Onderzoek handmatig AES toepassen

Naam: Bart Janssen
Studentennummer: 389419
Datum: 08-01-2021

Inleiding

Het AES-algoritme is een algoritme wat gebruikt wordt om bestanden en ander dataverkeer te versleutelen zodat de integriteit van de data bewaard blijft. Het AES-algoritme is een complex algoritme en daarom is het een uitdaging om verder onderzoek te doen naar hoe dit algoritme inhoudelijk werkt en waarom dit zo krachtig is. In de minor van de cyber security opleiding van Fontys hogeschool te Eindhoven zijn er verschillende uitdagingen opgesteld om de kennis van studenten op het gebied van cyber security te verbreden. Betreft het onderwerp AES zijn er twee uitdagingen geformuleerd. Eén om het algoritme in een programmeertaal naar keuze te implementeren, de andere om dit algoritme beter te leren kennen en de stappen handmatig toe te passen. Dit onderzoek betreft alleen het onderzoeken van het handmatig toepassen.

Inhoudsopgave

1 Probleemstelling

Het AES-algoritme bestaat uit verschillende stappen om tot het versleutelde resultaat te komen. Al deze stappen hebben een eigen werking waar ook weer een verschillende moeilijkheidsgraad aan vast zit. Het is dan de vraag hoe deze stappen handmatig toegepast kunnen worden voor educatieve doeleinden.

Het onderzoek is relevant voor de minor van de cyber security opleiding van Fontys hogeschool te Eindhoven omdat het toepassen van AES een leeronderdeel is. Om dit leeronderdeel te kunnen toepassen, is het wel belangrijk om te weten welke manieren geschikt zijn om het AES-algoritme handmatig toe te kunnen passen. Er zijn meerdere manieren om het AES-algoritme toe te passen en te bestuderen, alleen het zou mogelijk kunnen zijn dat de ene manier makkelijker is dan een andere manier.

2 Doelstelling

Het doel van dit onderzoek is een manier te vinden hoe het AES-algoritme handmatig toegepast kan worden, zodat hiervan geleerd kan worden en zodat de uitdaging op een goede manier uitgevoerd kan worden. Waarmee het belangrijkste doel, het leren en beter begrijpen van de inhoudelijke werking van het algoritme is. Het op te leveren product is een aanbeveling zijn zodat mensen die dit willen leren, beter een idee hebben van waar ze zouden moeten beginnen mocht iemand hier meer kennis van willen opdoen. Ook wordt er onderzocht of er verschillende manieren zijn om tot hetzelfde resultaat te komen en om in kaart te brengen welke gemakkelijker is om mee te beginnen. Dit onderzoek is puur gericht op educatieve ontwikkeling.

3 Onderzoeksmethoden

Binnen dit onderzoek wordt gebruik gemaakt van een hoofdvraag met deelvragen. De deelvragen worden onderzocht en beantwoord om de hoofdvraag te kunnen beantwoorden. Per deelvraag wordt er onderzoek gedaan om dat specifieke gedeelte van die deelvraag te kunnen onderzoeken en beantwoorden. De deelvragen worden onderzocht met de onderzoeksmethoden uit het DOT-framework, deze bestaan uit de Field, Library, Workshop, Lab en Showroom strategieën.

- De **Field** strategie kan gebruikt worden om te onderzoeken wie of wat met dit onderzoek te maken heeft en zo een inzicht te krijgen wat voor deze groep/persoon belangrijk is of wat er nodig is en waarom dit product van belang is;
- De **Library** strategie kan gebruikt worden om te onderzoeken wat al gedaan is qua dit onderwerp en of hier al iets van hergebruikt kan worden, of juist nieuwe technieken van kunnen worden geleerd. Dit wordt meestal gedaan met behulp van bijvoorbeeld een internet methode;
- De **Lab** strategie is bedoeld om bijvoorbeeld een concept of prototype van het onderwerp te testen, hieruit kan dan blijken dat er bijvoorbeeld nog veranderingen nodig kunnen zijn;
- De **Workshop** strategie wordt gebruikt om de ideeën uit te werken in een werkend prototype. Hieruit kan dan weer blijken dat er misschien veranderingen nodig zijn;
- De **Showroom** strategie wordt gebruikt om je ideeën te laten zien. Dit kan ook een prototype zijn zodat iemand zoals de klant een beeld krijgt van hoe een prototype eruit gaat zien of van hoe het gaat werken.

4 Hoofd- en deelvragen

De onderzoeksvragen zijn opgedeeld in een hoofdvraag met deelvragen. De hoofdvraag bevat het project in één zin. Met het onderzoeken en beantwoorden van de deelvragen, wordt de hoofdvraag beantwoordt.

4.1 Hoofdvraag

- Welke manieren zijn geschikt om het AES-algoritme handmatig te kunnen toepassen?

4.2 Deelvragen

Hieronder staan de opgestelde deelvragen vermeld met per deelvraag beschreven welke strategieën er gebruikt gaan worden.

4.2.1 Hoe werkt het AES-algoritme?

Deze onderzoeksvraag onderzoekt hoe het AES-algoritme inhoudelijk werkt.

Onderzoekmethodes:

- Met de Library strategie kan er onderzocht worden hoe het werkt. De methodes kunnen video's hierover bestuderen of informatie op het internet zoeken zijn.

Tijdsinschatting:

- Mijn tijdsinschatting voor deze deelvraag zou op ongeveer een dag komen.

4.2.2 Welke voorbeelden voor het handmatig toepassen bestaan er al?

Deze deelvraag worden al bestaande voorbeelden onderzocht.

Onderzoekmethodes:

- Met de Library strategie kan er onderzocht worden welke voorbeelden er al bestaan. De methodes hiervoor zullen voornamelijk het opzoeken op internet zijn om bestaande voorbeelden te kunnen onderzoeken en deze te begrijpen.

Tijdsinschatting:

- Mijn tijdsinschatting voor deze deelvraag zou een tot twee dagen zijn.

4.3.3 Wat voor methodes kunnen gebruikt worden om het toe te passen?

Deze deelvraag gaat in op de methodes die gebruikt kunnen worden.

Onderzoekmethodes:

- Met de Workshop strategie kunnen prototypes gemaakt worden om te onderzoeken welke manier het meest geschikt is. De methodes zullen afgangen van de voorbeelden.

Tijdschatting:

- Mijn tijdsinschatting voor deze deelvraag gaat naar een week.

5 Het onderzoek

De deelvragen beschreven in hoofdstuk 4.2, worden in dit hoofdstuk onderzocht. De hoofdvraag die eerder geformuleerd is, is gebaseerd op de probleemstelling om als resultaat een oplossing te kunnen vinden voor het probleem.

5.1 Hoe werkt het AES-algoritme?

Deze deelvraag betreft hoe het AES-algoritme zelf inhoudelijk werkt. Binnen deze deelvraag wordt er dieper ingegaan op de werking ervan en welke stappen nodig zijn om de data correct te kunnen versleutelen en decoderen. Dit hoofdstuk is erg technisch en er wordt zeer gedetailleerd op de stappen van het algoritme ingegaan.

5.1.1 Het algoritme

Om te beginnen heeft het AES-algoritme een sleutel nodig, deze sleutel is te vergelijken met een wachtwoord waarmee het mogelijk is om de data te versleutelen en decoderen. De sleutel kan variëren tussen een 128, 192 of 256 bit sleutel. Een 256 bit sleutel wordt tegenwoordig aangeraden. Eén byte is 8 bit, dus 256 / 8 zijn 32 bytes. Eén karakter is gelijk aan één byte, een 256 bit sleutel is dus 32 karakters lang.

Het algoritme bestaat uit vier stappen waarvan één op basis van de sleutel lengte herhaaldelijk uitgevoerd wordt. Bij een lengte van 128 bit word deze 10 keer herhaald, een lengte van 192 bit 12 keer en bij 256 bit 14 keer.

De te versleutelen tekst

De te versleutelen tekst wordt opgedeeld in segmenten van 128 bit (16 karakters).

Stel de te versleutelen tekst is “**Dit is een geheime tekst dat versleuteld moet worden!**”, dan wordt deze verdeeld in de volgende segmenten:

- Dit is een gehei
- me tekst dat ver
- sleuteld moet w
- orden!

Het laatste segment is geen 16 karakters (128 bit) lang, dit wordt vaak met een padding opgevuld.

Hierna worden de volgende stappen per te versleutelen tekst segment uitgevoerd.

- **KeyExpansion** – hiermee wordt de sleutel omgezet naar sub sleutels die per ronde gebruikt worden;
- **AddRoundKey** – hiermee wordt de sleutel toegevoegd aan de te versleutelen tekst;

De volgende vier stappen worden **9**, **11** of **13** keer uitgevoerd afhankelijk van de sleutellengte:

- **SubBytes** – Bij deze stap worden bytes vervangen met een byte met behulp van een opzoektabel;
- **ShiftRows** – Bij deze stap worden de rijen van een tekstsegment een plekje opgeschoven, dat per segmentrij opgeteld wordt;
- **MixColumns** – Bij deze stap worden de bytes samengevoegd met een voor gedefinieerde tabel;
- **AddRoundKey** – Bij deze stap wordt de sub sleutel weer toegevoegd.

- De laatste ronde, wat een totaal maakt van **10**, **12** of **14** rondes, bestaat uit de volgende onderdelen:
    - **SubBytes**;
    - **ShiftRows**;
    - **AddRoundKey**.

Voor het decoderen worden dezelfde stappen precies in tegenovergestelde richting herhaald, in de MixColumns stap, wordt dan ook de tabel gebruikt voor het decoderen.

5.1.2 KeyExpansion

Bij de KeyExpansion stap wordt de sleutel omgezet naar sub sleutels zodat iedere ronde een eigen sleutel heeft. Als voorbeeld wordt er een sleutel gebruikt van 128 bit, de sleutel is als volgt:
GR96dzKYZP73hbUG

Deze sleutel kan als hexadecimaal geschreven worden als:
47 52 39 36 64 7A 4B 59 5A 50 37 33 68 62 55 47

Deze hexadecimale waardes kunnen als bytes vernoemd worden met een $\mathbf{b_x}$ notatie als:
$\mathbf{b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9\ b_{10}\ b_{11}\ b_{12}\ b_{13}\ b_{14}\ b_{15}\ b_{16}}$

AES gebruikt voor bijna alle stappen een grit. In een grit zou de sleutel er als volgt uitzien:

b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}
b_4	b_8	b_{12}	b_{16}

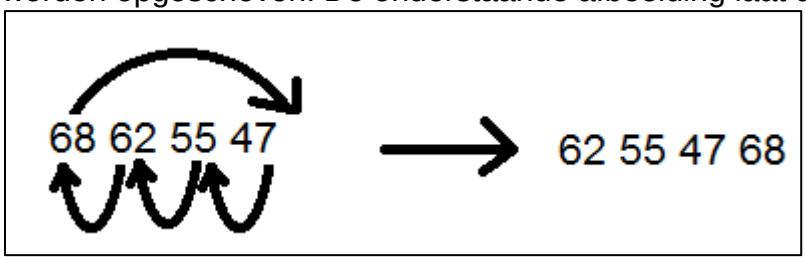
Iedere verticale rij wordt een ‘Word’ genoemd. Deze vier rijen worden benoemd als Word0, Word1, Word2 en Word3. Deze Words worden uitbereid naar 44 Words. Ieder blokje van 4 Words wordt gebruikt om het volgende blokje te bepalen. Ieder blokje van 4 Words heet een RoundKey, Die sleutel wordt gebruikt voor die ronde in het algoritme, de onderstaande tabel is de volledige sleutel in een grit, deze wordt gebruikt om de overige 40 sub sleutels te bepalen.

W0	W1	W2	W3
47	64	5A	68
52	7A	50	62
39	4B	39	55
36	59	33	47

Om W4, W5, W6 en W7 te bepalen wordt er een functie gebruikt G(W3). Deze functie pakt de laatste Word per blokje (W3 in het eerste blokje) en voert het volgende uit:

W3
68
62
55
47

De Word hierboven is W3, dit is de laatste kolom van het eerste blokje. Deze gebruikt een circulaire een-byte left shift. Dit betekent dat alle bytes van dit blokje een plaats worden opgeschoven. De onderstaande afbeelding laat dit zien van blok W3:



Afbeelding 1 Circulaire een-byte left shift

De uitkomst hiervan heet een RotWord en wordt benoemd als X1.

X1
62
55
47
68

Deze wordt nog met een voor gedefinieerde tabel vervangen. Deze tabel heet de S-Box en ziet eruit als volgt:

AES S-Box

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

Afbeelding 2 S-Box

X1
62
55
47
68

De X1 wordt ook weer per byte in deze tabel opgezocht en vervangen. **62** wordt vervangen met **AA**, **55** met **FC**, **47** met **A0** en **68** met **45**.

De uitkomst hiervan wordt een SubWord genoemd en is gedefinieerd als Y1.

Y1
AA
FC
A0
45

Vervolgens wordt er een XOR-berekening uitgevoerd op de Y1 met een voor gedefinieerde tabel, de Round constant.

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
01	02	04	08	10	20	40	80	1B	36
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

Afbeelding 3 Round constant

Bij iedere ronde (Round) wordt er per index de round kolom gebruikt met de XOR. Ronde 1 gebruikt R1, Ronde 2 R2 enz.

Voor Y1 in de eerste ronde wordt er R1 gebruikt, voor de makkelijke leesbaarheid wordt deze in binair opgeschreven, en eronder de binaire XOR uitkomst.

Y1
AA
FC
A0
45

	AA	FC	A0	45
Y1	10101010	11111100	10100000	01000101
R1	00000001	00000000	00000000	00000000
G	10101011	11111100	10100000	01000101

De uitkomst hiervan terug naar hexadecimaal is G(W3):

G
AB
FC
A0
45

W0	W1	W2	W3
47	64	5A	68
52	7A	50	62
39	4B	39	55
36	59	33	47

Om W4, W5, W6 en W7 te bepalen, wordt het volgende gedaan:

- XOR W0 en G om W4 te bepalen;
- XOR W1 en W4 om W5 te bepalen;
- XOR W2 en W5 om W6 te bepalen;
- XOR W3 en W6 om W7 te bepalen.

W0	W1	W2	W3	W4	W5	W6	W7
47	64	5A	68	EC	88	D2	BA
52	7A	50	62	AE	D4	84	E6
39	4B	39	55	99	D2	EB	BE
36	59	33	47	73	2A	19	5E

Dit wordt herhaald tot aan de V43 met iedere keer de laatste Word kolom als input van de G functie, in dit geval G(W7). W0-W43 maakt een totaal van 44 Words.

5.1.3 AddRoundKey

Voordat de rondes beginnen, wordt er een XOR-operatie uitgevoerd op de eerste sub sleutel met de te versleutelen tekst.

De eerste ronde van de sleutel werd in hoofdstuk 5.1.2 berekend.

W0	W1	W2	W3
47	64	5A	68
52	7A	50	62
39	4B	39	55
36	59	33	47

Het eerste tekstsegment dat versleuteld moet worden wat gedefinieerd staat in hoofdstuk 5.1.2, is: "**Dit is een gehei**"
In hexadecimaal: **44 69 74 20 69 73 20 65 65 6E 20 67 65 68 65 69**

Deze tekst kan ook weer omgezet worden naar een grit.

M0	M1	M2	M3
44	69	65	65
69	73	6E	68
74	20	20	65
20	65	67	69

Per karakter (byte) wordt hier een XOR-operatie op uitgevoerd. Als voorbeeld **47** (**$W0_0$**) en **44** (**$M0_0$**).

$W0_0$	01000111
$M0_0$	01000100
$AR0_0$	00000011

De uitkomst als hexadecimaal is 03
Als deze XOR-operatie over iedere byte van de blokken wordt uitgevoerd ziet dat er uit als volgt, dit blok heeft als naam, de 'State array':

A0	A1	A2	A3
03	0D	3F	0D
3B	09	3E	0A
4D	6B	19	30
16	3C	54	2E

Met deze state array als input gaan de rondes van start, deze bestaan uit de SubBytes, ShiftRows, MixColumns en AddRoundKey stappen.

5.1.4 SubBytes

In de eerste ronde is de input voor deze stap de state array uit hoofdstuk 5.1.3:

A0	A1	A2	A3
03	0D	3F	0D
3B	09	3E	0A
4D	6B	19	30
16	3C	54	2E

Bij de overige rondes zal dit de uitkomst zijn van de laatste stap uit de vorige ronde.

In deze stap wordt dezelfde S-Box tabel gebruikt eerder benoemd in hoofdstuk 5.1.2 (Afbeelding 2), elke waarde wordt daarmee vervangen. Als resultaat de volgende tabel:

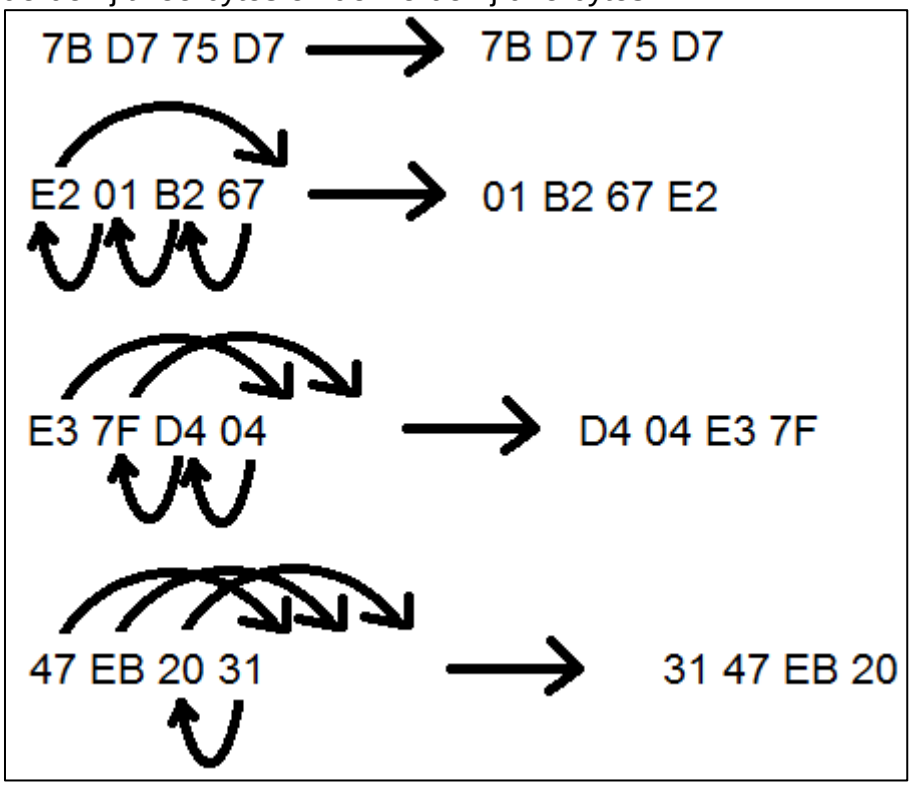
B0	B1	B2	B3
7B	D7	75	D7
E2	01	B2	67
E3	7F	D4	04
47	EB	20	31

5.1.5 ShiftRows

In deze stap wordt de uitkomst van de vorige stap SubBytes gebruikt.

B0	B1	B2	B3
7B	D7	75	D7
E2	01	B2	67
E3	7F	D4	04
47	EB	20	31

In deze stap wordt per rij een circulaire left shift operatie uitgevoerd, iedere rij krijgt een byte shift erbij. De eerste rij veranderd niet, de tweede rij schuift één byte op, de derde rij twee bytes en de vierde rij drie bytes.



Afbeelding 4 ShiftRows

Het resultaat hiervan ziet er uit als volgt:

R0	R1	R2	R3
7B	D7	75	D7
01	B2	67	E2
D4	04	E3	7F
31	47	EB	20

5.1.6 MixColumns

Deze stap neemt het resultaat van de ShiftRows stap als input.

R0	R1	R2	R3
7B	D7	75	D7
01	B2	67	E2
D4	04	E3	7F
31	47	EB	20

De MixColumns stap heeft voor het versleutelen en decoderen vaste tabellen waarmee de input vermenigvuldigd moet worden, de vastgestelde tabel voor het versleutelen ziet er uit als volgt:

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

Voor het decoderen is de vastgestelde tabel als volgt:

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

Voor dit voorbeeld wordt de tabel voor het versleutelen gebruikt.

Bij deze stap worden de rijen van de tabel voor het versleutelen, vermenigvuldigt met de kolommen van de input tabel en dat bij elkaar opgeteld, vervolgens een XOR-operatie over deze waardes.

De eerste rij van de versleutel tabel is:

02	03	01	01

De eerste kolom van de input tabel is:

R0
7B
01
D4
31

Het antwoord wat hier gezocht is, is (02 * 7B) $\oplus$ (03 * 01) $\oplus$ (01 * D4) $\oplus$ (01 * 31).

Het berekenen hiervan kan worden gedaan via polynomial multiplication, dit wordt hieronder uitgelegd.

Hiervoor worden de volgende twee waardes gebruikt; De eerste byte van de eerste rij van de versleutel tabel (02) en de eerste byte van de eerste kolom van de input tabel (7B).

Deze waardes worden omgezet naar binair (7B, 01111011) en (02, 00000010), dan staan de 1'en of 0'en op een locatie in de binaire rij, deze locaties worden een polynomial genoemd en zijn als volgt:

01111011
-- + x^6 + x^5 + x^4 + x^3 + -- + x^1 + x^0

00000010
-- + -- + -- + -- + -- + -- + -- + x^1 + --

Deze hele rij let polynomial 's wordt een Galois field genoemd.

De waardes worden eerst in een grit geplaatst:

	1
0	
1	
3	
4	
5	
6	

Vervolgens worden de waardes simpelweg met elkaar opgeteld:

	1
0	1
1	2
3	4
4	5
5	6
6	7

Alle getallen die een even aantal keer voorkomen worden weggelaten, in dit geval zijn het er geen.

	1
0	1
1	2
3	4
4	5
5	6
6	7

Deze getallen worden terug geschreven naar binair op basis van de polynomial's.
x^7 x^6 x^5 x^4 x^2 x1
11110110
In hexadecimaal: F6

Als de uitkomst hiervan onder een vast gedefinieerde waarde "100011011" is, dan is dit het antwoord, als dit antwoord groter is, dan moet er een XOR-operatie over gedaan worden met deze vaste waarde tot dat deze kleiner is.

Dit was enkel de 02 * 7B van de (02 * 7B) ⊕ (03 * 01) ⊕ (01 * D4) ⊕ (01 * 31). De overige drie worden op dezelfde manier gedaan, die staan hieronder in verkorte versie.

0x03 * 0x01
(00000011 * 00000001)

	0	1
0	0	1

00000010 (0x02)

0x01 * 0xD4
(00000001 * 11010100)

	0
2	2
4	4
6	6
7	7

11010100 (0xD4)

0x01 * 0x31
(00000001 * 00110001)

	0
0	0
4	4
5	5

00110001 (0x31)

Over deze vier waardes samen moet een XOR-operatie uitgevoerd worden.

0xF6	11110110
0x02	00000010
0xD4	11010100
0x31	00110001
0x11	00010001

Het resultaat is 0x11, dit is het antwoord op het eerste vakje, al deze stappen moeten per rij gedaan worden op de kolommen.

11	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

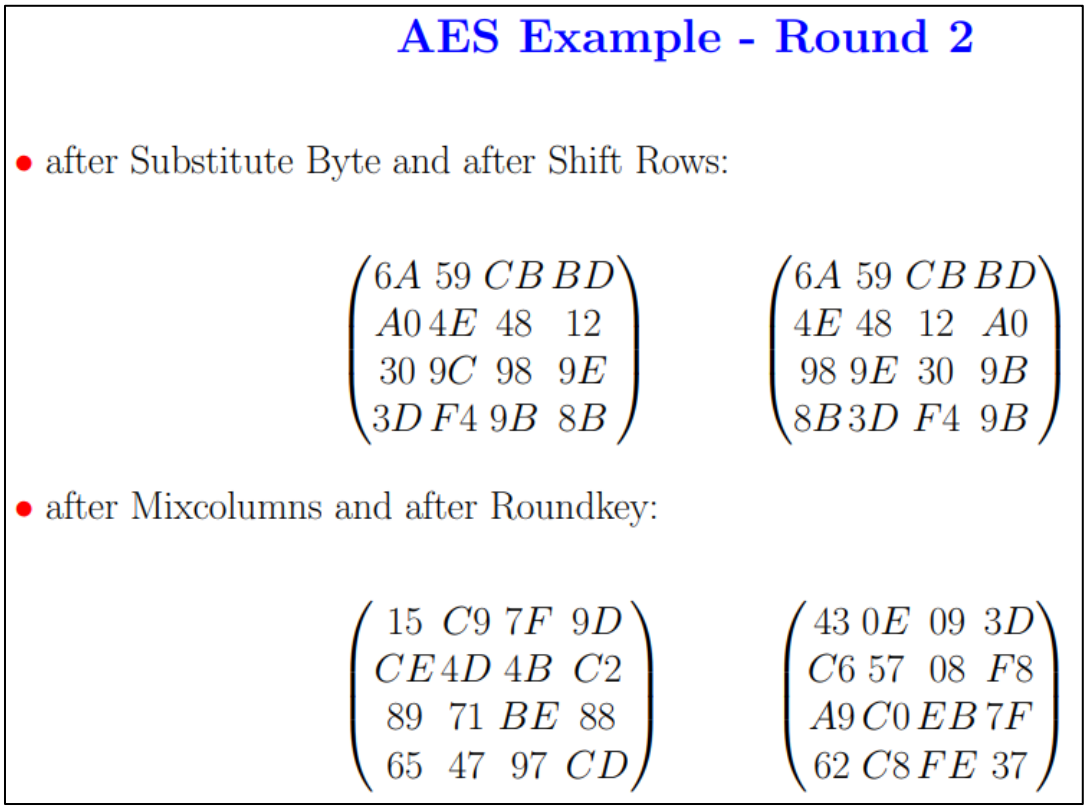
Al deze stappen worden herhaald per ronde, per te versleutelen tekstsegment. Dit voorbeeld ging over de werking per stap, het is niet relevant en erg veel werk om al de stappen door te gaan tot het uiteindelijke resultaat, dat wordt daarom ook niet uitgevoerd.

5.2 Welke voorbeelden voor het handmatig toepassen bestaan er al?

In deze deelvraag is onderzoek gedaan naar al bestaande voorbeelden voor het handmatig toepassen van het AES-algoritme.

5.2.1 Kavaliro

(Kavaliro, 2014) heeft een voorbeeld waar er eerst uitleg gegeven wordt over het AES-algoritme zelf, de uitleg is vrij abstract waardoor veel diepgaande details weggelaten zijn. Hierna wordt dit ook daadwerkelijk uitgevoerd met de uitkomst per ronde.



Afbeelding 5 Kavaliro AES-example

Zoals in de bovenstaande afbeelding te zien is, wordt hier geen verdere uitleg gegeven, de antwoorden staan wel per ronde opgeschreven zodat dit gemakkelijk nagedaan en gecontroleerd kan worden.

Voordelen

- Er is per ronde een antwoord zichtbaar, dit maakt het gemakkelijk te controleren of het gevonden antwoord klopt.

Nadelen

- Er is geen gedetailleerde uitleg over het algoritme zelf, waardoor een externe bron geraadpleegd moet worden.

5.2.2 Satish C J

(Satish C J, 2020), een professor van de Vellore Institute of Technology heeft vier video's online staan waar hij erg diep ingaat op de werking van het AES-algoritme. Niet alleen op de werking, maar ook op de achtergrond waarom bepaalde onderdelen op een manier werken en hoe deze uitgerekend kunnen worden.

Afbeelding 6 Satish C J

Hij heeft twee video's die diep ingaan op de achtergrond van welke wiskundige aspecten gebruikt worden zoals: Algebra, Group, Ring, Field en Finite field (Galois field).

De overige twee video's gaan over de stappen van het algoritme, per stap wordt er uitgelegd wat er gedaan wordt met voorbeelden die volledig uitgeschreven worden.

Voordelen

- Uitleg die precies verteld per stap wat er gebeurt en hoe bepaalde antwoorden gevonden worden;
- Als er informatie gezocht wordt op de wiskundige achtergrond.

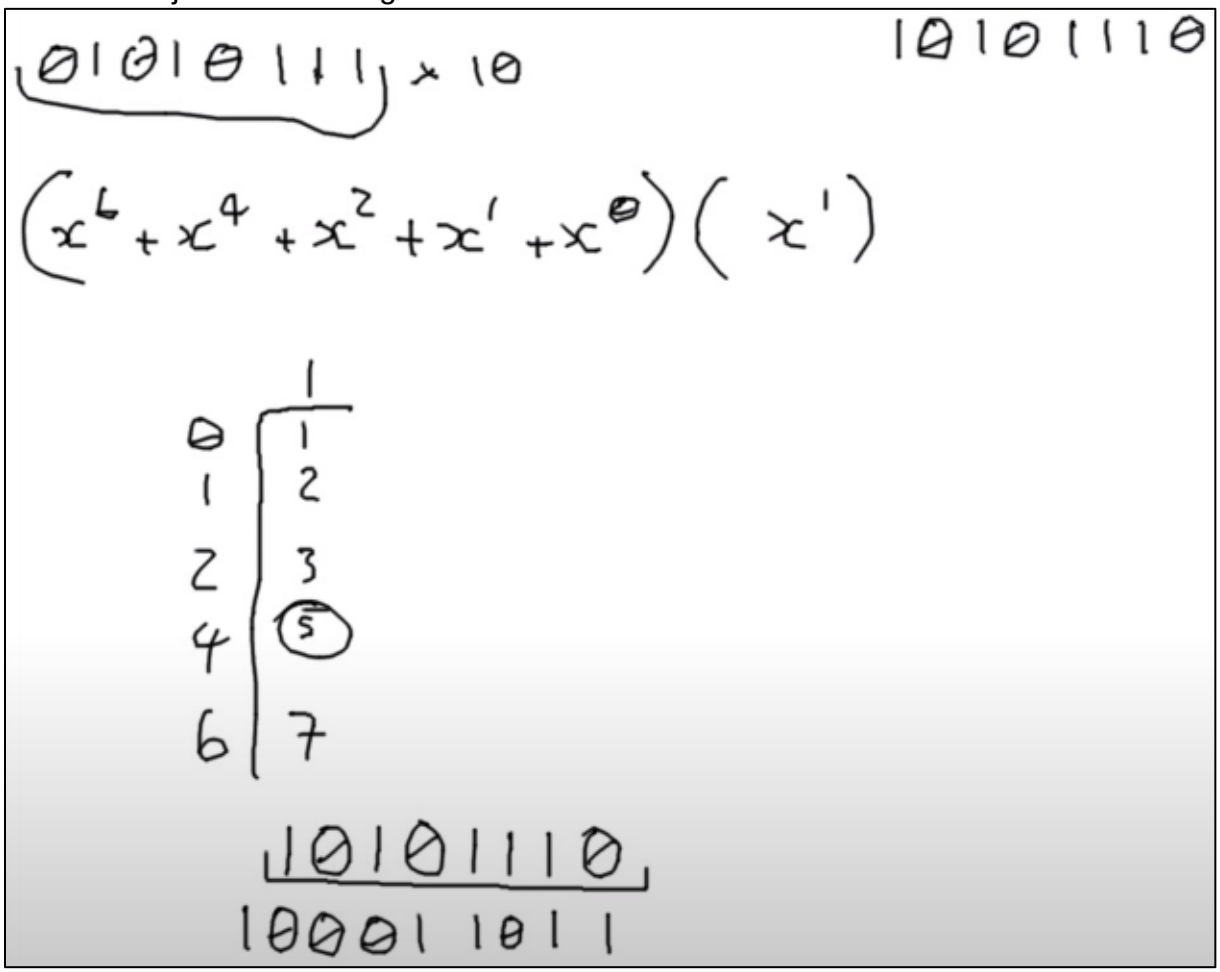
Nadelen

- Zeer gedetailleerde uitleg, dit kan voor beginners mogelijk als moeilijk worden ervaren;
- Geen volledige encryptie, alleen onderdelen om de stappen duidelijk te krijgen, hierdoor is weinig controle mogelijk of de antwoorden kloppen.

5.2.3 Creel

(Creel, 2016) is een YouTube lid die veel verschillende video's heeft over verschillende onderwerpen. AES is een van zijn onderwerpen. Hij gaat in zijn video's in op de werking per stap en legt ook uit waarom bepaalde dingen zodanig werken. Het meest opvallend is dat hij ook verschillende manieren laat zien om tot bepaalde antwoorden komen met daarbij uitleg waarom de een beter of gemakkelijker is dan de andere.

Voornamelijk de MixColumns stap uit het algoritme wordt gezien als een lastige stap. Hiervoor laat Creel verschillende manieren zien waarvan er één echt uit springt als een makkelijk en eenvoudige manier.



Afbeelding 7 MixColumns

Voordelen

- Hij laat meerdere manieren zien om tot hetzelfde resultaat te komen;
- Er worden duidelijke redenen en uitleg gegeven.

Nadelen

- Hij gebruikt veel woorden op hoog technische niveau.

5.2.4 TheSslStore

(TheSslStore, 2020) heeft een artikel gepubliceerd waar ze ingaan op de werking van AES en waarom dit gebruikt wordt. Ook wordt hier ingegaan op de stappen zelf in een minder gedetailleerd, maar meer abstracte manier. Hun laten dit meer vanuit een theoretisch en meer overzichtelijker oogpunt zien.

1. Key Expansion and AddRoundKey

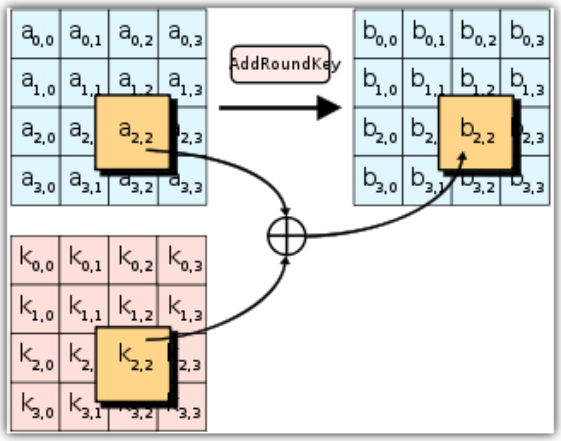


Image source:
https://en.wikipedia.org/wiki/Advanced_Encryption_Standard#Description_of_the_ciphers

As mentioned earlier, the key size determines the number of rounds of scrambling that will be performed. AES encryption uses the Rjindael Key Schedule, which derives the subkeys from the main key to perform the Key Expansion.

The AddRoundKey operation takes the current state of the data and executes the XOR Boolean operation against the current round subkey. XOR means "Exclusively Or," which will yield a result of true if the inputs differ (e.g. one input must be 1 and the other input must be 0 to be true). There will be a unique subkey per round, plus one more (which will run at the end).

Afbeelding 8 TheSslStore

Voordelen

- Duidelijk overzicht.

Nadelen

- Weinig diepgaand;
- Behoorlijk abstracte uitleg.

5.3 Wat voor methodes kunnen gebruikt worden om het toe te passen?

Er is onderzocht wat voor methodes er gebruikt kunnen worden om het AES-algoritme handmatig toe te passen. Er zijn verschillende manieren om dit te doen.

Als er gebruik gemaakt wordt van een kleinere sleutel en een eenvoudige tekst, dan zou dit prima kunnen. De stappen kunnen een voor een beschreven worden. Er kan ook gebruik gemaakt worden van een verkorte versie door niet alle rondes door te gaan, maar ervoor te kiezen om bijvoorbeeld één ronde uit te voeren. Als één ronde lukt, dan lukt 10 rondes ook. Om het complete algoritme uit te voeren zouden wel alle rondes gedaan moeten worden.

Voor een 100% complete toepassing kan het ook gebruikt worden om te decoderen, dit vereist om alle stappen in tegenovergestelde richting uit te voeren. Het eindresultaat zou dan gelijk moeten zijn aan de originele tekst.

5.3.1 Papier

Het AES-algoritme kan prima op papier uitgetekend worden. Het is eenvoudig om hier aantekeningen bij te maken en eventuele fouten te herstellen. Een nadeel is dat dit mogelijk veel papier kost als alle stappen doorgevoerd moeten worden. Een potlood zou de voorkeur hebben ten opzichte van een pen omdat het makkelijker is om potloodfouten te herstellen.

5.3.2 Computer

Het AES-algoritme op een computer toepassen werkt ook, dit kan gedaan worden door middel van verschillende manieren. Deze manieren kunnen bijvoorbeeld zijn: Een tekstbewerker, programma's zoals Paint of op andere creatieve manieren. De manier hoe maakt niet veel uit, een computer maakt het ook gemakkelijk fouten te herstellen.

5.3.3 Code

Een mogelijk goede manier om het toe te passen en te leren, zou zijn om het algoritme in code te maken. Geen libraries gebruiken die al encryptie toe kunnen passen, maar de stappen zelf implementeren. Het grotendeel van het AES-algoritme zijn XOR-operaties en bytes opzoeken en vervangen uit een tabel. Deze stappen zijn vrij eenvoudig te implementeren.

5.3.4 MixColumns stap

De MixColumns stap is voor vele een lastige stap en ook voor deze stap zijn er verschillende manieren om deze te berekenen.

Als voorbeeld worden er twee getallen gebruikt:
0xD6 * 0x36
11010110 * 00110110

Polynomial multiplication
De polynomial ‘s hiervan worden geschreven als:
$(x^7 + x^6 + x^4 + x^2 + x^1) * (x^5 + x^4 + x^2 + x^1)$

Bij polynomial multiplication worden de ‘tot de macht’ nummers bij elkaar opgeteld. Voor ieder nummer uit de eerste rij, wordt iedere macht van de tweede rij erbij opgeteld.
(7 + 5) + (7 + 4) + (7 + 2) + (7 + 1). Hierna is de 6^e polynomial aan de beurt enzovoorts.

$x^{12} + x^{11} + x^9 + x^8 +$
$x^{11} + x^{10} + x^8 + x^7 +$
$x^9 + x^8 + x^6 + x^5 +$
$x^7 + x^6 + x^4 + x^3 +$
$x^6 + x^5 + x^3 + x^2$

Polynomial multiplication MOD 2
Dit kan ook makkelijker gedaan worden door middel van een grit. Alle polynomial’s kunnen in de x- en y as van de grit geplaatst kunnen worden.

	1	2	4	5
1				
2				
4				
6				
7				

Hierna kunnen ze met elkaar opgeteld worden.

	1	2	4	5
1	2	3	5	6
2	3	4	6	7
4	5	6	8	9
6	7	8	10	11
7	8	9	11	12

Zoals te zien staan hier dezelfde nummers als de eerder uitgewerkte multiplication.
De polynomial ‘s hiervan terug geschreven naar binair is het antwoord. Maar meerdere getallen in de grit zou in binair op een 2 uitkomen. Het binaire stelsel kent geen 2, dus 2 MOD 2 is 0.

	1	2	4	5
1	**2**	~~3~~	~~5~~	**6**
2	~~3~~	**4**	**6**	~~7~~
4	~~5~~	**6**	**8**	~~9~~
6	~~7~~	**8**	**10**	~~11~~
7	**8**	~~9~~	~~11~~	**12**

$x^{12} + x^{10} + x^8 + x^6 + x^4 + x^2$
1010101010100

Opzoek tabel
Omdat het tweede getal van de voor gedefinieerde tabel altijd een 1, 2 of 3 is, zijn hier opzoektabellen voor. Deze tabellen kunnen worden gebruikt door een hexadecimale waarde als x- en y as te gebruiken voor deze tabel.

1
Als het keer 1 is, dan is het antwoord altijd hetzelfde, het is dan niet nodig om hiervoor een tabel te definiëren.

2
Voor keer 2 is de volgende tabel gedefinieerd:

```
0x00,0x02,0x04,0x06,0x08,0x0a,0x0c,0x0e,0x10,0x12,0x14,0x16,0x18,0x1a,0x1c,0x1e,
0x20,0x22,0x24,0x26,0x28,0x2a,0x2c,0x2e,0x30,0x32,0x34,0x36,0x38,0x3a,0x3c,0x3e,
0x40,0x42,0x44,0x46,0x48,0x4a,0x4c,0x4e,0x50,0x52,0x54,0x56,0x58,0x5a,0x5c,0x5e,
0x60,0x62,0x64,0x66,0x68,0x6a,0x6c,0x6e,0x70,0x72,0x74,0x76,0x78,0x7a,0x7c,0x7e,
0x80,0x82,0x84,0x86,0x88,0x8a,0x8c,0x8e,0x90,0x92,0x94,0x96,0x98,0x9a,0x9c,0x9e,
0xa0,0xa2,0xa4,0xa6,0xa8,0xaa,0xac,0xae,0xb0,0xb2,0xb4,0xb6,0xb8,0xba,0xbc,0xbe,
0xc0,0xc2,0xc4,0xc6,0xc8,0xca,0xcc,0xce,0xd0,0xd2,0xd4,0xd6,0xd8,0xda,0xdc,0xde,
0xe0,0xe2,0xe4,0xe6,0xe8,0xea,0xec,0xee,0xf0,0xf2,0xf4,0xf6,0xf8,0xfa,0xfc,0xfe,
0x1b,0x19,0x1f,0x1d,0x13,0x11,0x17,0x15,0x0b,0x09,0x0f,0x0d,0x03,0x01,0x07,0x05,
0x3b,0x39,0x3f,0x3d,0x33,0x31,0x37,0x35,0x2b,0x29,0x2f,0x2d,0x23,0x21,0x27,0x25,
0x5b,0x59,0x5f,0x5d,0x53,0x51,0x57,0x55,0x4b,0x49,0x4f,0x4d,0x43,0x41,0x47,0x45,
0x7b,0x79,0x7f,0x7d,0x73,0x71,0x77,0x75,0x6b,0x69,0x6f,0x6d,0x63,0x61,0x67,0x65,
0x9b,0x99,0x9f,0x9d,0x93,0x91,0x97,0x95,0x8b,0x89,0x8f,0x8d,0x83,0x81,0x87,0x85,
0xbb,0xb9,0xbf,0xbd,0xb3,0xb1,0xb7,0xb5,0xab,0xa9,0xaf,0xad,0xa3,0xa1,0xa7,0xa5,
0xdb,0xd9,0xdf,0xdd,0xd3,0xd1,0xd7,0xd5,0xcb,0xc9,0xcf,0xcd,0xc3,0xc1,0xc7,0xc5,
0xfb,0xf9,0xff,0xfd,0xf3,0xf1,0xf7,0xf5,0xeb,0xe9,0xef,0xed,0xe3,0xe1,0xe7,0xe5
```

Afbeelding 9 Opzoektabel Keer 2

3
Voor keer 3 is de volgende tabel gedefinieerd:

```
0x00,0x03,0x06,0x05,0x0c,0x0f,0x0a,0x09,0x18,0x1b,0x1e,0x1d,0x14,0x17,0x12,0x11,
0x30,0x33,0x36,0x35,0x3c,0x3f,0x3a,0x39,0x28,0x2b,0x2e,0x2d,0x24,0x27,0x22,0x21,
0x60,0x63,0x66,0x65,0x6c,0x6f,0x6a,0x69,0x78,0x7b,0x7e,0x7d,0x74,0x77,0x72,0x71,
0x50,0x53,0x56,0x55,0x5c,0x5f,0x5a,0x59,0x48,0x4b,0x4e,0x4d,0x44,0x47,0x42,0x41,
0xc0,0xc3,0xc6,0xc5,0xcc,0xcf,0xca,0xc9,0xd8,0xdb,0xde,0xdd,0xd4,0xd7,0xd2,0xd1,
0xf0,0xf3,0xf6,0xf5,0xfc,0xff,0xfa,0xf9,0xe8,0xeb,0xee,0xed,0xe4,0xe7,0xe2,0xe1,
0xa0,0xa3,0xa6,0xa5,0xac,0xaf,0xaa,0xa9,0xb8,0xbb,0xbe,0xbd,0xb4,0xb7,0xb2,0xb1,
0x90,0x93,0x96,0x95,0x9c,0x9f,0x9a,0x99,0x88,0x8b,0x8e,0x8d,0x84,0x87,0x82,0x81,
0x9b,0x98,0x9d,0x9e,0x97,0x94,0x91,0x92,0x83,0x80,0x85,0x86,0x8f,0x8c,0x89,0x8a,
0xab,0xa8,0xad,0xae,0xa7,0xa4,0xa1,0xa2,0xb3,0xb0,0xb5,0xb6,0xbf,0xbc,0xb9,0xba,
0xfb,0xf8,0xfd,0xfe,0xf7,0xf4,0xf1,0xf2,0xe3,0xe0,0xe5,0xe6,0xef,0xec,0xe9,0xea,
0xcb,0xc8,0xcd,0xce,0xc7,0xc4,0xc1,0xc2,0xd3,0xd0,0xd5,0xd6,0xdf,0xdc,0xd9,0xda,
0x5b,0x58,0x5d,0x5e,0x57,0x54,0x51,0x52,0x43,0x40,0x45,0x46,0x4f,0x4c,0x49,0x4a,
0x6b,0x68,0x6d,0x6e,0x67,0x64,0x61,0x62,0x73,0x70,0x75,0x76,0x7f,0x7c,0x79,0x7a,
0x3b,0x38,0x3d,0x3e,0x37,0x34,0x31,0x32,0x23,0x20,0x25,0x26,0x2f,0x2c,0x29,0x2a,
0x0b,0x08,0x0d,0x0e,0x07,0x04,0x01,0x02,0x13,0x10,0x15,0x16,0x1f,0x1c,0x19,0x1a
```

Afbeelding 10 Opzoektabel Keer 3

6 Conclusie

Het AES-algoritme is een uitgebreid en ingewikkeld algoritme. Het is onderverdeeld in verschillende stappen met ieder een eigen moeilijkheidsgraad en werking. Voor het handmatig toepassen van het AES-algoritme is basiskennis van computertechnologie aangeraden, er wordt veel uitgevoerd op bit en byte niveau.

Er zijn al verschillende online voorbeelden gemaakt waarin dit algoritme wordt uitgelegd. Veel video's zijn verschillende van elkaar, sommige video's gaan dieper in op de stappen zelf, andere hebben meer abstractere uitleg. Het hangt er vanaf in hoeverre iemand het algoritme wil bestuderen en toepassen welke video het best geschikt is.

Voor de manier van toepassen kunnen verschillende manieren gebruikt worden, enkele manieren zijn met pen en papier, op een tekstbewerker op een computer. De voordelen van deze manieren zijn dat fouten gemakkelijk hersteld kunnen worden. Een andere manier zou zijn om het algoritme in code te maken. Hierbij is het dan belangrijk dat er geen libraries gebruikt worden die dit al doen, maar alle stappen zelf implementeren. Dit geeft goede inhoudelijke kennis van het algoritme en geeft ook de mogelijkheid om het uit te voeren en de resultaten stap voor stap te bekijken.

De MixColumns stap is de lastigste stap uit het algoritme, deze kan op verschillende manieren bereken worden, dit kan met Polynomial multiplication (MOD 2) en via een opzoektabel.

7 Aanbeveling

Om het AES-algoritme toe te passen is het verstandig om alle stappen eerst te bestuderen. Deze kunnen allemaal uitgewerkt worden. Het is mogelijk om een makkelijkere versie ervan uit te voeren, hiervoor kunnen er bijvoorbeeld één of twee rondes berekend worden. Als één of twee rondes lukt, dan is het niets anders dan herhaling.

Om het algoritme goed te begrijpen en toe te passen zouden alle stappen en rondes uitgevoerd moeten worden. Het decoderen hoort daar ook bij, het eindresultaat van het decoderen zou gelijk moeten zijn aan de tekst die versleuteld moest worden. Is dit niet het geval dan zitten er nog fouten in.

Een erg goede toevoeging zou zijn om het algoritme zelf in code te implementeren, hiermee is het mogelijk om per stap te kunnen zien wat de uitkomst is en een computer heeft de herhaling van rondes vaak sneller uitgevoerd als een persoon. In code is het ook mogelijk om te debuggen, wat kan helpen om te achterhalen waar een mogelijke fout zit.

8 Planning

Hieronder staat de globale planning van dit onderzoek:

Deelvraag	**Tijdsinschatting in dagen**
Hoe werkt het AES-algoritme?	1-2
Welke voorbeelden voor het handmatig toepassen bestaan er al?	1-2
Wat voor methodes kunnen gebruikt worden om het toe te passen?	1-7
Totaal	**3-11**

Tabel 1 Globale planning

Literatuurlijst

Creel. (2016, 20 januari). *'AES Encryption 3: MixColumns 1 Dot Products'* Geraadpleegd op 7 januari 2021 van https://www.youtube.com/watch?v=dRYHSf5A4lw&t=1758s

Kavaliro. (2014, maart). *'AES Example - Input (128 bit key and message)'.* Geraadpleegd op 7 januari 2021 van https://kavaliro.com/wp-content/uploads/2014/03/AES.pdf

Satish C J. (2020, 26 augustus). *'AES III - Advanced Encryption Standard - Introduction'.* Geraadpleegd op 7 januari 2021 van https://www.youtube.com/watch?v=w4aWIVhcUyo

Satish C J. (2020, 26 augustus). *'AES IV - Advanced Encryption Standard - Encryption and Decryption'.* Geraadpleegd op 7 januari 2021 van https://www.youtube.com/watch?v=5PHMbGr8eOA

TheSslStore. (2020, 23 april). *'Advanced Encryption Standard (AES): What It Is and How It Works'.* Geraadpleegd op 7 januari 2021 van https://www.thesslstore.com/blog/advanced-encryption-standard-aes-what-it-is-and-how-it-works/

Afkortingen en woordenlijst

AES	Advanced Encryption Standard
MOD	Modulus
XOR	Exclusieve disjunctie
⊕	XOR-teken
Bit	Waarde uit het binaire stelsel, een 1 of 0