

Profielwerkstuk Wiskunde 5havo Havo

Alex Dekkers

Luca Keijzers

Noud van Zwam

# Voorwoord

Wiskunde is een uitdagend vak voor het maken van een profielwerkstuk. We hebben lang gezocht naar een geschikt onderwerp. Wij zochten een onderwerp waarbij wiskunde in verband kan worden gebracht en kan worden gebruikt om hedendaagse problemen uit de maatschappij comfortabel en eenvoudiger te maken. Wiskunde is immers een vak met veel afgebakende aangelegenheden, waardoor het moeilijk is om een goede samenhang te vinden tussen het onderzoek en de wiskundige aspecten. Na veel oriëntatie tijdens de voorbereidende opdracht is er uiteindelijk een lijst ontstaan met eventuele onderwerpen. Zoals verwacht lagen de onderwerpen ver uit elkaar: van oneindigheid tot kansberekening. Samen met onze begeleider zijn we gaan kijken naar het meest voordehand liggende onderwerp. Zo doende zijn andere onderwerpen weggevallen en er is een definitief onderwerp vastgesteld, “de Coderingstheorie”. Dit spreekt ons erg aan, omdat iedereen op de wereld er elke dag gebruik van maakt en het een groot deel uitmaakt van ons digitale leven. Dit spreekt ons ook erg aan omdat dit in de toekomst nog veel meer voor ons gaat betekenen.

## Dankwoord

Ten eerste willen we iedereen bedanken die onze proef heeft gemaakt. Hierdoor hebben wij redelijk nauwkeurige resultaten gekregen waar we veel mee konden. Ook willen we Jan Coppens bedanken voor het begeleiden van ons groepje. Hij is erg kritisch geweest en dit heeft ons zeker geholpen. Ook heeft hij handige tips gegeven hoe we ons profielwerkstuk vorm konden geven.

# Hoofdvraag:

Wat houdt de coderingstheorie in?

## Deelvragen:

Wie heeft de coderingstheorie bedacht?

Wat hebben we aan coderingstheorie in de praktijk?

Wat is cryptografie?

Wat heeft cryptografie met coderingstheorie te maken?

# Samenvatting

‘’Wat houdt coderingstheorie in?’’ Dat was de vraag waarmee wij ons profielwerkstuk begonnen. Het is een ontzettend breed onderwerp waar iedereen dagelijks mee te maken heeft maar het niet door heeft. Bij het ontstaan van de eerste computers ontstonden ook de eerste coderingstheorieën. Tot op de dag van vandaag worden coderingstheorieën gebruikt en ze gaan ook nog heel veel voor ons betekenen. Het doorsturen van informatie zijn technieken voor nodig. Computers werken met binaire cijfers 0 en 1. Informatie moet dus worden omgezet in een keten van nullen en enen anders kan het niet verstuurd worden. Dit kan op verschillende manieren. Maar helaas treedt er tijdens het versturen ruis op daarom moet er aan een keten een aantal beveiligingstekens toe worden gevoegd om ervoor te zorgen dat de informatie juist ontcijferd wordt.

Inhoud

[Voorwoord 3](#_Toc504032377)

[Dankwoord 3](#_Toc504032378)

[Hoofdvraag: 3](#_Toc504032379)

[Deelvragen: 3](#_Toc504032380)

[Samenvatting 4](#_Toc504032381)

[1 Oorsprong 7](#_Toc504032382)

[1.1 Wie heeft de coderingstheorie bedacht? 7](#_Toc504032383)

[1.1.1 Claude Shannon (1916-2001) 7](#_Toc504032384)

[1.1.2 Eerste theorieën 8](#_Toc504032385)

[1.1.3 Marcel J.E. Golay (1902-1989) 11](#_Toc504032386)

[2 Wat hebben we aan coderingstheorie in de praktijk? 12](#_Toc504032387)

[3 Soorten 13](#_Toc504032388)

[3.0 Welke soorten coderingstheorieën bestaan er? 13](#_Toc504032389)

[3.1 Herhalingscode 14](#_Toc504032390)

[3.2 Hamming Code 14](#_Toc504032391)

[3.3 Lineaire Codes 16](#_Toc504032392)

[3.3.1 Blokcode 16](#_Toc504032393)

[4 Beveiliging 18](#_Toc504032394)

[4.1 Wat is cryptografie? 18](#_Toc504032395)

[Caesarcryptografie 19](#_Toc504032396)

[Affiene cryptografie 20](#_Toc504032397)

[Monoalfabetische substitutie 20](#_Toc504032398)

[Vigenère 21](#_Toc504032399)

[4.2 Wat heeft cryptologie met coderingstheorie te maken? 23](#_Toc504032400)

[5 Ethische Paragraaf 25](#_Toc504032401)

[Casus 25](#_Toc504032402)

[Achtergrondinformatie 25](#_Toc504032403)

[Belanghebbende 26](#_Toc504032404)

[Ethische vraag 26](#_Toc504032405)

[Dilemma 27](#_Toc504032406)

[Afweging en Oordeel 27](#_Toc504032407)

[Afweging 27](#_Toc504032408)

[Oordeel 30](#_Toc504032409)

[Norm 30](#_Toc504032410)

[Eigen mening 31](#_Toc504032411)

[Noud 31](#_Toc504032412)

[Alex 31](#_Toc504032413)

[Luca 31](#_Toc504032414)

[6 Onderzoek 31](#_Toc504032415)

[Onderzoeksvraag: Zijn crypto-techtnieken makkelijk te ontcijferen? 31](#_Toc504032416)

[Lob paragraaf 36](#_Toc504032417)

[Conclusie 37](#_Toc504032418)

[Reflectie 37](#_Toc504032419)

[Noud 37](#_Toc504032420)

[Luca 37](#_Toc504032421)

[Alex 37](#_Toc504032422)

[7 Bijlagen 38](#_Toc504032423)

[7.1 Hoofdstuk 1 38](#_Toc504032424)

[7.2 Hoofdstuk 2 38](#_Toc504032425)

[7.3 Ethische Paragraaf 38](#_Toc504032426)

[8 Bibliografie 39](#_Toc504032427)

[9 Logboek 40](#_Toc504032428)

# 1 Oorsprong

Op 4 december 1997 lanceerde het Amerikaanse ruimtestation NASA het meetstation de “Pathfinder”. Het doel van de missie was om de metingen van de atmosfeer die tijdens het landen waren genomen door de sturen naar de aarde en diende als ondersteuning van de “Sojourner”. Dit is een kleine robotverkenner die het meetstation met zich meebracht. Al decennialang sturen ruimtestations informatie door naar de aarde en het internet staat vol met beeldmateriaal van andere planeten en zonnestelsels. Echter is de kracht van die zenders slechts een paar watt. Dit is te vergelijken met een gemiddelde gloeilamp. Hoe kan deze informatie dan toch worden overgedragen over honderden miljoenen kilometers zonder dat het signaal wordt beïnvloed van buitenaf?

Figuur 1 Robotverkenner 'Sojourner'

## 1.1 Wie heeft de coderingstheorie bedacht?

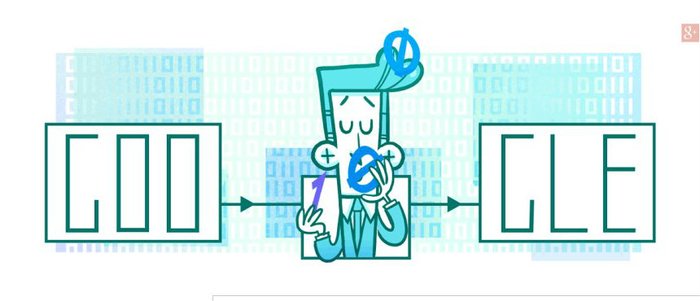
Het ontstaan van de coderingstheorie kan niet onder een naam worden benoemd. Het proces van de ontwikkeling is een fenomeen wat tot de dag van vandaag nog steeds gaande is. De drang naar het toevoegen van redundantie aan gecodeerde informatie is samen met de ontwikkeling van de digitale wereld opgegroeid. Zonder digitale informatie en signalen zou de coderingstheorie niet van toepassing zijn. Immers blijkt uit de geschiedenis dat er een aantal belangrijke grondleggers zijn die de eerste steentjes tot de digitale wereld zoals die vandaag bestaat, hebben gelegd.

### 1.1.1 Claude Shannon (1916-2001)

Claude Shannon was een Amerikaanse wiskundige en elektrotechnicus en studeerde aan de universiteit van Michigan. Hij bouwde bijvoorbeeld in de jaren zeventig een jonglerende robot. Het ging verder als alleen jongleren voor de grap: Shannon bedacht zelfs een eigen formule voor het jongleren:

(F + D)H = (V + D)N

Hierin bracht hij de tijd dat een bal in de lucht blijft ***F*** in verband met de tijd dat een bal in de hand blijft ***D*** en vermenigvuldigde hij de som van deze twee met het aantal handen ***H***. Dit stond gelijk aan de som van de tijd dat een hand leeg ***V*** is en de tijd dat een bal in de hand blijft ***D*** vermenigvuldigd met het aantal ballen ***N***.

Toen Shannon in 2001 stierf toonde Google een eerbetoon aan hem aan de hand van een Doodle (een speciaal logo op de voorpagina van Google). Hierop was Shannon jonglerend te zien.

Figuur 2 Doodle van Claude Shannon

Shannon zag logica en circuits gekruist tot nieuwe mogelijkheden net als codes en genen. Op zijn identieke manier, op zoek naar een kader om al zijn kabels te verbinden, begon Shannon een theorie voor informatie te verzamelen. Hij vond dat *informatie* iets uitzonderlijks moest betekenen voor de doeleinden van de wetenschap. Zijn grote voorbeeld was Isaac Newton (1642-1727) die drie eeuwen eerder de natuurkundige wetenschap in een nieuw daglicht had gezet. Oude en vage begrippen werden op de schop gegooid en kregen nieuwe betekenissen. Kracht, massa, beweging en tijd werden in eenheden gemaakt, geschikt voor gebruik in de wiskunde.

Shannon zocht een manier om informatie te ‘meten’, hiervoor koos hij gebruik te maken van logaritmische basis. Deze worden *binary digits* genoemd. Kortweg, bits. Een apparaat bijvoorbeeld met twee mogelijke uitgangen - zoals een relais of flip-flop circuit – kan een *bit* informatie opslaan. Dit betekent dat *N* van zulke apparaten *N* bits kan opslaan, aangezien het aantal mogelijke uitgangen 2*N*is en log22*N* = *N*. Als grondtal 10 wordt gebruikt, zouden de eenheden decimaal moeten zijn aangezien

log2*M*  = log10*M* / log102

= 3.32log10*M*

### 1.1.2 Eerste theorieën

#### 1.1.2.1 Richard Hamming (1915-1998)

Richard Wesley Hamming was een Amerikaanse wiskundige. Tot zijn werk behoorde de Hammingcode, Hammingwindow en de Hammingafstand. Deze werken hebben enorme invloed gehad op de informatica en telecommunicatie.

Hamming kreeg een licentie van de Universiteit van Chicago, een mastersdegree aan de Universiteit van Newbraska en daarna zijn Doctoraat van de Universiteit van Illinois. Aan het begin van de Tweede Wereldoorlog was Hamming een professor aan de Universiteit van Louisville. Hij besloot zijn baan op te geven en mee te werken aan het Manhattan Project in 1945. Het Manhattan Project is het plannen van een van de eerste digitale computers om de oplossingen van de vergelijkingen die door de fysici van het project te berekenen. Met als doel erachter te komen of de ontploffing van een kernwapen kon vuren in de atmosfeer.

Tussen 1946 en 1976 werkte Hamming samen met Claude Shannon bij Bell Labs of Bell Telephone Laboratories, oorspronkelijk een onderzoeks- en ontwikkelingslaboratorium van het Amerikaanse nationale telefoonbedrijf Bell System. Hamming en Shannon behandelde onderwerpen zoals wiskunde, natuurkunde, informatietheorie, materiaalkunde, gedragswetenschappen, en informatica.

De Hamming-codes zijn lineaire codes, en kunnen 1 of 2 bit fouten detecteren. Omdat deze code nog steeds gebruikt wordt zal het later in het hoofdstuk SOORTEN uitgebreid aan bod komen. Een aantal fout detecterende codes waren al voor Hamming in gebruik, echter waren deze bij gelijke redundanties veel minder effectief.

#### 1.1.2.2 Pariteitsbit

Een voorbeeld van een minder effectieve code is de pariteit bit. Onder de pariteit van een bitreeks verstaan het even of oneven zijn van het aantal enen in die bitreeks. Als voorbeeld voegen we een pariteitsbit toe aan een bitreeks. Werken we met even pariteit, dan maakt het pariteitsbit de hoeveelheid enen in de bitreeks even. In het voorbeeld is een pariteitsbit aan een zestal bytes toegevoegd.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

Dit kan worden gebruikt om enkelvoudige fouten te ontdekken. Als er een reeks met oneven aantal enen wordt ontvangen, moet er iets mis zijn gegaan. Alleen is dan nog niet duidelijk welke bit fout ontvangen wordt. In het volgende voorbeeld wordt geïllustreerd hoe het optreden van een fout wordt opgemerkt. Het is hierbij natuurlijk wel van belang dat zender en ontvanger afstemmen of er met even of oneven pariteit gewerkt wordt.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

Naast horizontale detectie is er ook verticale detectie mogelijk. Dat betekent dat in zowel de horizontale als in de verticale reeks een even aantal enen voorkomen.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Nu kan er een tweedimensionale check worden uitgevoerd. Daarbij kan een fout niet alleen worden gevonden maar ook worden verbeterd. Dit wordt in het volgende voorbeeld verduidelijkt.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Bij een tweedimensionale pariteitscheck kunnen ook fouten worden ontdekt als meerdere bits in een rij fout zijn. Dit is handig bij bijvoorbeeld *bursts*, verstoringen met fouten die meestal meerdere keren achter elkaar voorkomen.

### 1.1.3 Marcel J.E. Golay (1902-1989)

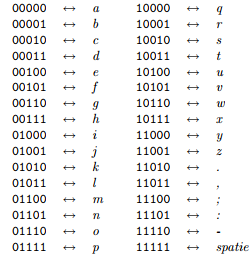
Marcel J.E. Golay was Zwitserse wiskundige en informatietheoreticus. Hij paste wiskunde toe op praktische militaire en industriële problemen. Golay was bekend om zijn uitvinding van de Golay-codes. Zijn theorie is te vergelijken met een invulformulier van voetbalwedstrijden. Stel dat er elf wedstrijden plaatsvinden met drie mogelijke uitkomsten. Winst, verlies of gelijkspel. Dat betekent dat een ingevuld formulier is een woord van lengte elf in drie letters, waarvoor we F3 = {0, 1, 2} nemen. Dat geeft 311 mogelijke uitkomsten van het formulier. In de elf dimensionale ruimte over F3 ligt nu de Golay code. Deze code bestaat uit 36 = 729 woorden en wordt uitgedrukt door de matrix *G*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 2 |
| 0 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 1 | 2 |
| 0 | 0 | 0 | 0 | 1 | 0 | 2 | 2 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 1 | 0 |

Ga ervan uit dat de minimumafstand tussen de woorden is d = 5, dit maakt de code twee-fouten-verbeterend. Hieruit volgt 35 = 243. Er geldt 729 x 243 = 311, dus de codewoorden overdekken de gehele ruimte. Je hoeft bij het wedstrijdformulier dus maar 729 opties in te vullen om zeker te zijn dat je op een formulier hoogstens twee fouten had. Op deze manier is veel geld verdient, totdat de directieleden hierachter kwamen en het systeem veranderden.

# 2 Wat hebben we aan coderingstheorie in de praktijk?

Als je informatie opslaat wordt dit omgezet in een binaire code van 0 en 1 hieronder is schematisch weergegeven hoe dat werkt.

Een normaal woord wordt beschreven in nullen en enen. Hieraan worden een paar extra tekens toegevoegd als beveiliging. Bij het verzenden gaat er vaak iets mis dit is de verstoring. Maar door de extra beveiliging wordt dit juist. Om normale woorden om te zetten in nullen en enen is een codelijst nodig. Een code lijst kan er heel simpel uit zien bijvoorbeeld:

Als je dus een berichtje verstuurt of opslaat wordt het berichtje omgezet in een lange keten van enen en nullen. Er is gekozen voor het gebruik van nullen en enen omdat je in computers op het meest elementaire niveau maar met twee toestanden kunt werken: ‘stroom aan’ of ‘stroom uit’. Als het codewoord verstuurd wordt treedt er ruis op daarom worden er code tekens toegevoegd. Ook wel coderen genoemd. Dit wordt gedaan zodat er met het decoderen geen oneffenheden kunnen optreden. Als dit niet gedaan zou worden is de kans erg groot dat jij je ontvangen berichtje niet begrijpt, omdat er een aantal foute letters tussen staan. Coderingstheorie vergemakkelijkt ons leven dus heel erg ook al weten veel mensen niet wat het inhoud. Tevens gebruikt men het wel elke dag. Wiskunde staat in sterk verband met de coderingstheorie, omdat er verschillende manieren zijn bedacht om logica aan te brengen in al die nullen en enen. Hiervoor zijn in veel gevallen wiskundige formules nodig om het te kunnen berekenen. Ook kun je met wiskunde de meest efficiënte manier te berekenen om al deze informatie zo snel mogelijk te kunnen versturen. Veel mensen denken bij coderingstheorie aan het beveiligen van informatie maar dat is cryptologie, wat wel een onderdeel is van coderingstheorie.

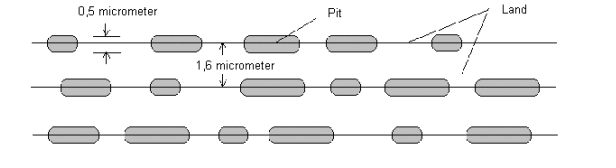
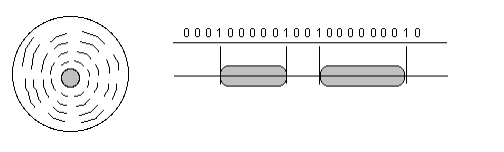
Een praktisch voorbeeld van een coderingstheorie is de ISBN- code op een boek. Deze codes hebben een ingebouwde parity-check die ervoor zorgt dat er geen fouten komen bij het scannen. Bij het voorbeeld van de ISBN-code wordt aan een reeks van 10 cijfers (a1,…,a10) een automatische parity-check toe gevoegd op de volgende manier:

Als je de code 3764322306 op je boek hebt staan wordt er met het versturen de code omgezet in:

Zo wordt dus de bekende ISBN-code verstuurd. Dit wordt dagelijks gedaan. 0 mod 11 houdt in dat als je het eind antwoord deelt door 11 je 0 rest overhoud. In het volgende hoofdstuk gaan we meer soorten codering theorieën bekijken.

# 3 Soorten

## 3.0 Welke soorten coderingstheorieën bestaan er?

Vroeger werd informatie op verschillende manieren verstuurd en opgeslagen. Toen ging dat met pen en papier. Tegenwoordig wordt alles digitaal opgeslagen. Dit niet alleen omdat het een stuk sneller gaat maar ook omdat het veel minder opslagruimte kost. Informatie die jij wilt opslaan wordt omgezet in enen en nullen. Deze nullen en enen kun je vervolgens opslaan op bijvoorbeeld een cd. Als iemand informatie op een cd wil zetten zegt hij of zei vaak dat hij het gaat branden. Het opslaan van informatie op een cd wordt zo genoemd, omdat er tijdens het overzetten van de informatie sporen in de cd worden gemaakt. Sporen zijn dikkere en dunneren gedeeltes in de cd. De nullen en enen op een cd worden ook wel pit en land genoemd.

Niet alles wordt opgeslagen of verstuurd via cd’s. Het grootste deel van de informatie die wordt opgeslagen en verstuurd wordt via internet gedaan. Voor het versturen en opslaan worden ook nullen en enen gebruikt. Om zo’n grote hoeveelheid informatie op te kunnen slaan zijn er in de loop der tijd verschillende coderingstheorieën ontwikkeld waarmee de betrouwbaarheid van communicatiemiddelen wordt gecontroleerd. Signalen worden namelijk verzonden door kanalen die niet ruis vrij zijn. Hierdoor komen er fouten te zitten in de berichtjes die je verstuurd. Daarom worden er controlesymbolen, ook wel redundantie genoemd, toegevoegd aan een signaal. Hierdoor worden fouten eerder ontdekt en verbeterd. Een situatie kun je voorstellen door het gebruik van natuurlijke talen. Hier is ook veel redundantie aanwezig. Namelijk als iemand een schrijffout maakt herken je de fout en kun je hem vaak gemakkelijk corrigeren. Als het je het “dichtstbijzijnde woord” uit het woordenboek. Dan bevat het woordenboek de codewoorden. Afhankelijk van het kanaal bestaan er ‘’codewoorden’’ waarmee de kans op een decodeerfout wordt geminimaliseerd. Om het gehele proces zo goed mogelijk te laten verlopen is er nodig:

1. De kans op een decodeerfout zo klein mogelijk is.
2. De hoeveelheid redundantie (codewoorden) beperkt blijven.
3. Het codeer- en decodeerproces zo eenvoudig mogelijk is.

Om te bereiken dat dit proces zo goed mogelijk verloopt zijn er verschillende mogelijkheden bijvoorbeeld een algebraïsche structuur aan te brengen. De algebraïsche structuur wordt in de praktijk het meest gebruikt. Ook gaan we uitzoeken hoe de herhalingscodering werkt.

## 3.1 Herhalingscode

De meest eenvoudige methode om te coderen is de herhalingscode. De herhalingscode houdt in dat je het bit dat je verstuurd drie keer wordt herhaald. Als je het bit 1 verstuurd zorgt de codeerder dat er 111 vervolgens wordt het bericht pas verstuurd. Als erop tijdens het verzenden iets fout gaat en het bericht niet in originele staat aankomt zal de decodeerder opzoek gaan naar de code die er het meest op lijkt.

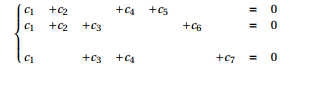
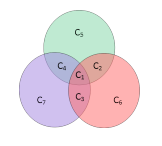
**Voorbeeld**: Stel het woord 0 wordt gecodeerd met de herhalingscode, dan zal het gecodeerde woord gelijk zijn aan 000. Stel het woord 010 wordt ontvangen, dan zal dit verbeterd worden naar 000 en vervolgens gedecodeerd naar 0. Hier is de verbetering goed gegaan, maar stel het woord 011 wordt uit het kanaal ontvangen. Er zijn twee bits veranderd en dit woord wordt verbeterd naar 111, wat na decodering 1 wordt. Bij het laatst ontvangen woord zijn de fouten dus niet goed verbeterd.

## 3.2 Hamming Code

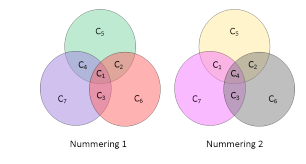
De Hamming methode is een fout-verbeterende methode. De Hamming methode werd vroeger veel gebruikt bij telecommunicatie, maar nu zijn daar efficiëntere coderingstechnieken voor. De Hamming code voegt partybits toe aan het originele woord. De code is een samenstelling van vlakke figuren. De bedoeling is dat de som van elk vlak gelijk is aan nul. In het algemeen geldt hoe meer vlakke figuren hoe ingewikkelder maar ook efficiënter de code wordt. Er bestaan oneindig veel Hamming codes. De simpelste is gelijk aan de herhalingscode en heet de Hamming(3,1) code. 1 staat voor de lengte van het eerste woord en na het coderen is de lengte 3.

In het geval van de Hamming(3,1) code heb je hetzelfde probleem als met de herhalingscode want als er een fout optreedt waarbij c2 en c3 worden veranderd zal de decodeerder c1 ook veranderen in de fout.

Beide cirkels zijn door de ruis veranderd in 1 en daarom wordt c1 ook veranderd in 1. Een ander voorbeeld van de Hamming code is de Hamming(7,4). Met de code kun je woorden van 4 lang versturen en hij codeert ze naar 7. Deze vorm van de Hamming code is al een stuk efficiënter dan de Hamming(3,1) code, want je kunt 4 keer zoveel woorden versturen en hij heeft er minder cirkels voor nodig.

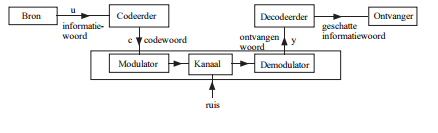
De bits c1 tot en met c4 vormen de woorden uit de bericht lijst en de bits c1 tot en met c7 de woorden uit de codelijst. Dus zijn de bits c5 tot en met c7 de parity bits. De minimale afstand tussen de woorden uit deze code is gelijk aan drie. De som van een cirkel moet gelijk zijn aan 0. 0 houdt in dat er even veel enen als nullen in de som moeten zitten. Een voorbeeld hiervan in het de Hamming(7,4) is als je de binaire code 0101 wilt versturen. Dit betekent dat c1= 0, c2= 1, c3= 0 en c4= 1. Als we nu de som van de cirkels op gaan stellen komt het er zo uit te zien:

Dit kunnen we uitwerken door de formule in te vullen voor cirkel 1 geldt: 0 + 1 + 1 + c5=0. Omdat er even veel enen als nullen in de som zitten zal c5 0 moeten zijn. Dit komt omdat 1 + 1 wordt 0 en 0 +0 wordt 0 nu klopt de som van cirkel 1, want deze moet altijd 0 zijn. Op deze manier kunnen we ook c6 en c7 kunnen bereken. C6 = 0 + 1 + 0, dit betekent dat c6= 1, want dan is de som weer 0 omdat er even veel enen als nullen zijn. Dit geldt ook voor c7, want 0 + 0 + 1 zorgt ook voor een ongelijke hoeveelheid enen. Uit de formule blijkt dat het codewoord dat wordt verstuurd bestaat uit 0101011. De Hamming code kan oneindig groot worden. Dit gebeurt alleen met computers omdat het voor mensen erg moeilijk wordt om het helemaal uit te werken. De Hamming code is wel een erg goede fout detecterende code, want als er door de ruis een 0 veranderd in een 1, of andersom, dan zal de dit ontdekt worden doordat de som niet meer gelijk is aan 0 en daardoor kan het verbeterd worden.

Bij het gebruik van Hamming code moet er wel goed worden gelet op welke code lijst er wordt gebruikt. Door de nummering in de cirkels te veranderen ontstaan er verschillende codes.

In het voorbeeld is te zien dat als je de code 0101 wilt versturen via nummering 1. Dan krijg je als codewoord 0101011. Door het veranderen van de plaatsen van c1 tot en met c4 komt er een ander codewoord dit codewoord wordt dan 0101001. Hieruit blijkt dat het erg belangrijk is dat de ontvanger (decodeerder) weet wat welke codelijst er is toegepast op het verzonden codewoord.

## 3.3 Lineaire Codes



De bron zendt het informatiewoord uit met een lengte genaamd k, vervolgens wordt er door de codeer controlesymbool toegevoegd hierdoor wordt de lengte n. lengte k is oneindig lengte n is ook oneindig. U is het informatiewoord en die verschilt in lengte, daarom zeggen we u= (u1,….,uk). X is het codewoord wat bestaat uit het informatiewoord U en het controlesymbool. X kan ook meerdere tekens bestaan, daarom geldt voor X ook X= (x1,…,xn). Hieruit ontstaat de formule:

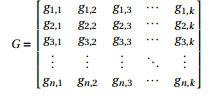


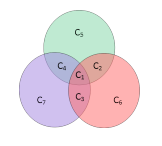
Als er maar een informatiewoord wordt doorgegeven noemen we het een blok code. Als er op extra informatiewoorden bijkomen van eerder tijdstippen noemen we het een convolutiecode. De codeerder werkt dan met geheugen. Lineaire codes hebben belangrijke voordelen boven niet-lineaire:

1. Lineaire codes zijn eenvoudig te beschrijven: het is voldoende een basis van k vectoren te geven;
2. Lineair Coderen komt neer op matrixvermenigvuldiging en dit is technisch goed uitvoerbaar.

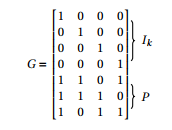
Ook kun je zeggen dat tot op zekere hoogte de lineaire code hetzelfde is als de stelling die C. Shannon bedacht heeft.

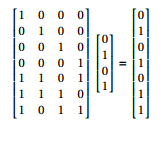
### 3.3.1 Blokcode

De blokcode is een onderdeel van de lineaire codering het is tevens ook de meest gebruikte vorm van lineaire codering het is opgebouwd uit K en N. K staat voor de lengte van het woord en N is de lengte van het codewoord. De grote van de matrix wordt opgebouwd door N x K. K geeft de lengte van het woord verticaal en N geeft het codewoord horizontaal. Het aantal woorden dat je per matrix verstuurt zijn de verticale kolommen elke verticale kolom is een woord. Als er aan de woorden alleen maar parity bits toe worden gevoegd kun je al een stuk meer zeggen over je matrix. Je kunt zeggen dat N-K de hoeveelheid parity bits zijn die zijn toegevoegd aan je matrix.

Als verduidelijking een voorbeeld. We gebruiken weer de Hamming(7,4) code.

Vanuit de Hamming(7,4) code kunnen we een parity submatrix maken. De matrix moet 4 breed zijn, want er worden 4 woorden verzonden en hij moet 7 lang zijn omdat er 3 parity bits per woord worden toegevoegd. Uiteindelijk komt de submatrix er zo uit te zien



De parity submatrix is opgebouwd uit: kolom 1 is c1 = 1. Dan moeten door het oplossen van de som c1 + c2 + c4 +c5 =0 blijken dat c5 = 1, omdat in de andere 4 kolommen de andere woorden worden uitschreven en dan is die c= 1. Door c1 = 1 te nemen is te zien dat c5 t/m 7 ook 1 moeten zijn. Nu is de matrix gemaakt waarmee je je code wilt versturen. De uiteindelijke code is 1000111. Als we nu via de lineaire code het woord 0101 willen versturen moeten we het volgende doen:

De uiteindelijke uitkomst 0101011 is gelijk aan wat we al eerder hebben gezien bij het uitwerken van de Hamming(7,4) code. Het rekenen met de lineaire code gaat een stuk sneller dan met de Hamming code, want je kunt 24 =16 verschillende woorden versturen. De 2 is omdat er 2 mogelijkheden zijn (0 en 1) en 4 omdat je matrix 4 breed is. Het voordeel van de blokcode is dus dat je hem oneindig groot kunt maken. Dit is ook mogelijk bij de Hamming code, maar met de Hamming code kun je minder woorden tegelijk versturen. Ook is het voor de decodeerder duidelijker hoe de code gedecodeerd moet worden. Hierdoor is het een efficiënter systeem. Ook is het de grote van je verstuurde bericht kleiner bij de blokcode dan bij de Hamming code daardoor is het ook nog energie besparend.

# 4 Beveiliging

Al jarenlang stuurt men via deze coderingstheorieën informatie naar elkaar. Soms moet de inhoud van deze berichten geheim blijven. Denk hierbij aan brieven of vertrouwelijke documenten van regering of defensie, maar ook het online bankieren van tegenwoordig of geld opnemen bij een pinautomaat. Door de opkomst van al deze computernetwerken en internet is er veel behoefte aan het beveiligen van deze informatie. Een techniek uit het oude Griekenland is het schrijven van boodschappen op hout, het vervolgens te bedekken met laag was zodat het lijkt alsof het een ongebruikt tablet is.

Vandaag de dag zijn er verschillende mensen die de techniek gebruiken om hun eigendom te beveiligen of geheime informatie te versturen via internet. Door bijvoorbeeld een watermerk aan te brengen kan de auteur zijn unieke stempel drukken op een afbeelding of geluidsbestand. Ook kunnen mensen via afbeeldingen op webpagina's publiekelijk informatie versturen, terwijl alleen ingewijden de informatie kunnen achterhalen.

## 4.1 Wat is cryptografie?

De kunst van het beveiligen van berichten wordt cryptografie genoemd. Het woord stamt uit het Grieks κρυπτει kryptei "verborgen," en γράφω gráfo "schrijven". In de cryptografie bekijkt men de methode waarop personen of apparaten op een veilige manier boodschappen kunnen uitwisselen.

* Vercijferen of versleutelen is het omzetten van een boodschap in een gecodeerde boodschap

Voorbeeld: neem een boodschap met alleen hoofdletters, leestekens en spaties. Neem vervolgens een geheel getal *k*.

|  |
| --- |
| DE GROTE HOND VAN DE BUURMAN IS VORIGE WEEK ZATERDAG BIJ DE HUISARTS GEWEEST, VOOR EEN ONDERZOEK AAN DE OREN. |

Stap 1 – Verwijder alle leestekens en spaties

|  |
| --- |
| DEGROTEHONDVANDEBUURMANISVORIGEWEEKZATERDAGBIJDEHUISARTSGEWEESTVOOREENONDERZOEKAANDEOREN |

Stap 2 – Verdeel de tekst in blokjes van *k* letters. We nemen hier *k =* 4.

|  |
| --- |
| DEGR OTEH ONDV ANDE BUUR MANI SVOR IGEW EEKZ ATER DAGB IJDE HUIS ARTS GEWE ESTV OORE ENON DERZ OEKA ANDE OREN |

Stap 3 – Schrijf de tekst per blokje achterstevoren op.

|  |
| --- |
| RGED HETO VDNO EDNA RUUB INAM ROVS WEGI ZKEE RETA BGAD EDJI SIUH STRA EWEG VTSE EROO NONE ZRED AKEO EDNA NERO |

Stap 4 – Voeg alle blokjes weer bij elkaar tot één geheel

|  |
| --- |
| RGEDHETOVDNOEDNARUUBINAMROVSWEGIZKEERETABGADEDJISIUHSTRAEWEGVTSEEROONONEZREDAKEOEDNANERO |

Stap 5 – Verdeel de tekst in blokjes van *l* letters. Hier is *l* = 5

|  |
| --- |
| RGEDH ETOVD NOEDN ARUUB INAMR OVSWE GIZKE ERETA BGADE DJISI UHSTR AEWEG VTSEE ROONO NEZRE DAKEO EDNAN ERO |

Stap 6 – vul het laatste blokje aan met willekeurige letters tot het lengte *l* bereikt

|  |
| --- |
| RGEDH ETOVD NOEDN ARUUB INAMR OVSWE GIZKE ERETA BGADE DJISI UHSTR AEWEG VTSEE ROONO NEZRE DAKEO EDNAN EROVN |

De waarde *k* = 1 heeft geen effect op het bericht en heeft dus geen zin om te gebruiken. De grootste waarde die je kunt nemen is de lengte van het bericht. Als het bericht *n* tekens bevat is de waarde van *k* = 1,2,…,*n*. De *k*-ruimte is dus de verzameling {1,2,…,*n*}.

##### Principe van Kerckhoff

Probeer nu maar eens uit te zoeken wat het eindproduct betekent zonder naar de stappen te kijken. Echter, de methode uit het voorbeeld is een erg slecht cryptosysteem. Als de waarde *k* onbekend is, kun je hem makkelijk ontcijferen door maar wat in te vullen, als je weet welke methode gebruikt wordt tenminste. Daarom wordt in de cryptografie het principe van Kerckhoff gehanteerd. Dit houdt in dat als er een vercijferde boodschap wordt gestuurd, ervan uit mag worden gegaan dat een mogelijke afluisteraar weet welk cryptosysteem er gebruikt wordt.

Aanleiding tot het principe van Kerckhoff:

* Een methode is meestal ingebouwd in een apparaat en is dus vrij moeilijk vervangbaar;
* Vaak wordt gebruik gemaakt van software in webbrowsers en email-programma’s. Dit kan natuurlijk niet geheim blijven;
* Bij ontwikkelaars van de methode is deze bekend. Toch wil je dat zij je boodschap niet kunnen lezen;
* Zo hoef je zo weinig mogelijk informatie geheim te houden en dat is makkelijker dan veel informatie geheimhouden;

### Caesarcryptografie

Een van de oudste cryptosystemen is het Caesarcryptosysteem. Dit werd gebruikt door Julius Caesar om geheime boodschappen over te brengen zonder dat de koerier en eventuele tussenpersonen te weten kunnen komen wat het betekent. Dit systeem is gebaseerd op het verschuiven van letters in het alfabet. Bij dit schuifsysteem kiest men dus weer een geheel getal *k* en schuift iedere letter *k* plaatsen op. Ik dit voorbeeld is een alfabet van 26 letters gekozen. Dit is niet noodzakelijk. Er kan worden afgesproken om leestekens en spaties toe te voegen, hierdoor zijn er meer tekens en is het lastiger om de code te ontcijferen. Helaas hadden spionnen vaak al snel door hoe deze boodschappen werden versleuteld en konden zij het zelf terugdraaien naar het oorspronkelijke bericht.

### Affiene cryptografie

Men moet vaak berekeningen uitvoeren bij het vercijferen van een boodschap. De letters worden dan eerst omgezet in rangnummers. Met deze rangnummers kan dan worden gerekend. Bij dit voorbeeld krijgt A rangnummer 0, B rangnummer 1… Dit maakt Z rangnummer 25.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| K | L | M | N | O | P | Q | R | S | T |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 20 |
| U | V | W | X | Y | Z |  |  |  |  |
| 21 | 22 | 23 | 24 | 25 | 26 |  |  |  |  |

Het probleem dat optreedt is dat je al gauw getallen krijgt die niet meer in het gebied [0…25] zitten. Om dit op te lossen, schrijft men het alfabet oneindig vaak achter elkaar.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| … | W | X | Y | Z | A | B | C | D | … | X | Y | Z | A | B | C | … | Y | Z | A | B | C | … |
| … | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | … | 23 | 24 | 25 | 26 | 27 | 28 | … | 50 | 51 | 52 | 53 | 54 | … |

Dit kan worden aangegeven met een cirkel, net als een klok. Het aantal keer dat er om de cirkel wordt heen gegaan. Het reststukje is de plek van uitkomst op de cirkel. Rest wordt dus berekent door aantal letters in het alfabet. Als voorbeeld, een klok gaat na twaalf uur weer op nul. Als er honderd uur wordt geteld, dan moet er honderd door twaalf worden gedeeld. 100:12=8 rest 4. En zo ook met het alfabet hierboven. Neem bijvoorbeeld het getal 85. 85:26=3 rest 7.

een functie waarmee we een rangnummer kunnen achterhalen is de encryptiefunctie, meestal aangegeven met de letter E. Een voorbeeld is de encryptiefunctie. Voor deze functie geldt  en . Zo worden de letters berekend.

### Monoalfabetische substitutie

bij monoalfabetische substitutie wordt het alfabet helemaal door elkaar gehusseld.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| *E(x)* | b | m | y | k | z | a | s | c | r | l | n | i | e | o | d | p | w | j | q | u | h | x | t | v | g | f |

De methode om een sleutel te vinden door alle sleutels uit te proberen heet “Exhaustive key search”. Doordat bepaalde letters vaker voorkomen dan andere is dit systeem makkelijk om te ontcijferen door middel van een letterfrequentietabel.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E 19.06% | N 9.41% | T 6.74% | A 6.72% | R 6.45% | I 6.44% | D 5.91% |
| O 5.87% | S 4.00% | L 3.94% | G 3.14% | V 2.90% | M 2.41% | H 2.32% |
| K 2.28% | U 1.93% | B 1.80% | C 1.60% | P 1.59% | W 1.57% | J 1.49% |
| Z 1.18% | F 0.74% | Y 0.29% | Q 0.11% | X 0.11% |  |  |

Als de letter E wordt verandert in de letter K dan is ongeveer 19% de letter K. Er hoeft dus alleen maar geteld worden hoe vaak er een letter in een stuk tekst voorkomt.

### Vigenère

Het Vigenère-cryptosysteem werd in 1553 bedacht door Giovanni Batista Belaso. Maarhij stond bekend als Blaise de Vigenère. Hij bedacht om een sleutelwoord steeds achter elkaar te herhalen onder een stuk tekst. Het rangnummer is de letter van een woord van de zin plus een letter van het sleutelwoord. Hier een voorbeeld met als sleutelwoord wiskunde.

deze tekst vercijferen we met vigenere

wisk undew iskundewisk un dew iskundew

zmro nrnwp ewbwvmjazwx qr pip daqyahva

om een zin te maken of te ontcijferen wordt er een vigenère-tabel gebruikt.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| a | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| b | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a |
| c | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b |
| d | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c |
| e | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d |
| f | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e |
| g | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f |
| h | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g |
| i | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h |
| j | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i |
| k | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j |
| l | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k |
| m | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l |
| n | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m |
| o | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n |
| p | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| q | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p |
| r | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q |
| s | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r |
| t | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s |
| u | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t |
| v | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u |
| w | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v |
| x | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w |
| y | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x |
| z | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |

## 4.2 Wat heeft cryptologie met coderingstheorie te maken?

Als je tegen mensen zegt coderingstheorie dan denken ze vaak aan het beveiligen van bestanden zodat niemand ze kan lezen. Dit is niet zo het beveiligen van bestanden is cryptologie. Cryptologie is het versleutelen van data. Met de juiste codering theorieën. Kun je je data zo versturen dat niemand het kan onderscheppen al dan niet lezen. Voor terroristen is dit ideaal, want zo kunnen zij alles bespreken voor ze iets uitvoeren. De regering wil daarom de sleepwet invoeren. Deze wet houdt in dat de AIVD in eerste instantie kan zien met wie je contact hebt. Als ze daartussen iets verdachts vinden kunnen ze verder gaan en daadwerkelijk kijken waar je het over hebt gehad. Veel mensen willen dit niet, omdat zij bang zijn voor hun privacy. Maar de regering wil het alleen maar invoeren om criminaliteit te kunnen bestrijden en niet om te zien wat mensen allemaal doen in hun vrije tijd. Door het versturen van versleutelde bestanden is het heel belangrijk dat er niks fout gaat in het coderen en versturen. Het is voor de ontvanger al erg moeilijk om het aangekomen bestand te ontcijferen. Als hierbij ook nog fouten zijn opgetreden in het versturen wordt het praktisch onmogelijk. Daarom worden er voor het versturen van versleutelde bestanden de moeilijkste soort codering theorieën toegepast. Dit wordt gedaan zodat alleen ‘’super’’ computers ze kunnen decoderen en ontcijferen. Hier wordt erg veel gebruik van gemaakt door overheden die niet willen dat er hackers uit andere landen mee kunnen kijken. Ook bij het versturen van een code kunt je cryptologie toepassen voor een extra beveiliging.

Als je de volgende code wilt versturen dan kun je dit ook beveiligen.

|  |
| --- |
| 010101101011 |

Hak de code in stukjes:

|  |
| --- |
| 010 101 101 011 |

Vervolgens draai je de blokjes om

|  |
| --- |
| 010 101 101 110 |

Na het aan elkaar plakken deel je ze weer op in blokjes

|  |
| --- |
| 0101 0110 1110 |

Draai ook deze blokjes weer om

|  |
| --- |
| 1010 0110 0111 |

Plak de code weer aan elkaar

|  |
| --- |
| 1010110011 |

Nu heb je ook je verstuurde code gecodeerd. Als de ontvanger hier niets aan doet zal er bij het decoderen van een hoop fout gaan. Daarom moet je je decodeerder afstellen op de techniek waarmee de code versleuteld is. Als dit op een juiste manier gebeurd en op veel grotere schaal dan het in hierboven uitgevoerde voorbeeld, is het een hele effectieve manier van het beveiligen van je code. Het is namelijk heel erg moeilijk om een logische volgorde te vinden in alle nullen en enen. Toch wordt deze manier van versleutelen niet vaak gebruikt om dat het veel foutgevoeliger is dan bij normaal versleutelde zinnen die gecodeerd worden.

# 5 Ethische Paragraaf

## Casus

In 2012 is er een terroristische aanval geweest in Parijs door IS. Heel de wereld stond hier ondersteboven van en alle landen hebben sindsdien betere beveiliging geregeld om hun land beter te beschermen. De regering denkt er nu dus aan om privéberichten, op sociale media platformen, van verdachte personen af te tappen, om de veiligheid van burgers te verbeteren, maar er worden dan ook berichten afgetapt van onschuldige burgers. Daarom zijn er veel protesten geweest om deze wet tegen te gaan. Veel burgers stellen hun privacy heel hoog, en zouden dit niet willen verliezen. Daartegenover zijn er ook nog andere burgers die het wel met de regering eens zijn, zodat ze veiliger zijn en zich ook veiliger voelen. Nu zit de tweede kamer met een tweestrijd tussen burgers en de regering. De wet die zal worden ingevoerd heet de sleepwet.

## Achtergrondinformatie

De wet die de regering wil invoeren heet de sleepwet. Hierbij worden berichten die door middel van ruis worden verstuurd, opgevangen door de regering om te bekijken of er iets verdachts aan zit, ook mogen telefonische gesprekken worden afgeluisterd, dit wordt een 'sleepnet' genoemd. Alle geautomatiseerde apparaten mogen gehackt worden zoals; een mobiele telefoon, computer of tv. Er mag een geheime map met gegevens van één of meerdere personen worden gemaakt waar iedereen in kan komen. Alle opgeslagen data mag met de buitenlandse overheid gedeeld worden. De opgeslagen data zal 3 jaar lang bewaard worden.

Tegenstanders van deze wet zijn SP, GroenLinks, D66, Partij voor de Dieren, Groep Kuzu/Öztürk en Klein, Amnesty International, Milieudefensie, Privacy Barometer, Privacy First, Radar, Bits of Freedom. En de burgers die tegen de wet zijn. Voorbeelden van argumenten van deze groepen zijn:

- Als dit beleid niet in het buitenland wordt gevoerd, waarom dan wel in Nederland;

- er is te weinig budget om dit beleid te voeren, want de mensen die alle data moet bekijken kosten heel veel geld en dit gaat ten koste van agenten die op patrouille gaan, vanwege er geen geld meer is om die agenten te betalen;

- het is schending van privacy

- de wet is te vaag en het plan van aanpak is nog te onduidelijk;

- het kan dat de opgeslagen data wordt gehackt en in de verkeerde handen valt;

- er moet een referendum komen om te zien wat de maatschappij wilt.

Voorstanders van de wet zijn ChristenUnie, 50 plus, CDA, VVD, Forum voor Democratie, SGP, PVV en burgers voor de wet. Voorbeelden van argumenten van deze meerderheid zijn:

- Nederland wordt een stuk veiliger;

- Er kunnen extra overtreders van de wet mee worden genomen bij invoering van de wet.

- vernieuwing van de wet is nodig, want de oude wet is al 50 jaar oud;

- Een referendum zal geen verschil maken, want de wet komt er sowieso

## Belanghebbende

|  |  |  |
| --- | --- | --- |
| Belanghebbenden | Belang in de situatie | Waarden |
| Burgers voor de wet | Burgers die voor de wet zijn willen dat hun veiligheid wordt verbeterd, zij hebben dus belang aan veiligheid. | * Veiligheid (sociaal) |
| Burgers tegen de wet | Burgers die tegen de wet zijn die vinden hun privacy belangrijker dan hun veiligheid, zij hebben belangen aan privacy. | * Privacy (sociaal) |
| De regering | De regering wil de wet invoeren, omdat de burgers dan beter worden beschermd, zij hebben belangen aan bescherming. | * Veiligheid (sociaal) |
| Maatschappij | De maatschappij is in de context opgesplitst in twee groepen. Een groep vindt dat ze hun privacy verliezen, wanneer de wet wordt ingevoerd. De andere groep wil juist dat de wet wordt ingevoerd, zodat er eerder misdadigers kunnen worden gepakt en ze zich veel veiliger zullen voelen | * Privacy (sociaal) * Veiligheid (sociaal) |
| Sociale media | De sociale media krijgt ontzettend veel aandacht op het moment, want zij spelen een erg grote rol in de nieuwe wet. Zij zouden alle gegevens van alle gebruikers die de applicatie gebruiken moeten vrijstellen aan de overheid wanneer de wet wordt ingevoerd, ook al willen ze dit niet. Het wordt er eigenlijk tot toe gedwongen om data bloot te stellen. | * Aandacht (sociaal) * Dwang (sociaal) |

## Ethische vraag

Moet de regering social-media platformen verplicht stellen om gegevens van privéberichten vrij te geven aan de overheid om criminaliteit te voorkomen?

## Dilemma

Mensen willen graag dat privéberichten voor hunzelf te zien zijn, want mensen sturen soms kwetsbare berichten naar elkaar, maar die berichten worden ook ontvangen bij de overheid, als de regering een wet invoert om berichten te onderscheppen. De waarden veiligheid en vrijheid botsen hier tegen elkaar. Mensen weten niet waar hun berichten naartoe gaan, maar ze zijn wel een stuk veiliger.

## Afweging en Oordeel

### Afweging

|  |  |  |  |
| --- | --- | --- | --- |
| Waarde/keuze | Belanghebbenden | Gevolg +/- | Waarde(n) |
| Veiligheid | Burgers voor de wet | + burgers voor de wet zullen blij zijn als de wet wordt ingevoerd en voelen ze zich veiliger. | * Vreugde * Veiligheid * Vertrouwen |
|  |  | * Door invoering van de wet zou het kunnen dat het fout gaat in het systeem en zo, wordt vertrouwen verloren. |  |
|  | Burgers tegen de wet | + de burgers worden veiliger wanneer de wet wordt ingevoerd, ook al verliezen ze privacy. | * Vrijheid * Veiligheid * Vertrouwen |
|  |  | * De burgers zijn bang dat het systeem zal worden gehackt en dat er dan data in de verkeerde handen valt. Ook verliezen de burgers vertrouwen wanneer de wet wordt ingevoerd |  |
|  | De regering | + de regering wil dat de wet wordt ingevoerd, vanwege veiligheid voor de burger en ook voor vernieuwing, want de wet die hedendaags geldt is al 50 jaar oud. | * Veiligheid * Veiligheid |
|  |  | * De regering moet wel optimale beveiliging hebben voor de opgeslagen data, want er mag absoluut niks fout gaan in het systeem, anders is er geen vertrouwen meer en is heel Nederland boos. |  |
|  | Sociale media | + sociale media zorgt voor de extra veiligheid van de burgers. | * Veiligheid * Vrijheid |
|  |  | * Sociale media moet wel tegen hun zin gegevens vrijgeven die privé zijn. |  |
|  | De maatschappij | + de maatschappij heeft door deze wet wel meer veiligheid maar minder privacy. | * Veiligheid * Vrijheid |
|  |  | * De maatschappij voelt zich ook weer minder veilig, want mensen weten niet wat er met hun gegevens gedaan wordt. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Waarde/keuze | Belanghebbenden | Gevolg +/- | Waarde(n) |
| Vrijheid | Burgers voor de wet | + de vrijheid voor deze groep wordt een stuk beperkt, maar ze hebben dat ervoor over als ze meer vrijheid krijgen. | * Vrijheid * Vertrouwen |
|  |  | * Het zou kunnen dat er dingen worde gedaan met de gegevens van een persoon waar ze niks van weten. Vertrouwen kan hierdoor verloren worden. |  |
|  | Burgers tegen de wet | * Burgers tegen de wet willen hun vrijheid absoluut niet verliezen en zullen vertrouwen verliezen in de regering. | * Vrijheid * Vertrouwen |
|  | De regering | + de regering moet de vrijheid beperken om meer veiligheid voor de burger te verzorgen. | * Veiligheid * Vrijheid |
|  |  | * Ook al wil de regering dat de burger zo veilig mogelijk is moeten de burgers een stuk van hun vrijheid inleveren, ook al wil de regering dit niet voor de burger. |  |
|  | Sociale media | * Sociale media moet tegen hun zin vrijheid van andere opgeven. Dit gaat tegen hun principes in. | * Vrijheid |
|  | De maatschappij | + een groep wil dat hun vrijheid wordt beperkt, zodat ze meer veiligheid hebben. | * Vrijheid * Veiligheid * Vertrouwen |
|  |  | * Een groep wil dat hun vrijheid niet wordt beperkt en dat ze gewoon vertrouwen kunnen hebben in de regering. |  |

### Oordeel

Wij vinden dat de waarde veiligheid het beste is in deze situatie. Vrijheid is natuurlijk wel belangrijk en als je vertrouwen kan verliezen in de regering, wanneer de wet wordt ingevoerd. Maar al vele jaren is iedere Nederlander er mee eens dat er een democratie in Nederland wordt gevoerd. In een democratie is het, dat de burger een stuk van zijn vrijheid moet inleveren, zodat er een bepaalde groep mensen het volk als beloning bescherming geeft. Dus veiligheid heeft duidelijkere standpunten dan vrijheid.

## Norm

Een wet om berichten op te vangen, zodat criminaliteit voorkomen kan worden mag worden uitgevoerd.

## Eigen mening

### Noud

Ik vind dat de regering recht heeft op het aftappen van deze informatie. Ik mijn opzicht is veiligheid belangrijker dan privacy. Bovendien vind ik dat privacy of vrijheid niet eens van toepassing is, de informatie die wordt verschaft wordt namelijk niet openbaar gemaakt, maar blijft in handen van de overheid. Daardoor zal er geen privacy worden geschonden. Dus waarom ertegen zijn als je niets te verbergen hebt? Volgens mij zit daar het gevoelige punt bij tegenstanders. Mensen die misschien qua gedrag op internet niets met terreur en dergelijke te maken hebben maar wel met andere illegale zaken kunnen op deze manier ook worden ontmaskerd. Aangezien ik daar zelf geen voorbeeld van ben, blijft mijn mening dat ik er niet op tegen ben.

### Alex

Ik vind dat de regering het recht moet hebben om mee te kunnen kijken bij de berichten van mensen die hun verdacht vinden. Veiligheid is in mijn optiek vele male belangrijker dan privacy. Als het vrijgeven van berichten kan helpen aan het bevorderen van de veiligheid vind ik dit een goede zaak. Ook vraag ik me af bij de mensen die erop tegen zijn wat hun er erg aan vinden. Als je niks te verbergen hebt maakt het toch niks uit. Daarentegen vind ik wel dat er alleen maar gekeken moet worden bij mensen die ook verdacht zijn en niet willekeurig. Ik vind dat er voor alle criminele zaken de politie de macht moet hebben om mee te kunnen kijken. Dus mijn mening blijft dat ik er niet op tegen ben.

### Luca

Mijn mening is; De regering mag berichten opvangen van onschuldige burgers en verdachte burgers, want ik voel me daardoor veiliger, omdat ik weet dat er een grotere kans is om criminelen te vangen en ik vertrouw de regering er tot toe, dat ze mijn gegevens veilig zullen bewaren en er niks mee zullen doen. Ik heb zelf niks te verbergen dus ik heb niks te vrezen.

# 6 Onderzoek

##### Inleiding

Het is vrij lastig deze theorieën praktisch te visualiseren. Hiervoor zijn servers en ingewikkelde software nodig die niet tot onze beschikking zijn. Daarom is er gekozen een proef op te stellen op basis van de cryptografie. Dit staat tevens in verband met codering en komt aan de orde in dit werkstuk.

Maar hoe betrouwbaar is het beveiligen van deze bestanden? Kunnen beveiligingstechnieken zoals in het hoofdstuk Beveiliging makkelijk worden ontcijferd? Als deze relatief makkelijke methoden door mensen snel worden gekraakt, kunnen computers dit nog veel sneller en effectiever.

## Onderzoeksvraag: Zijn crypto-techtnieken makkelijk te ontcijferen?

##### Hypothese

De resultaten zullen verschillen per groep. De deelnemers geven aan in welke leeftijdscategorie en maatschappelijke positie zich bevinden wat gevolgen zal hebben in de resultaten. Ook de gewijde tijd zal van invloed zijn.

• De leeftijdscategorie bepaalt de ervaring met eventuele ontcijfering en of onderzoeken van bepaalde teksten/codes;

• De maatschappelijke positie van iemand kan van belang zijn omdat mensen die tijdens hun werk met zulke codes te maken hebben sneller in staat zullen zijn deze te ontcijferen dan mensen die bijvoorbeeld in een andere sector werken;

• Tenslotte is de tijd bepalend omdat een persoon die langer aan het proberen is meer kans heeft op het ontdekken dan iemand die na korte tijd opgeeft.

Uiteindelijk zal een deel de zinnen kunnen ontcijferen en de resultaten worden 50,0% juist en 50,0% onjuist.

##### Materialen en Methoden

Voor het uitvoeren van de proef is gekozen voor het maken van een enquête met behulp van Google Forms.

De deelnemer kiest eerst geslacht, leeftijd en maatschappelijke positie. Vervolgens wordt gevraagd een zin te ontcijferen.

|  |
| --- |
| DE RECHTER HEEFT BESLOTEN DAT DE MAN EEN TAAKSTRAF MOET UITVOEREN |
| * D RCHTR HFT BSLTN DT D MN N TKSTRF MT TVRN. |

Hier is gekozen voor een van de meest eenvoudige manieren van omschrijven, namelijk door alle klinkers weg te halen. Vervolgens wordt het gecompliceerd door een zin in blokjes van *l* = 4 te verdelen en de letters om te draaien.

|  |
| --- |
| DE TAFEL KOMT DIRECT UIT DE WOONWINKEL |
| * ATED KLEF DTMO CERI TIUT OWED IWNO LEKN |

De derde zin bestaat uit blokjes van *l* = 5. De volgorde van de lengte binnen de blokjes in onaangepast hoewel de volgorde van de blokjes zelf wel is aangepast.

|  |
| --- |
| DE HOND VAN DE BUREN HEEFT EEN GEBROKEN POOT |
| * DEHON TEENG DVAND EBURE ENPOO NHEEF EBROK T |

Bij de laatste zin zijn de stappen uit het voorbeeld van hoofdstuk Beveiligen genomen. Hierbij is de zin opgedeeld in blokjes van *l* = 4. Vervolgens zijn de letters omgedraaid en de volledige zin weer bij elkaar gebracht. Ten slotte is de tekst weer in blokjes van *l* = 5 verdeeld en het laatste blokje met willekeurige letters tot lengte 5 aangevuld.

|  |
| --- |
| DE GROTE HOND VAN DE BUURMAN IS VORIGE WEEK ZATERDAG BIJ DE HUISARTS GEWEEST VOOR EEN ONDERZOEK AAN DE OREN |
| * RGEDH ETOVD NOEDN ARUUB INAMR OVSWE GIZKE ERETA BGADE DJISI UHSTR AEWEG VTSEE ROONO NEZRE DAKEO EDNAN EROVN |

##### Resultaten

Hieronder staat de verdeling van de deelnemers qua leeftijd, geslacht en maatschappelijke positie weergegeven in een diagram.

De resultaten van de ontcijfering worden binominaal weergegeven. Dit wil zeggen goed of fout. Foute antwoorden worden namelijk verschillend ingegeven, zowel niets als onzin. Bovendien heeft men exact het goede antwoord nodig om de beveiliging te omzeilen.

|  |  |
| --- | --- |
| Resultaten eerste zin |  |
| Juist | 57,1% |
| Onjuist | 42,9% |

|  |  |
| --- | --- |
| Resultaten tweede zin |  |
| Juist | 21,4% |
| Onjuist | 78,6% |

|  |  |
| --- | --- |
| Resultaten derde zin |  |
| Juist | 50,0% |
| Onjuist | 50,0% |

|  |  |
| --- | --- |
| Resultaten vierde zin |  |
| Juist | 0,0% |
| Onjuist | 100,0% |

##### Verwerking van de resultaten

Wat opvalt is de spreiding van de resultaten. Deze zijn hieronder weergegeven in een spreidings-diagram

Er wordt verwacht dat de grafiek een rechte dalende lijn zal zijn naarmate de zinnen moeilijker worden te ontcijferen, maar in dit geval vindt er een stijging plaats van de vragen twee naar drie. Dit betekent dat vraag twee makkelijker wordt ervaren als vraag drie.

Wat ook opvalt is het resultaat van de laatste zin. Niemand wist deze zin te ontcijferen. Dit is vreemd omdat deze zin beveiligd is op een basisniveau.

##### Conclusie

Crypto-technieken zijn erg betrouwbaar. Vooral gezien het feit dat de laatste zin uit de proef door niemand kon worden ontcijferd. Hierbij kan de hypothese dus ook worden verworpen. Omdat deze eenvoudig beveiligde zinnen niet kunnen worden ontcijferd, stellen we crypto-technieken als betrouwbaar.

# Lob paragraaf

Voor de Lob paragraaf hebben we uitgezocht in welke studie het vak coderingstheorie voorkomt. We zijn erachter gekomen dat het voorkomt je tijdens de opleiding informatica het vak coderingstheorie voor komt. Wij hebben een aantal bronnen gebruikt van de Technische universiteit Eindhoven. Coderen is bij informatica een basisvak. Hier leren studenten het binair coderen en gaan verder in op talstelsels. Ook komt het vak coderingstechniek voor in soortgelijke studies zoals technische informatica. Het probleem is dat coderen alleen door computers wordt gedaan. Het heeft voor mensen geen zin meer om het zelf te doen. Er zijn steeds meer middelbare scholen die ook coderen en programmeren als keuzevak beschikbaar stellen. Dit wordt ook op scholen gegeven als een soort masterclasses wat we op het TBL kennen. Coderen gaat voor ons nog veel betekenen daarom is het goed dat er al eerder begonnen wordt met het geven van lessen over dit onderwerp.

## Conclusie

Coderingstheorie is de manier waarop informatie verstuurd wordt via computers. Het wordt dagelijks heel veel gebruikt. Het is tegelijkertijd ontstaan met het ontstaan van de computer. Er zijn veel verschillende soorten van coderen die kunnen worden toegepast. Helaas zijn ze niet allemaal even betrouwbaar en belangrijk. Samen met cryptologie is coderingstheorie een manier om informatie te kunnen versturen zodat niemand ze ooit kan ontcijferen.

## Reflectie

### Noud

Dit project was erg uitdagend, vooral omdat we een onderwerp kiezen waar je elke dag mee te maken hebt hoewel je je er niet bewust van bent. Wel heb ik de tijd erg onderschat. Vooral het doen van bronnenonderzoek. Voorheen dacht ik nooit veel na over de betrouwbaarheid van een website of tijdschrift. Tijdens het profielwerkstuk heb ik geleerd dat het juist heel belangrijk is of de auteur deskundige is en om te bepalen of de informatie uit deskundige bronnen wordt opgedaan. Over het algemeen is alles goed verlopen. De taakverdeling ging vlekkeloos en ieder nam de verantwoordelijkheid en zorgde dat alles op tijd in orde was.

### Luca

Ik vond dit project erg interessant om te doen, ik heb tijdens het maken van dit project geleerd om creatief te denken en natuurlijk over de coderingstheorie. De samenwerking ging heel goed en ik werd ook geholpen door Noud en Alex, wanneer ik het nodig het en dat was erg fijn. Ik heb dit project wel onderschat in de tijd, want er zit erg veel tijd en inzet in. Ook heb ik geleerd om verder te zoeken naar bronnen dan alleen de populaire websites, want die zijn niet allemaal even betrouwbaar. Ik ben blij dat ik dit project met mijn vrienden heb mogen doen en dat we dit onderwerp mochten doen.

### Alex

Ik vond de samenwerking erg goed gaan. Als er iets gedaan moest worden deed niemand er moeilijk over om het te doen. Dit was erg fijn, want dat betekent dat iedereen het leuk vond om te doen. Ook heb ik veel geleerd van het opzoeken van informatie. Ik ben erachter gekomen dat veel informatie die op internet staat niet altijd bruikbaar is en je niet zomaar mag overnemen. Ik had wel eerder moeten beginnen en de tijd moeten spreiden. In het begin had ik niet echt een idee hoe het profielwerkstuk er nou uit moest komen te zien daarom vond ik het erg moeilijk om een begin te maken. Maar uiteindelijk vind ik het een mooi project geworden waar ik persoonlijk erg veel van geleerd heb. Niet alleen de nieuwe kennis over coderingstheorie maar ook hoe je een project moet maken.

# 7 Bijlagen

## 7.1 Hoofdstuk 1

[http://pub.math.leidenuniv.nl/~jongrsde/olderteaching/ct2](http://pub.math.leidenuniv.nl/~jongrsde/olderteaching/ct2     010/Acodinput.pdf)

[010/Acodinput.pdf](http://pub.math.leidenuniv.nl/~jongrsde/olderteaching/ct2     010/Acodinput.pdf)

<https://plus.maths.org/content/os/issue3/codes/index>

<https://www.webcitation.org/5s0Ome6c1?url=http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>

<http://lietraloe.com/article/richard-hamming>

<http://www.stg-ece.nl/informatica/moduled/lesd4_foutdetectie_en_correctie.pdf>

file://hhh-w-smb003.carmelad.nl/home/HHH19688/Downloads/19557A.pdf

http://www.fi.uu.nl/wiskrant/artikelen/252/252december\_geer.pdf

## 7.2 Hoofdstuk 2

<http://pub.math.leidenuniv.nl/~jongrsde/olderteaching/ct2010/Acodinput.pdf>

<http://www.win.tue.nl/wskpr/lessen/codering/Hamming-Text-v1.1.pdf>

<https://repository.tudelft.nl/islandora/object/uuid:2adb6ec5-e168-41f2-bd49-a20c9c08c125/datastream/OBJ/download>

<http://www.win.tue.nl/~jessers/aansluiting/foutje.pdf>

## 7.3 Ethische Paragraaf

<https://sleepwet.nl/>

<https://geensleep.net/>

<https://www.bof.nl/2017/02/14/tweede-kamer-kiest-helaas-voor-sleepnet-wie-stemde-wat/>

<https://matroosoft.tweakblogs.net/blog/15359/10-redenen-om-tegen-de-sleepwet-te-stemmen/>

<https://big.nl/de-nieuwe-sleepwet-helemaal-niet-om-de-privacy-van-burgers-maar-om-terrorismebestrijding/>

<https://big.nl/deze-studenten-zijn-op-weg-naar-een-referendum-over-jouw-privacy/>

<https://joop.bnnvara.nl/opinies/red-de-democratie-stop-de-sleepwet/>

<https://www.privacybarometer.nl/nieuws/3914/De_vijf_grootste_misverstanden_over_het_sleepwet-referendum>

<http://www.fi.uu.nl/ctwo/WiskundeD/MateriaalDomeinenWiskundeD/CryptografieVwo/docs/Cryptografie.doc>

# 8 Bibliografie

(Claude), C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 55.

Bakker, m. S. (2017). *Cryptografie.* Retrieved from Cryptografie: http://www.fi.uu.nl/ctwo/WiskundeD/MateriaalDomeinenWiskundeD/CryptografieVwo/docs/Cryptografie.doc

Bits Of Freedom. (2017). *Geen sleepwet.* Retrieved from Geensleep.net: https://geensleep.net/

Collins, G. P. (2002, Oktober 14). *Claude E. Shannon: Founder of Information Theory*. Retrieved from scientificamerican.com: https://www.scientificamerican.com/article/claude-e-shannon-founder/

Essers, J. (n.d.). *Foutje? Dat verbeteren we toch*. Retrieved from Ethan Frome: http://www.win.tue.nl/~jessers/aansluiting/foutje.pdf

G van der Geer, M. v. (2017, 11 23). *pub math leiden univ*. Retrieved from leidenunv.nl: http://pub.math.leidenuniv.nl/~jongrsde/olderteaching/ct2010/Acodinput.pdf

G. van der Geer, M. v. (n.d.). *Syllabus Coderingstheorie.* Retrieved from Leiden University.

Gijsberts, A. (2017, Oktober 2). *Studenten op weg naar referendum*. Retrieved from Big.nl: https://big.nl/deze-studenten-zijn-op-weg-naar-een-referendum-over-jouw-privacy/

Graaf, J. d. (2017, Oktober 7). *Red de sleepwet stop de Democratie*. Retrieved from Bnnvara.nl: https://joop.bnnvara.nl/opinies/red-de-democratie-stop-de-sleepwet

Korteweg, D. (2017, Februari 14). *Tweede Kamer kiest helaas voor sleepwet.* Retrieved from Bof.nl: https://www.bof.nl/2017/02/14/tweede-kamer-kiest-helaas-voor-sleepnet-wie-stemde-wat/

Lans, D. (2011). *Foutdetectie en Correctie.* Retrieved from Stichting Emmauscollege Rotterdam: http://www.stg-ece.nl/informatica/moduled/lesd4\_foutdetectie\_en\_correctie.pdf

Lint, J. v. (1980). *Inleiding in de Coderingstheorie.* Amsterdam: MC Syllabus 31.

Pennekamp, J. (2017, oktober 2). *De nieuwe sleepwet tegen terrorisme* . Retrieved from Big.nl: https://big.nl/de-nieuwe-sleepwet-helemaal-niet-om-de-privacy-van-burgers-maar-om-terrorismebestrijding/

Plus. (1997). Coding theory: the first 50 years. *Plus*, 13.

Puijnenbroek, J. v. (2016, Juni 28). *Richard Hamming*. Retrieved from LietraLoe: http://lietraloe.com/article/richard-hamming

Redactie Privacy Barometer. (2017, september 27). *De vijf grootste misverstanden over de sleepwet*. Retrieved from PrivacyBarometer.nl: https://www.privacybarometer.nl/nieuws/3914/De\_vijf\_grootste\_misverstanden\_over\_het\_sleepwet-referendum

TIjn, L. N. (2017). *De sleepwet*. Retrieved from Sleepwet.nl: https://sleepwet.nl/

Veer, G. v. (2005). Codes. *Wiskrant*, 30.

Visser, Y. (2016, April 16). *Historiek.net*. Retrieved from Claude Shannon (1916-2001) - Amerikaanse Wiskundige: https://historiek.net/claude-shannon-google-doodle/58748/

# 9 Logboek

|  |  |  |  |
| --- | --- | --- | --- |
| Wie | Wanneer  Datum | Hoelang min | Wat |
| Noud, Luca en Alex | 07-07-2017 | 180 | Begonnen start opdracht |
| Noud, Luca en Alex | 10-07-2017 | 240 | Afmaken start opdracht |
| Noud, Luca en Alex | 5-9-2017 | 50 | Taken verdeeld, overlegd |
| Alex | 14-09-2017 | 40 | Plan van aanpak |
| Luca en Noud | 03-10-2017 | 50 | Les |
| Noud | 04-10-2017 | 25 | Plan van aanpak |
| Luca | 05-10-2017 | 20 | Taken verdeling |
| Luca | 07-10-2017 | 60 | Bronnen opzoeken |
| Noud | 10-10-2017 | 20 | Voorwoord gemaakt |
| Noud | 11-10-2017 | 180 | Deelvraag 1 bronnen gezocht en begonnen aan hoofdstuk |
| Alex | 11-10-2017 | 50 | Voorwoord veranderd en indeling pws verbeterd. |
| Luca | 14-10-2017 | 60 | Bronnen opzoeken |
| Alex | 17-10-2017 | 20 | Opmaak bestand |
| Noud | 24-10-2017 | 50 | Deelvraag 1 uitwerken. |
| Alex | 24-11-2017 | 150 | Soorten Coderingtheorie |
| Luca | 25-11-2017 | 100 | Ethische paragraaf |
| Luca | 26-11-2017 | 100 | Ethische paragraaf |
| Alex | 27-11-2017 | 120 | Soorten coderingstheorie |
| Alex | 28-11-2017 | 60 | Soorten coderingstheorie |
| Noud | 28-11- 2017 | 180 | Hoofdstuk 1 oorsprong |
| Noud | 29-11-2017 | 50 | Literatuuronderzoek Hoofdstuk 1 |
| Luca | 1-12-2017 | 150 | Ethische paragraaf |
| Luca | 2-12-2017 | 120 | Ethische paragraaf |
|  |  |  |  |
| Noud | 5-12-2017 | 60 | Bibliografielijst opgesteld |
| Alex | 5-12-2017 | 100 | Hoofdstuk 2 |
| Luca | 8-12-2017 | 180 | Ethische paragraaf |
| Luca | 9-12-2017 | 60 | Ethische paragraaf |
| Luca | 11-12-2017 | 90 | Ethische paragraaf |
| Luca | 14-12-2017 | 120 | Ethische paragraaf |
| Luca | 15-12-2017 | 40 | Ethische paragraaf |
| Alex | 18-12-2017 | 50 | Cryptografie |
| Noud, Luca en Alex | 05-01-2018 | 240 | Praktische Opdracht |
| Noud | 08-01-2018 | 100 | Onderzoek uitwerken |
| Alex | 08-01-2018 | 200 | Samenvatting conclusie |
| Noud en Alex | 09-01-2018 | 60 | Bronnen beheren |
| Alex | 09-01-2018 | 90 | Bronnen Beheren Lob |
| Noud en Alex | 10-01-2018 | 50 | Dankwoord en Onderzoek |
| Noud | 13-01-2018 | 240 | Reflectie, Indeling, Onderzoek en Opmaak |
| Alex en Noud | 15-01-2018 | 100 | Opbouw en controle |
| Alex | 16-01-2018 | 120 | Cryptologie |
| Noud | 16-01-2018 | 160 | Cryptologie |
| Noud | 17-01-2018 | 50 | Controle |
| Luca | 17-01-2018 | 200 | Cryptologie, reflectie en controle |