2024/2025

Student: Simon Vercoutter

Klasnummer: 3

Project Titel: Symmetrische en Asymmetrische encryptie

Opzichter: E. Gheysen

Symmetrische en Asymmetrische encryptie

Provinciaal Technisch instituut West-Vlaanderen

Inhoudsopgave

Hoofdstuk 1

Geschiedenis

1.1 Oude en klassieke periode

1.1.1 Scytale (Sparta)

In het oude Griekenland, rond 400 v.C. gebruikten Spartaanse soldaten een hulpmiddel genaamd de scytale om berichten te coderen. Dit was een eenvoudige versleuteling waarbij de letters van een bericht op een spiraalvormig lint werden geschreven.

1.1.2 Caesarversleuteling (Romeinse Rijk)

Julius Caesar gebruikte een eenvoudig Caesar-cijfer. Bij deze methode wordt elke letter in het bericht vervangen door een letter een vast aantal posities verder in het alfabet. Deze methode was eenvoudig, maar effectief tegen vijanden.

1.2 Modernere cryptografie

1.2.1 Enigmacode (Tweede Wereldoorlog)

De Duitse Enigma-machine was een complex mechanisch en elektrisch apparaat dat berichten versleutelde door middel van een serie rotoren en schakelaars. De versleutelde berichten werden als onbreekbaar beschouwd tot Alan Turing en zijn team een manier vonden om de Enigma te kraken. Dit wordt gezien als een keerpunt in de oorlog.

1.2.2 Purple Machine (Japan)

Japan gebruikte tijdens de Tweede Wereldoorlog een soortgelijk apparaat, bekend als de Purple Machine, om communicatie te versleutelen. De Verenigde Staten wisten de machine te kraken, wat hen strategisch voordeel gaf in de oorlog tegen Japan.

1.3 Moderne computer cryptografie

1.3.1 DES (Data Encryption Standard)

In 1975 werd de DES-standaard ontwikkeld door IBM en goedgekeurd door de Amerikaanse overheid. DES was een symmetrische versleuteling dat de basis legde voor toekomstige standaarden in databeveiliging. Het gebruikte een 56-bits sleutel en bleef jarenlang de standaard.

1.3.2 RSA-algoritme

In 1977 ontwikkelden Ron Rivest, Adi Shamir en Leonard Adleman het RSA-algoritme, een van de eerste publieke systemen. RSA werd al snel een standaard voor beveiliging op internet en wordt tot op de dag van vandaag gebruikt in veel applicaties, van e-mail tot digitale handtekeningen.

Informatie van dit volledige hoofdstuk uit (1) → zie Hoofdstuk 5.

Hoofdstuk 2

Symmetrische encryptie

2.1 Algemene Werking

2.1.1 Algemene principe

Symmetrisch wil zeggen dat de ontvanger en de verzender dezelfde sleutel hebben om een boodschap te versturen.(De ontvanger moet het tegenovergestelde doen van de verzender om tot het originele bericht te komen.)
sleutels $S_{sen}(x) = S_{rec}(x)$ worden gecreëerd.
Met volgende eigenschappen:

1. $S_{sen}(Bericht) = Crypted$
2. $S_{rec}(Crypted) = Bericht$
3. $\Rightarrow S_{sen}(S_{rec}(Bericht) =$ Bericht
4. $\Rightarrow S_{sen}({S_{sen}}^{-1}(Bericht) =$ Bericht
5. $\Rightarrow S_{rec}(Bericht) = {S_{sen}}^{-1}(Bericht)$

Begrippen:

1. Bericht:
   De boodschap die getypt wordt door de verzender vóór versleuteling en gelezen door de ontvanger na ontcijfering.
2. Crypted:
   Het bericht dat versleuteld is door onleesbaarheid nadat de verzender zijn sleutel heeft gebruikt en voordat de ontvanger het bericht weer ontcijferd.

Informatie uit (2) → zie Hoofdstuk 5.

2.2 Voorbeelden van symmetrische versleuteling

2.2.1 Caesar-cijfer (100 V.C.)

Dit is een van de oudste methodes van geheime berichten verzenden waarvan men vanaf weet. Caesar-cijfer valt onder de soort Handcijfers of pen- en papiercodes.
Hiermee wordt er bedoeld dat deze soort code enkel snel kan worden ontsleuteld als er al meer informatie beschikbaar is gesteld (door voorgaande communicatie) of trager door brute rekenkracht
Deze versleutelingsmethode is nu beter bekend als Caesarrotatie of ROT als afkorting.
In tegenstelling tot veel andere methodes is ROT niet beschermd tegen cryptoanalyse de studie van encryptie.
De werking van deze methode gaat als volgt:

1. Er wordt een bericht en een willekeurig vast cijfer tussen 1 en 26 gekozen.
2. Dan worden alle letters in het bericht met het cijfer gedraaid.
   vb. ROT3 $\Rightarrow$ A $\rightarrow$ D

Om te ontcijferen worden volgende stappen ondernomen:

1. Er wordt een versleuteld bericht ontvangen.
2. Dan worden alle letters in het bericht met het vooraf afgesproken cijfer gedraaid met het complement van 26.
   vb. ROT3 $\Rightarrow$ A $\rightarrow$ D. Dan ROT(26 -3) $\Rightarrow$ ROT23 $\Rightarrow$ D $\rightarrow$ A
   (Met de voorwaarde dat het cijfer al eerder werd afgesproken.)

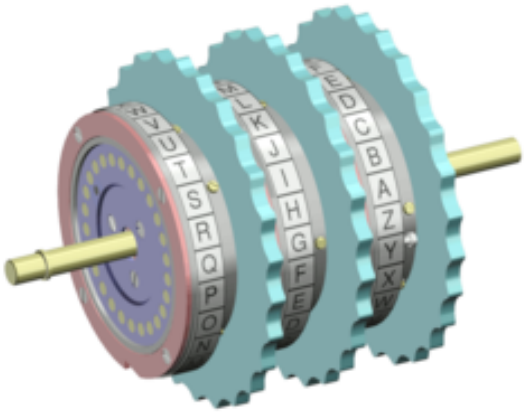
Informatie uit (3) → zie Hoofdstuk 5.

2.2.2 Enigma (1920)

Van deze machine werd voor een lange tijd gedacht dat de code onbreekbaar was, dankzij zijn toenmalige versleuteling.
De Enigma was een rotormachine: een elektromechanisch apparaat voor cryptologische doeleinden.
Het werd gebruikt in de Tweede Wereldoorlog door Duitse troepen om militaire berichten te verzenden naar het front om hen te informeren en hun plannen door te zenden. Pas in september 1938 werd de code gekraakt door de Poolse inlichtingendienst, maar werd verbeterd.
De werking van de verbeterde verzending gaat als volgt:

1. Beide partijen verbinden verschillende letters met elkaar op een stekkerbord en zet 3 rotoren op "A"
2. Er wordt een letter ingevoerd op het toetsenbord door de eerder verbonden letters verandert deze naar de verbonden letter.
3. De derde rotor verspringt met één. Hierdoor wordt de tweede rotor verduwt met een half-willekeurig aantal, net als de eerste rotor. ⇒ Als er nog eens dezelfde letter wordt getypt zal deze verschillen.
4. De eerste rotor verzet de reflector. Die op zijn beurt alle rotoren weer verzet, beginnend met de eerste en eindigend met rotor 3 (de rotoren draaien niet terug naar hun oorspronkelijke positie).
5. De derde rotor geeft dan het signaal door naar het stekkerbord dat het via de eerder verbonden kabels nogmaals verandert naar een andere letter dan de ingetypte letter.
6. Dit wordt herhaald tot alle letters van het bericht klaar zijn en wordt dan pas verzonden.

Om de boodschap te ontcijferen wordt de boodschap in tegengestelde zin ontcijferd door alle stappen van het versleutelen in omgekeerde bewerking te voltooien.
Maar ook dit ingewikkelde systeem heeft zijn beperkingen.
Waardoor Alan Turing de schijnbaar onbreekbare codes kon kraken.



Figuur 2.1: Rotoren in Enigma

Informatie uit (4) → zie Hoofdstuk 5.

2.2.3 Data Encrytion Standard (1975)

Data Encryption Standard of DES was voortgevloeid uit "Lucifer"deze 2 beveiligingssystemen waren zo ontwikkeld dat ze hun data opsplitsten in blokken van 64 bit (Lucifer ook in 48, 56 en 128) en een sleutel van 56 bit om deze informatie verborgen te houden. DES gebruikt 16 stappen om te encrypteren:

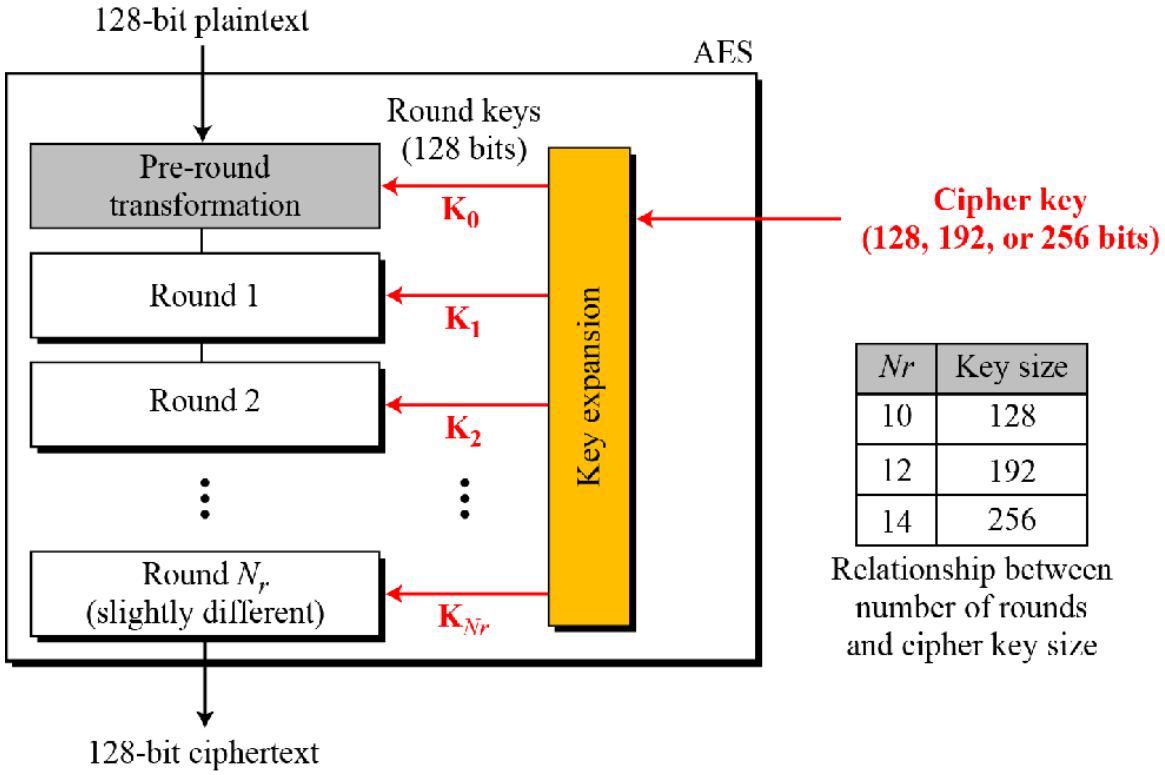
1. Er wordt begonnen met een 64 bit tekstbestand (8 tekens) in een hexadecimale vorm.
2. Alle tekens worden hierna gepermuteerd (vervormd) volgens een vast patroon naar binaire lijsten.
3. Dan worden de nieuwe bitstrings opgedeeld in de linkerhelft en de rechterhelft.
4. Hierop volgt de sleutelgeneratie, S1: een sleutel van 56 bit (7 tekens).
5. Daarna worden de linker- en rechterhelft verlengd met 16 bit, dus het totale bericht is nu 96 bit.
6. Dan wordt de sleutel S1 gebruikt om door XOR (Als beide inputs niet gelijk zijn = 1, indien wel gelijk is de output 0) te verkorten naar 64 bit (16 bit verkorten aan de linker- en rechterhelft).
7. De linker- en rechterhelft worden met elkaar verwisseld.
8. Herhaal de stappen(5-7) 16 keer.
9. Als laatst wordt er gepermuteerd met het inverse van de eerste keer dat er werd gepermuteerd.

Informatie uit (5) en (6) → zie Hoofdstuk 4

2.2.4 Advanced Encryption Standard (2001)

In de jaren 2000 kwamen cryptoanalysten meer te weten over DES die te gemakkelijk was te ontcijferen volgens de standaard waarvoor DES stond.
Hierdoor werd er gezocht naar een alternatief. IBM, RSA en Rijndael (Joan Daemen en Vincent Rijmen) waren de kandidaten. Waarvan Rijndael beter bekend als Advanced Encryption Standard (AES) won.
Om deze encryptie te volbrengen wordt het bericht als volgt bewerkt:

1. Het bericht dat verstuurd wordt , moet worden voorbereid door het op te splitsen in 4X4 tabellen.
2. Daarna wordt en sleutel toegevoegd met XOR net als met DES met een sleutel van 128 of 256 bit (16 of 32 tekens). In tegenstelling met DES die een sleutel van 56 bit (7 tekens) heeft.
3. De tabel wordt dan in een substitutiebox vervormd om patronen te verminderen.
4. Na deze stap worden de rijen van de tabel verschoven.
5. Hierna worden de kolommen ook verschoven (behalve de laatste ronde).
6. Stappen (3-5) worden 10 of 14 keer herhaald.



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Figure 2. General Design of AES Encryption Cipher [8]

Figuur 2.2: Schema AES

Informatie uit (7) → zie Hoofdstuk 5.

2.3 Gebruik en mogelijkheden

2.3.1 Gebruik van Symmetrische Encryptie

Symmetrische encryptie wordt vooral gebruikt voor situaties waarin snelheid en efficiëntie nodig zijn, zoals:

1. Bescherming van gegevens op opslagmedia
   Symmetrische encryptie wordt vaak gebruikt om harde schijven, USB-sticks en andere opslagapparaten te beveiligen. Hierdoor kunnen ongewenste personen geen toegang krijgen tot de gegevens zonder de juiste sleutel.
2. Beveiligde communicatie
   Voor het verzenden van grote hoeveelheden gegevens via beveiligde verbindingen (bijvoorbeeld VPN's) wordt symmetrische encryptie vaak gebruikt door de hoge versleutelingssnelheid.
3. Versleutelen van bestanden en archieven
   Bestanden en archieven (zoals ZIP-bestanden) kunnen met symmetrische encryptie beschermd en verkleind worden. Dit wordt vaak gebruikt om gevoelige informatie gemakkelijk te beveiligen.

2.3.2 Mogelijkheden

1. Koppeling met asymmetrische encryptie
   Om het probleem van veilige sleutelverdeling op te lossen, wordt symmetrische encryptie vaak gecombineerd met asymmetrische encryptie. Hierbij wordt de symmetrische sleutel versleuteld met een asymmetrische sleutel.
2. Gebruik van sterke sleutellengtes
   Om weerstand te bieden tegen brute-force-aanvallen, wordt aanbevolen om lange sleutels te gebruiken, vooral bij algoritmen als AES (meestal 128, 192, of 256 bits).
3. Regelmatige sleutelvernieuwing
   Voor extra veiligheid kunnen symmetrische sleutels periodiek vernieuwd worden, zodat een overeenkomende sleutel niet lang risico veroorzaakt.

2.4 Voordelen en beperkingen

2.4.1 Voordelen

1. Snelheid en efficiëntie
   Symmetrische encryptie is sneller dan asymmetrische encryptie, vooral bij grote hoeveelheden data.
2. Lagere rekenkrachtvereisten
   De verwerking vraagt minder van de CPU, wat voordelig is voor kleinere toestellen.
3. Eenvoudiger algoritme
   Symmetrische encryptie-algoritmen zijn eenvoudiger dan asymmetrische en daardoor makkelijker te gebruiken.

2.4.2 Nadelen

1. Sleutelbeheer
   Een van de grootste uitdagingen is het veilig delen van de sleutel tussen de verzender en ontvanger, omdat beide dezelfde sleutel moeten hebben.
2. Sleuteldistributie
   Bij communicatie over grote afstanden is het veilig doorsturen van de sleutel een potentieel risico, vooral wanneer het kanaal onbeveiligd is.
3. Geen authenticatie
   Symmetrische encryptie biedt op zichzelf geen verificatie. Hierdoor is het moeilijk om te controleren of de gegevens niet zijn vervormd of afkomstig zijn van een betrouwbare bron.

Informatie van onderdelen 2.3 en 2.4 uit (1) → zie Hoofdstuk 5.

Hoofdstuk 3

Asymmetrische encryptie

3.1 Algemene Werking

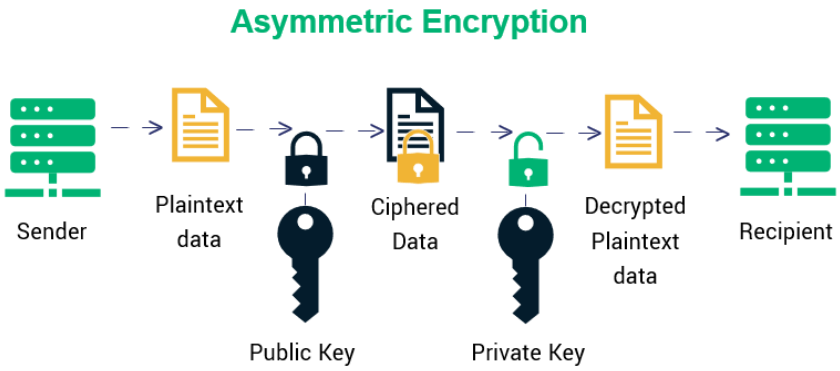
3.1.1 Algemene principe

Asymmetrisch betekend dat de ontvanger en de verzender elk 2 verschillende sleutels hebben:
een privé- en openbare sleutel. Met deze sleutels kunnen ze een algemeen getal bekomen die door niemand gekend kan worden omdat deze gebruik maken van de privésleutel.
Het algemeen principe werkt als volgt:

1. $E(S_{pub}, Bericht) = Crypted$
2. $D(S_{priv}, Crypted) = Bercicht$
3. $\Rightarrow D(S_{priv}, E(S_{pub}, Bericht)) = Bercicht$
4. S_{priv} is ongekend door anderen behalve de eigenaar
5. S_{priv} heeft geen verband met S_{pub}.
6. uit 4 en 5 $\Rightarrow S_{priv}$ kan moeilijk worden gevonden.

Ook lost asymmetrische encryptie al enkele problemen op die symmetrische encryptie had:

1. Sleuteluitwisseling
   Via symmetrische versleuteling kan er moeilijk een sleutel worden verstuurd die geheim blijft voor andere gebruikers. Waar asymmetrische algoritmes dan een privé sleutel en een openbare sleutel gebruiken om dit probleem op te lossen
2. Eigenheid
   Met symmetrische encryptie was het ook niet mogelijk om direct te herkennen van wie het bericht kwam. Asymmetrische versleuteling geeft de gebruiker een unieke handtekening om te herkennen van wie de gegevens afkomstig zijn.
3. Beveiliging
   Ook de algoritmes die achter asymmetrische beveiliging zitten zijn sterker dan symmetrische. Waardoor asymmetrische versleuteling veiliger is dan symmetrische versleuteling.



Figuur 3.1: Asymmetrische versleuteling

Informatie uit (8) → zie Hoofdstuk 5.

3.2 Voorbeelden van Asymmetrische versleuteling

3.2.1 Diffie-Hellman (1976)

Deze methode was een van de eerste asymmetrische encrypties. Dit principe was de eerste methode om het probleem op te lossen dat alle voorgaande hadden: er moest vooraf gecommuniceerd worden om een sleutel te maken zelfs als alle kanalen worden beluisterd.
De oplossing waarop Diffie en Hellman kwamen ging als volgt:

1. Er wordt door de verzender een priemgetal (P) en een basisgetal (C) gekozen.
2. Hierna wordt een privé getal (a) gekozen door de verzender.
3. A wordt dan berekend door $A = (C^a) mod(P)$.
4. Dan worden P,C en A verzonden naar de ontvanger.
5. Hierna wordt een privé getal (b) gekozen door de ontvanger.
6. B wordt dan berekend door $B = (C^b) mod(P)$.
7. Dan wordt B verzonden naar de verzender.
8. De verzender berekend dan $S = (B^a) mod(P)$.
9. De ontvanger berekend dan $S = (A^b) mod(P)$.
10. A,B,C en P zijn publiek gekend maar de sleutel S is enkel beschikbaar voor de verzender en ontvanger.

Informatie uit (9) → zie Hoofdstuk 5.

3.2.2 RSA (1977)

RSA of Rivest, Shamir en Adleman vernoemd naar de ontwerpers van deze asymmetrische beveiliging. Er werd gewerkt om een nieuwe manier te verwezenlijken. Deze methode wordt nog steeds veel gebruikt in versleutelde berichten en dataverkeer. Dit hebben ze zo aangepakt:

1. De verzender kiest twee verschillende grote priemgetallen. p_{1V} en p_{2V}
2. Van deze twee priemgetallen neem je het product : P_V
3. Dan wordt de indicator berekend $=\varphi(P_V) = (p_{1V} - 1)(p_{2V} - 1)$
4. Kies hierna een waarde S waarvoor geldt: $ggd(S; (p_{1V} - 1)(p_{2V} - 1)) = 1$
5. Daarna wordt S en P_V verzonden naar de ontvanger.
6. Dan wordt T gemaakt met voorwaarde: $T = \frac{k \cdot \varphi(P_V)+1}{S}$ $met\ k\ \in Z$

Om te versleutelen heeft de ontvanger S nodig maar deze kan enkel met T worden ontsleuteld.
Merk op dat dit per sleutel maar in één richting werkt.
⇒ Hetzelfde wordt gedaan bij de ontvanger naar de verzender met ander priemgetallen.

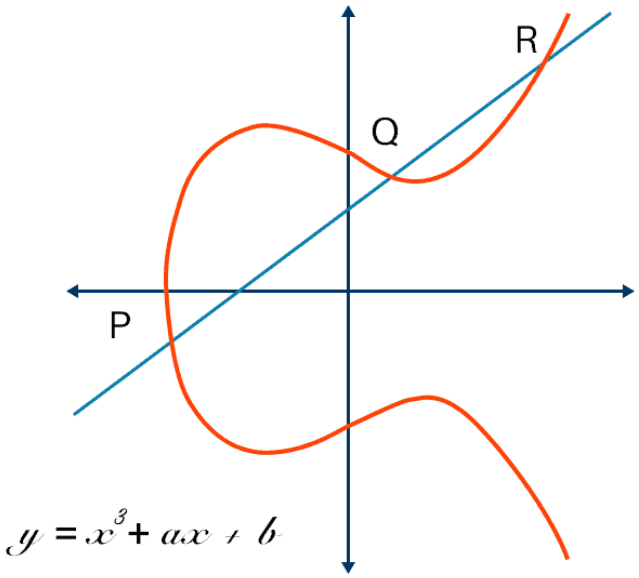
Informatie uit (10) → zie Hoofdstuk 5.

3.2.3 Elliptic Curve Crypography (1985)

Net zoals RSA is Elliptic Curve Cryptography (of ECC) asymmetrisch. dit werd door twee ontwerpers onafhankelijk uitgewerkt door Neal Koblitz en Victor S. Miller. Dit systeem werkte met kleinere getallen dan RSA met een gelijkaardige beveiligingsniveau met dus als voordeel dat het sneller werkte dan RSA.
Er wordt als volgt gewerkt:

1. Als voorbereiding wordt er een wiskundige kromme aangemaakt met vergelijking: $y^2 = x^3 + ax + b$ waarvan a en b grote constanten zijn. Deze kromme is openbaar.
2. Er wordt door de verzender dan een willekeurig getal P_V gekozen in het domein van de functie als privégetal.
3. Dan wordt P_V vermenigvuldigd met een openbaar gekend punt (x,y) op de curve om een openbaar getal O_V te verkrijgen.
4. Ook de ontvanger maakt een willekeurig getal P_O in het domein van de functie als privégetal.
5. Die wordt ook vermenigvuldigd met hetzelfde punt (x,y) om openbare O_O te bekomen.
6. De verzender vermenigvuldigt P_V *met* O_O om sleutel S uit te komen.
7. De ontvanger vermenigvuldigt P_O *met* O_V om hetzelfde getal voor sleutel S uit te komen.

De sleutel S kan niet gevonden worden omdat P_V en P_O door niemand gekend zijn dan de eigenaren.



Figuur 3.2: Elliptic Curve Crypography

Informatie uit (11) → zie Hoofdstuk 5.

3.3 Gebruik en mogelijkheden

3.3.1 Gebruik van Asymmetrische Encryptie

Asymmetrische encryptie wordt vooral toegepast in situaties waarin veilige communicatie en verificatie belangrijk zijn, zoals:

1. Beveiliging van internetcommunicatie
   Bij beveiligde websites (HTTPS) worden asymmetrische en symmetrische encryptie gecombineerd om een veilige verbinding op te zetten. De asymmetrische sleutels helpen bij het uitwisselen van de symmetrische sleutel.
2. Digitale handtekeningen
   Asymmetrische encryptie maakt het mogelijk om digitale handtekeningen te maken. Dit garandeert de authenticiteit en integriteit van documenten, omdat de afzender de handtekening met zijn private sleutel genereert.
3. Versleutelde e-mails
   E-maildiensten gebruiken asymmetrische encryptie om de vertrouwelijkheid van e-mails te waarborgen.
4. Authenticatie en identiteitsverificatie
   Bij toepassingen zoals toegangspassen, VPN's wordt asymmetrische encryptie gebruikt om gebruikersidentiteit te verifiëren zonder gevoelige sleutels te delen.

3.3.2 Mogelijkheden

1. Combinatie met symmetrische encryptie
   In beveiligde internetnetwerken zoals TLS wordt asymmetrische encryptie gebruikt voor sleuteluitwisseling, waarna een snelle symmetrische sleutel de dataversleuteling verwerkt.
2. Gebruik van lange sleutels
   Voor voldoende bescherming tegen brute-force-aanvallen worden lange sleutels aanbevolen. RSA-sleutels zijn minimaal 2048 bits, terwijl Elliptic Curve Crypography met sleutels van 256 bits veilig is.
3. Beheer van private sleutels
   De beveiliging van de private sleutel is essentieel. Deze moet veilig worden opgeslagen en beschermd, vaak met behulp van hardwarebeveiliging.
4. Toepassen van certificering (CA)
   Bij digitale certificaten kan een CA de authenticiteit van een publieke sleutel bevestigen, wat belangrijk is voor beveiligde communicatie.

3.4 Voordelen en beperkingen

3.4.1 Voordelen

1. Veilig sleutelverdeling
   Doordat de publieke sleutel openbaar mag zijn, is het delen van sleutels veel veiliger.
2. Authenticatie
   Asymmetrische encryptie ondersteunt digitale handtekeningen, waarmee de authenticiteit van de verzender en de eigenheid van het bericht gegarandeerd kunnen worden.
3. Geschikt voor verschillende systemen
   Asymmetrische encryptie is voordelig voor situaties waarin veel gebruikers toegang nodig hebben, omdat elke gebruiker zijn eigen paar sleutels kan hebben.

3.4.2 Nadelen

1. Rekenintensief
   Asymmetrische algoritmen vereisen veel meer rekenkracht dan symmetrische algoritmen, wat het minder geschikt maakt voor grootschalige versleuteling.
2. Trager dan symmetrische encryptie
   Door de complexe wiskundige operaties is asymmetrische encryptie trager, vooral bij grote hoeveelheden data.
3. Minder geschikt voor grote bestanden
   Om deze redenen wordt asymmetrische encryptie vaak alleen gebruikt voor de initiële sleuteluitwisseling, waarna symmetrische encryptie de data versleutelt.

Informatie van onderdelen 3.3 en 3.4 uit (8) → zie Hoofdstuk 5.

Hoofdstuk 4

Verificatie

4.1 Algemene werking

4.1.1 Algemeen principe

In tegenstelling tot symmetrische en asymmetrische encryptie is verificatie niet bedoeld om de data te versturen door te versleutelen en dan te laten ontcijferen door de ontvanger, maar zal daarentegen nagaan of de data bij het juiste adres is toegekomen.

1. Registreren = H($Gebruiker_Bedoeld$)
2. Ophalen = H($Gebruiker_Openen$)
3. als (Ophalen = Registreren)
4. ⇒ Geautoriseerd om de data te ontvangen

Informatie uit (12) → zie Hoofdstuk 5.

4.2 Voorbeelden van verificatie

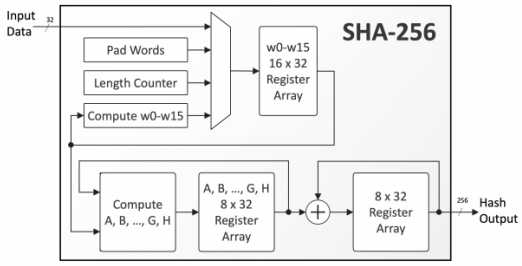
4.2.1 SHA-256 (2001)

Ook wel beter gekend als secure hash algorithm (of hashing). SHA-256 is niet symmetrisch noch asymmetrisch. Deze methode van beveiligen is bedoeld om te verifiëren of de juiste persoon het ontvangen heeft. Als er een hash van data wordt gevormd kan deze praktisch onmogelijk terug naar de oorspronkelijke data worden berekend door data verlies bij data langer dan 256 bit.
Bij deze methode zal de data die verstuurd worden en vergeleken worden met de eerder verstuurde hash die eerst werd verstuurd en daarna zou de data verkregen kunnen worden.
Volgende zaken waren belangrijk bij het ontwerpen van SHA-256:

1. Gegeven snel berekenen: zodat informatie snel kan worden gecontroleerd bij ontvanger.
2. Moeilijk terug te rekenen van de hash naar originele data.
3. Een kleine verandering brengt grote verandering in de hash. ⇒ snowball-effect

De hash wordt als volgt gemaakt:

1. Het bericht wordt verlengt tot het een lengte heeft van een veelvoud van 64 ASCII-tekens.
2. Er worden acht keer 32 initialisatiebits toegekend aan de data.
3. De gegevens worden verwerkt in blokken van 512 bits. Elk blok ondergaat 64 rondes van bewerkingen waarbij er XOR, AND, en bitverschuivingen worden gebruikt



Figuur 4.1: Hashing algoritme

Informatie uit (13) → zie Hoofdstuk 5.

Hoofdstuk 5

Conclusie

5.1 Eigen methode

Uit bovenstaande hoofdstukken kunnen de verschillende categorieën encryptie worden afgeleid om te vergelijken met een eigen gecreëerde versleuteling.

Die versleuteling is deels afgeleid van de Vigeneretabel.

1. Er wordt eerst een tabel gevormd van 64 bij 64 (rijen en kolom benoemd met een volgorde uit een lijst)
2. Daarin worden alle verschillende mogelijke combinaties van 2 tekens (uit een lijst van lengte 64) deels willekeurig geplaatst dat de eerste letter van de combinatie niet meer voorkomt in die kolom en rij.

Om te vercijferen zal de verzender volgende stappen ondernemen:

1. Eerst zal de verzender een begin letter kiezen (Volledig willekeurig).
2. Dan zal hij zoeken in de rij van de letter die hij wil vercijferen naar het eerste teken die hetzelfde is als de volledig willekeurig teken.
3. Daarna zal deze stap herhaald worden met te zoeken in de kolom van het laatste teken van de vorige beurt te zoeken .
4. Daarna zal het laatste tekens van de vorige zoekopdracht weer gezocht worden als eerste teken in de rij van de volgende letter die de verzender wil vercijferen.
5. Stappen 3-4 worden herhaald tot het hele bericht is versleuteld.

om te ontsleutelen worden deze stappen ondernomen:

1. De ontvanger doorzoekt de volledige tabel (dezelfde tabel als de verzender) naar de combinatie van de eerste 2 tekens uit het versleutelde bericht die de ontvanger ontvangt.
2. Wanneer deze gevonden is zal hij de rijbenoeming opschrijven.
3. Dan zal dit herhaald worden de 2 bekeken tekens met 1 opgeschoven dus het 2de en 3de teken worden gezocht in de tabel en dan zal daarvan de kolomnaam worden genoteerd.
4. Herhaal stappen 1-3 met telkens de beide tekens die bekeken worden met 1 naar rechts op te schuiven

voorbeeld:

\	A	B	C	D
A	DB	AC	CC	BB
B	BC	DC	AB	CA
C	CD	BD	DD	AD
D	AA	CD	BA	DA

1. Het bericht DCBA wordt verstuurd naar persoon B.
2. met de sleutel wordt in rij D gezocht naar het vakje dat begint met "B" $\Rightarrow$ "BA"
3. De laatste letter van "BA" is "A" dus zoek in de kolom C (de 2de letter) naar "A" $\Rightarrow$ "AB".
4. De laatste letter van "AB" is "B" dus zoek in de rij B (de 3de letter) naar "B" $\Rightarrow$ "BC".
5. De laatste letter van "BC" is "C" dus zoek in de kolom A (de 4de letter) naar "C" $\Rightarrow$ "CD".
6. als alles wordt samengevoegd wordt het verzonden bericht "$BABCD$".
   Merk op als er wordt begonnen met "A" in plaats van "B" dan wordt het bericht "$ACCAA$".

5.2 besluit

als de sleutel via een andere methode wordt doorgestuurd naar de ontvanger zal dit resulteren in een symmetrische versleuteling. Maar het doel was om dit volledig in 1 methode te verwerken waar beide sleutels niet gelijk zijn.
Dit zou er voorzorgen dat het niet tot bovenstaande technieken behoord maar nog een andere methode vergen.

Hoofdstuk 6

Bibliografie

6.1 Informatiebronnen

1. Wikipedia-bijdragers. (2024, oktober 15). Cryptografie. Wikipedia. https://nl.wikipedia.org/wiki/Cryptografie
2. Van Zanten, M. A. (2011). Opportunities to learn offered by primary school mathematics textbooks in the Netherlands. https://doi.org/10.33540/81. Cursus Discrete Wiskunde 2011-2012.
3. Wikipedia-bijdragers. (2024, 19 februari). Caesarcijfer. Wikipedia. https://nl.wikipedia.org/wiki/Caesarcijfer
4. Wikipedia-bijdragers. (2024, juni 28). Enigma (codeermachine). Wikipedia. https://nl.wikipedia.org/wiki/Enigma_(codeermachine)
5. Wikipedia-bijdragers. (2023, 27 oktober). Data Encryption Standard. Wikipedia. https://nl.wikipedia.org/wiki/Data_Encryption_Standard
6. http://www.itl.nist.gov/fipspubs/fip46-2.htm
7. Wikipedia-bijdragers. (2024e, augustus 30). Advanced Encryption Standard. Wikipedia. https://nl.wikipedia.org/wiki/Advanced_Encryption_Standard
8. Wikipedia-bijdragers. (2024, februari 18). Diffie-Hellman-sleuteluitwisselingsprotocol. Wikipedia. https://nl.wikipedia.org/wiki/Diffie-Hellman-sleuteluitwisselingsprotocol
9. Wikipedia-bijdragers. (2024, mei 8). RSA (cryptografie). Wikipedia. https://nl.wikipedia.org/wiki/RSA_(cryptografie)
10. Wikipedia contributors. (2024, 24 september). Elliptic-curve cryptography. Wikipedia. https://en.wikipedia.org/wiki/Elliptic-curve_cryptography
11. Wikipedia-bijdragers. (2023, juni 28). SHA-familie. Wikipedia. https://nl.wikipedia.org/wiki/SHA-familie
12. Wikipedia-bijdragers. (2023, oktober 1). Asymmetrische cryptografie. Wikipedia. https://nl.wikipedia.org/wiki/Asymmetrische_cryptografie

6.2 Afbeeldingen

1. Rotoren in Enigma:
   Wikipedia-bijdragers. (2024, juni 28). Enigma (codeermachine). Wikipedia. https://nl.wikipedia.org/wiki/Enigma_(codeermachine)
2. Schema AES:
   Sharma, D., Bhardwaj, A., Prasad, H., Kandpal, J., Saxena, A., Kant, K., & Verma, G. (2016). Design of Low Power and Secure Implementation of SBox and Inverse-SBox for AES. https://www.semanticscholar.org/paper/Design-of-Low-Power-and-Secure-Implementation-of-Sharma-Bhardwaj/083283e227cfac7d339e1cf5223bf4d7788bd9a5/figure/2
3. Asymmetrische versleuteling:
   Thakkar, J. (2020, 30 april). Types of Encryption: What to Know About Symmetric vs Asymmetric Encryption - InfoSec Insights. InfoSec Insights. https://sectigostore.com/blog/types-of-encryption-what-to-know-about-symmetric-vs-asymmetric-encryption/
4. Elliptic Curve Crypography:
   What is Elliptic Curve Cryptography? Definition & FAQs | VMware. (z.d.). https://www.vmware.com/topics/elliptic-curve-cryptography
5. Hashing algoritme:
   SHA-256. (z.d.). CAST. https://www.cast-inc.com/security/encryption-primitives/sha-256