'aplica el mètode conegut com a linealització. En aquest mètode, s'afegeix una nova variable per cada terme i es resol el sistema lineal. Per exemple, sigui

$$F(x,y) = x + y + xy + 1.$$

Aleshores podem definir

$$s = f(x,y) = x + y,$$
$$t = g(x,y) = xy.$$

Així que

$$F(x,y) = F(s,t) = s + t + 1,$$

que ara és una funció lineal.

Capítol 4
Conclusions

Des del naixement de l'AES s'han buscat maneres de poder trencar-ho. Com que els mètodes més utilitzats en l'actualitat semblen no ser massa eficaços, se n'ha hagut d'inventar de nous i un d'ells és l'XSL. En aquest treball hem descrit l'XSL i hem explicat l'àlgebra que hi ha enrere. També hem vist que en el seu estat actual l'algoritme té una complexitat molt elevada, ja que s'ha de fer linealització. Resoldre aquest tipus de sistema té una complexitat $\mathrm{O}(n^3)$, on n és el nombre d'equacions del sistema. En aquest cas, $n = R$ i, com hem vist fins ara, a l'acabar el mètode T', tenim que $R \geq T$, sent T el nombre total de monomis que hi ha en el sistema. Hem vist que aquests monomis formen una base de $\mathcal{C}$, que és un espai de dimensió 2^{2560}. Tot i això, aquest algoritme és una bona primera aproximació als atacs algebraics contra criptosistemes de clau privada.

Referències

[Ci,Le-05] Carlos Cid, Gaëtan Leurent: An analysis of the XSL Algorithm. *Advances in Criptology-ASIACRYPT 2005*, vol. 3788 de *Lecture Notes in Computer Science*, p. 333-352. Springer, 2005.

[Co,Kl,Pa,Sh-] Nicolas Courtois, Alexander Klimov, Jacques Patarin, Adi Shamir: Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. *Advances in Cryptology-EUROCRYPT 2000*, vol. 1807 de *Lecture Notes in Computer Science*, p. 392-407. Springer, 2000.

[Co,Pi-02-1] Nicolas Courtois, Josef Pieprzyk: Cryptoanalysis of Block Ciphers with Overdefined System of Equations. *Crytology ePrint Archive*, Report 2002/044, 2002.
https://eprint.iacr.org/2002/044

[Co,Pi-02-2] Nicolas Courtois, Josef Pieprzyk: Cryptoanalysis of Block Ciphers with Overdefined System of Equations. *Advances in Cryptology-ASIACRYPT 2002*, vol. 2501 de *Lecture Notes in Computer Science*, p. 267-287. Springer, 2002.

[Da,Ri-02] Joan Daemen, Vincent Rijmen: *The Design of Rijndael* Springer-Verlag, Berlin, Heidelberg, New York, 2002.
ISBN 978-3-642-07646-6, e-ISBN 978-3-662-04722-4

[Mu,Ro-02] Sean Murphy, Matthew J.B. Robshaw: Essential Algebraic Structure within the AES. *Advances in Cryptology-CRYPTO 2002*, vol. 2442 de *Lecture Notes in Computer Science*, p. 1-16. Springer, 2002.

[NIST-01] Morris J. Dworkin, Elaine B. Barker, James R. Nechvatal, James Foti, Lawrence E. Bassham, E. Roback, James F. Dray Jr. : Announcing the Advanced Encryption Standard (AES), *National Institute of Standards and Technology (NIST)* Federal Information Processing Standards Publication 197 (2001).

[Sh-49] Claude E. Shannon: Communication Theory of Secrecy Systems. *Bell System Technical Journal*, vol. 28-4, p. 656-715. 1949.

Apèndix A
Taules

Una manera alternativa de representar $\mathbb{F}_{2^8}$ és mitjançant paraules formades per nombres hexadecimals. Cada quatre bits es poden representar com un nombre hexadecimal, per tant, un element el $\mathbb{F}_{2^8}$ es pot representar amb xy, on

$$x, y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}.$$

Si utilitzem aquesta notació i la descrita en 2.1, l'escriptura de les taules queda més senzilla.

$f(xy)$:

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	5D	42	1F	00	21	3E	9B	84	A5	BA	E7	F8	D9	C6
	1	92	8D	AC	B3	EE	F1	D0	CF	6A	75	54	4B	16	09	28	37
	2	80	9F	BE	A1	FC	E3	C2	DD	78	67	46	59	04	1B	3A	25
	3	71	6E	4F	50	0D	12	33	2C	89	96	B7	A8	F5	EA	CB	D4
	4	A4	BB	9A	85	D8	C7	E6	F9	5C	43	62	7D	20	3F	1E	01
	5	55	4A	6B	74	29	36	17	08	AD	B2	93	8C	D1	CE	EF	F0
	6	47	58	79	66	3B	24	05	1A	BF	A0	81	9E	C3	DC	FD	E2
	7	B6	A9	88	97	CA	D5	F4	EB	4E	51	70	6F	32	2D	0C	13
	8	EC	F3	D2	CD	90	8F	AE	B1	14	0B	2A	35	68	77	56	49
	9	1D	02	23	3C	61	7E	5F	40	E5	FA	DB	C4	99	86	A7	B8
	A	0F	10	31	2E	73	6C	4D	52	F7	E8	C9	D6	8B	94	B5	AA
	B	FE	E1	C0	DF	82	9D	BC	A3	06	19	38	27	7A	65	44	5B
	C	2B	34	15	0A	57	48	69	76	D3	CC	ED	F2	AF	B0	91	8E
	D	DA	C5	E4	FB	A6	B9	98	87	22	3D	1C	03	5E	41	60	7F
	E	C8	D7	F6	E9	B4	AB	8A	95	30	2F	0E	11	4C	53	72	6D
	F	39	26	07	18	45	5A	7B	64	C1	DE	FF	E0	BD	A2	83	9C

$f^{-1}(xy)$:

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	05	4F	91	DB	2C	66	B8	F2	57	1D	C3	89	7E	34	EA	A0
	1	A1	EB	35	7F	88	C2	1C	56	F3	B9	67	2D	DA	90	4E	04
	2	4C	06	D8	92	65	2F	F1	BB	1E	54	8A	C0	37	7D	A3	E9
	3	E8	A2	7C	36	C1	8B	55	1F	BA	F0	2E	64	93	D9	07	4D
	4	97	DD	03	49	BE	F4	2A	60	C5	8F	51	1B	EC	A6	78	32
	5	33	79	A7	ED	1A	50	8E	C4	61	2B	F5	BF	48	02	DC	96
	6	DE	94	4A	00	F7	BD	63	29	8C	C6	18	52	A5	EF	31	7B
	7	7A	30	EE	A4	53	19	C7	8D	28	62	BC	F6	01	4B	95	DF
	8	20	6A	B4	FE	09	43	9D	D7	72	38	E6	AC	5B	11	CF	85
	9	84	CE	10	5A	AD	E7	39	73	D6	9C	42	08	FF	B5	6B	21
	A	69	23	FD	B7	40	0A	D4	9E	3B	71	AF	E5	12	58	86	CC
	B	CD	87	59	13	E4	AE	70	3A	9F	D5	0B	41	B6	FC	22	68
	C	B2	F8	26	6C	9B	D1	0F	45	E0	AA	74	3E	C9	83	5D	17
	D	16	5C	82	C8	3F	45	AB	E1	44	0E	D0	9A	6D	27	F9	B3
	E	FB	B1	6F	25	D2	98	46	0C	A9	E3	3D	77	80	CA	14	5E
	F	5F	15	CB	81	76	3C	E2	A8	0D	47	99	D3	24	6E	B0	FA

$g(xy)$:

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
	1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
	2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
	3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
	4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
	5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
	6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
	7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
	8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
	9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
	A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
	B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
	C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
	D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
	E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
	F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

Amb aquestes tres taules i anomenant $S_{RD}(xy) = f(g(xy))$, podem obtenir les dues taules següents.

$S_{RD}(xy)$:

		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

$S_{RD}^{-1}(xy)$:

		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

Apèndix B
Algoritmes

Els algoritmes en pseudocodi que es presenten a continuació són extrets de [NIST-01, pag. 15, 20, 21 i 25].

B.1 Algoritme Rijndael de xifratge

Anomenem *State* a la matriu d'estat, *Nb* al nombre de columnes per bloc, *Nr* al nombre de rondes i *w* a la clau expandida.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
    begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w[0, Nb-1])

    for round = 1 step 1 to Nr-1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

    out = state
    end
```

B.2 Algoritme d'expansió de la clau

Anomenem *key* a la clau donada com a dada, *w* a la clau expandida, *Nk* al nombre de columnes de la clau original, *Rcon* al vector de constants, *SubWord* a l'aplicació de la caixa S i *RotWord* a una rotació cíclica dels bytes d'una paraula.

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
  begin

    word temp

    i = 0

    while (i < Nk)
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while

    i = Nk

    while (i < Nb * (Nr+1)]
        temp = w[i-1]
        if (i mod Nk = 0)
            temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
        else if (Nk > 6 and i mod Nk = 4)
            temp = SubWord(temp)
        end if
        w[i] = w[i-Nk] xor temp
        i = i + 1
    end while
  end
```

B.3 Algoritme Rijndael de desxifratge

Utilitzant la mateixa notació que a la primera secció d'aquest apèndix i anomenant *dw* a la clau expandida aplicant-li la funció *InvMixColumns* obtenim els següents.

B.3.1 Algoritme intuitiu

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
  begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

    for round = Nr-1 step -1 downto 1
        InvShiftRows(state)
        InvSubBytes(state)
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
        InvMixColumns(state)
    end for
```

```
    InvShiftRows(state)
    InvSubBytes(state)
    AddRoundKey(state, w[0, Nb-1])

    out = state
  end
```

B.3.2 Algoritme alternatiu

```
EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
  begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, dw[Nr*Nb, (Nr+1)*Nb-1])

    for round = Nr-1 step -1 downto 1
        InvSubBytes(state)
        InvShiftRows(state)
        InvMixColumns(state)
        AddRoundKey(state, dw[round*Nb, (round+1)*Nb-1])
    end for

    InvSubBytes(state)
    InvShiftRows(state)
    AddRoundKey(state, dw[0, Nb-1])

    out = state
  end
```

Per a aquest desxifratge alternatiu, al final de la rutina d'expansió de la clau s'afegeix el pseudocodi següent.

```
for i = 0 step 1 to (Nr+1)*Nb-1
    dw[i] = w[i]
end for

for round = 1 step 1 to Nr-1
    InvMixColumns(dw[round*Nb, (round+1)*Nb-1])
end for
```