Memo

Aan: Werkgroep normontwikkeling 'Pseudonimisatie dienst'
Van: Eric Verheul , namens het Ministerie van Volksgezondheid, Welzijn en Sport)
Datum: 26 september 2014

Onderwerp: Voorstel voor cryptografische specificatie pseudonimisering

---

1 Introductie

De Directie Zorgverzekeringen van het Ministerie van Volksgezondheid, Welzijn en Sport (hierna: VWS) maakt ten behoeve van de Risicoverevening gebruik van een pseudonimiseringsdienst. De pseudonimiseringsdienst wordt gebruikt om de zorgkosten te kunnen analyseren en te beheersen. Deze dienstverlening wordt opnieuw aanbesteed.

Door het gebruik van pseudonimisering binnen de risicoverevening wordt gerealiseerd dat er geen persoonsgegevens behoeven te worden verwerkt door betrokken partijen. De pseudonimisering dient daartoe te voldoen aan de volgende richtlijnen opgesteld door het College Bescherming Persoonsgegevens (CBP)[1]:

> "Bij toepassing van pseudonimisering is geen sprake van de verwerking van persoonsgegevens, indien aan de volgende voorwaarden is voldaan:
> a. Er wordt (vakkundig) gebruik gemaakt van pseudonimisering, waarbij de eerste encryptie plaatsvindt bij de aanbieder van de gegevens
> b. Er zijn technische en organisatorische maatregelen genomen om herleidbaarheid van de versleuteling ("replay back") te voorkomen;
> c. De verwerkte gegevens zijn niet indirect identificerend;
> d. In een onafhankelijk deskundig oordeel (audit) wordt voor aanvang van de verwerking en daarna periodiek vastgesteld dat aan de voorwaarden a, b en c is voldaan;
> e. De pseudonimiseringsoplossing dient op heldere en volledige wijze te zijn beschreven in een openbaar document, zodat iedere betrokkene kan nagaan welke garanties de gekozen oplossing biedt."

Deze memo is een open voorstel voor een specificatie van een (openbaar) cryptografisch algoritme en gerelateerde datastructuren conform met bovenstaande voorwaarden a. en e.. VWS is van plan deze specificatie deel uit te laten maken van de aanbesteding procedure voor de pseudonimiseringsdienst als een specificatie die kan worden toegepast door dienstverleners.

De hoop van VWS is dat deze specificatie wordt omarmd zodat de pseudonimisering dienstverlening een open markt wordt waarin organisaties eenvoudig gebruik kunnen maken van pseudonimiseringsdiensten en ook eenvoudig kunnen overstappen naar andere leveranciers.

---

[1] Bijvoorbeeld in 'Pseudonimisering risicoverevening', College Bescherming Persoonsgegevens, 6 maart 2007. Zie ook http://www.cbpweb.nl/downloads_uit/z2006-1382.pdf.

Deze memo is als volgt opgebouwd:

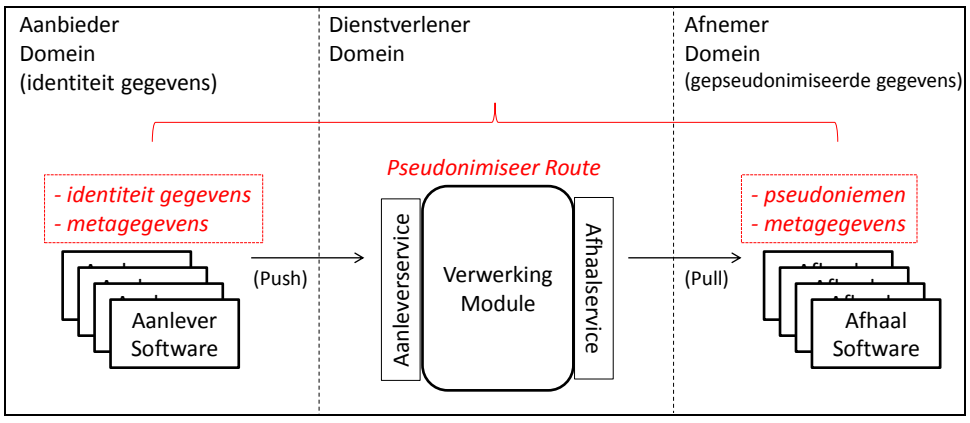
- Sectie 2 beschrijft de opzet van pseudonimisering in algemene zin, de daarbij gebruikte terminologie alsmede de reikwijdte van deze specificatie. In deze sectie wordt ook een eerste interpretatie gedaan van bovengenoemde CBP richtlijnen.
- Sectie 3 van de memo beschrijft, binnen de gestelde reikwijdte, de eisen die worden gesteld aan de pseudonimisering methode.
- Sectie 4 is de kern van deze memo. Hierin wordt in lijn met de gestelde eisen een open pseudonimisering methode gespecifieerd. Dit omvat cryptografische operaties en de onderliggende datastructuren. Ook worden in deze sectie minimale eisen gesteld aan het sleutelbeheer van de pseudonimiseringsdienst. Enerzijds is dit ingegeven vanuit de CBP richtlijnen en anderzijds om de portabiliteit onder pseudonimiseringsdiensten te bevorderen.
- In Sectie 5 wordt onderbouwd dat het gespecificeerde in Sectie 4 voldoet aan de eisen gesteld in Sectie 3.
- Sectie 6 bevat de referenties gebruikt in deze memo.

Deze memo bevat ook nog twee bijlagen. Bijlage A bevat voorbeeld berekeningen van de voorgestelde pseudonimisering methode. Bijlage B heeft een informatie karakter en bevat nadere informatie over het gebruik van Hardware Security Modulen (HSMs). Vanuit deze specificatie is het gebruik hiervan niet verplicht.

2 Pseudonimisering context, terminologie en specificatie reikwijdte

2.1 Pseudonimisering context en terminologie

Een pseudonimiseringsdienst onderkent *aanbieders* van gegevens en *afnemers* van gegevens. De pseudonimiseringsdienst treedt daarbij op als een centrale Trusted Third Party (*TTP*). Een aanbieder biedt de dienst een *Inputbestand* aan dat deze vervolgens omzet naar een gepseudonimiseerd *Outputbestand* voor de afnemer. In de praktijk gebruikt de aanbieder daarvoor *aanlever software* en de afnemer *afhaal software*. De gegevens worden verwerkt in een verwerking module bij de pseudonimiseringsdienst. Zie onderstaande figuur. In deze figuur worden de gegevens bewerkt door de aanbieder aangeboden (*push*) aan een (web)dienst bij de dienstverlener. Daar worden ze verwerkt en beschikbaar gesteld aan de afnemer. De afnemer kan ze vervolgens ophalen (*pull*). De bescherming van de uitgewisselde gegevens en de verbindingen tussen de partijen (Aanbieder, pseudonimiseringsdienst en Afnemer) vallen buiten de reikwijdte van deze specificatie.



Binnen de gegeven specificaties van inputbestanden wordt onderscheid gemaakt tussen *direct identificerende gegevens*, zoals voor- en achternaam en Burgerservicenummer (BSN), en *meta*

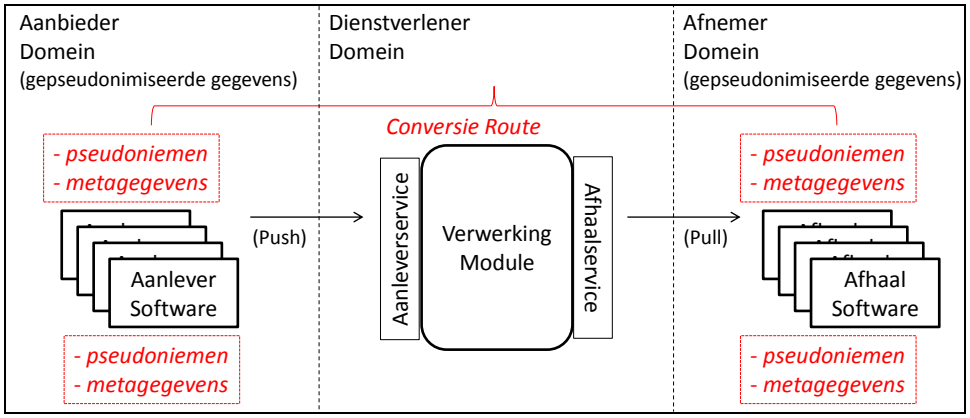
gegevens, i.e. gegevens *over* de betrokkene. Zoals de naam ook aangeeft maken direct identificerende gegevens directe identificatie van de betrokkene in beginsel mogelijk. Ook metadata, indien voldoende rijk, kan identificatie van de betrokkene mogelijk maken. De bedoeling van pseudonimisering is dat het beide potentiele vormen van identificatie in praktische zin voldoende uitsluit.

Zoals geduid in bovengenoemde CBP richtlijnen voert de aanbieder van de gegevens een eerste 'encryptie' uit op de direct identificerende gegevens; dit leidt tot *prematuur pseudoniemen*. Verder 'dunt' de aanbieder van de gegevens de metagegevens zodanig uit dat het risico op identificatie van betrokkenen bij de afnemer 'voldoende' wordt gereduceerd. Dit omvat zowel het risico op identificatie met betrekking tot publiek beschikbare registraties (e.g. telefoonboeken) als met betrekking tot niet-publieke registraties waar de afnemer over beschikt.

Een geboortedatum kan bijvoorbeeld worden uitgedund tot geboortejaar. De zodanig bewerkte gegevens worden veilig verstuurd naar de pseudonimiseringsdienst die een tweede 'encryptie' uitvoert op de eerder bewerkte direct identificerende gegevens. Deze tweede 'encryptie' leidt tot de *pseudoniemen* en de zo ontstane dataverzameling is het gepseudonimiseerde outputbestand. Dit bestand wordt dan veilig aangeboden aan de afnemer van de dienst. De ontwikkelde pseudoniemen zijn specifiek voor een betrokkene bij de afnemer. Dat wil zeggen dat het pseudoniem bij afnemer X verschillend is van dat bij afnemer Y. Het totaal van pseudoniemen bij een afnemer van een bepaald type heet diens *pseudoniem domein*. Het *type* van het pseudoniem wordt bepaald door de identificerende gegevens die de basis voor het pseudoniem vormde. Deze identificerende gegevens moeten per type eenduidig moeten zijn vastgelegd omdat verschillende representaties zullen leiden tot verschillende pseudoniemen. In dit document leggen we twee pseudoniem typen vast: Adres pseudoniemen en BSN pseudoniemen.

Een fundamentele functionele eigenschap van pseudoniemen is *compatibiliteit*, i.e. dat zij onafhankelijk moeten zijn van de aanbieder. Dat wil zeggen, als aanbieder A een pseudoniem van persoon P laat afleveren bij een afnemer X dan levert dit hetzelfde pseudoniem op als wanneer dit pseudoniem van persoon P was aangeleverd aan X door een andere aanbieder B.

Behalve de boven beschreven pseudonimisering van identificerende gegevens kan een pseudonimiseringsdienst ook *domein conversie* ondersteunen. Hierbij wordt een eerder gepseudonimiseerd bestand voor afnemer A geconverteerd naar het domein van een andere afnemer B. Bij een domein conversie wordt het pseudoniem van een betrokkene in domein A aldus naar diens pseudoniem in domein B vertaald. Ook bij een domeinconversie dient het risico op identificatie voldoende gereduceerd te zijn. In de praktijk zal de opzet van een domeinconversie erg lijken op die van een pseudonimisering. Zie onderstaande figuur.



Fundamentele beveiligingseigenschappen van pseudoniemen zijn:

1. *Pseudoniemen niet omkeerbaar*
   Voor partijen buiten de pseudonimiseringsdienst mag het in cryptografische zin niet mogelijk zijn om een pseudoniem om te keren, i.e. terug te brengen tot de direct identificerende gegevens waar het op gebaseerd is. Ook voor de pseudonimiseringsdienst mogen pseudoniemen niet omgekeerd kunnen worden, hoewel deze bescherming daar in lijn met CBP richtlijn b. ook mede gebaseerd mag zijn op organisatorische maatregelen.
2. *Compartimentering van afnemer domeinen*
   Bij verschillende afnemers krijgt een persoon ook verschillende pseudoniemen. Cryptografisch moet worden afgedwongen dat deze verschillende pseudoniemen niet te converteren zijn buiten de pseudonimiseringsdienst om. Een conversie kan onder bepaalde voorwaarden wel worden geboden door de pseudonimiseringsdienst die hiervoor dan over geheim cryptografisch sleutelmateriaal moet beschikken. Zonder dit sleutelmateriaal mag domeinconversie aldus niet mogelijk zijn. De voorwaarden voor conversie liggen buiten de reikwijdte van dit memo.
3. *Compartimentering van domein typen*
   Het cryptografische proces waarop een pseudoniem van een bepaald type tot stand komt moet voorkomen dat daarmee ook een pseudoniem van een ander type kan worden gevormd. Deze eigenschap komt er eenvoudig gesteld op neer dat als een aanbieder via een pseudonimiseringsdienst in staat wordt gesteld om Adres pseudoniemen te leveren hij dit proces niet moet kunnen manipuleren om BSN pseudoniemen te leveren. Complicerend bij deze eigenschap is dat de pseudonimiseringsdienst de oorspronkelijke adres input niet kan controleren omdat deze reeds versleuteld worden aangeboden door de aanbieder.

Het is van belang dat de genoemde, eerste 'encryptie' bij de aanbieder van de gegevens door alle aanbieders op compatibele wijze wordt uitgevoerd. Indien de ene aanbieder een andere 'encryptie' uitvoert dan een andere kan nooit tot compatible pseudoniemen worden. De eerste 'encryptie' zal worden uitgevoerd door software die wijd verspreid zal zijn. Daarom is het niet zinvol de eerste 'encryptie' een operatie te laten zijn gebaseerd op geheime cryptografische sleutels. Daarom is in de staande praktijk de eerste 'encryptie' gebaseerd op een zogenaamde *secure hash* operatie. Een dergelijke operatie $\mathcal{H}(.)$ zet een (lange) string S om in een string van vaste lengte $\mathcal{H}(S)$. Daarbij is het niet mogelijk vanuit de hash uitkomst de string S terug te berekenen anders dan door alle mogelijke input strings uit te proberen. Een dergelijke berekening heet een *brute-force*. Het succes van een *brute-force* berekening is sterk afhankelijk van de complexiteit ('entropie') van de oorspronkelijke string S.[2] De praktijk is daarom dat de eerste encryptie bij de aanbieder slechts een beperkte beveiliging biedt en dat de echte cryptografische beveiliging vanuit verdere encryptie bij de pseudonimiseringsdienst moet komen.

We merken nog op dat in de context van pseudonimisering ook het begrip 'omkeerbaar' (*reversible*) pseudoniem' bestaat. Dit is een pseudoniem type waarbij het functioneel juist wel mogelijk is om terug te keren naar de direct identificerende gegevens waar het op gebaseerd is, dat wil zeggen zonder een brute-force berekening te hoeven uitvoeren. Een omkeerbaar pseudoniem bevat daartoe een versleutelde versie van deze direct identificerende gegevens die door de pseudonimiseringsdienst ontsleuteld kan worden. Omkeerbare pseudoniemen worden bijvoorbeeld toegepast bij medische onderzoeken waarbij het mogelijk moet zijn de betrokkene (patiënt) in kwestie te kunnen waarschuwen als het onderzoek daar aanleiding toe geeft. Omkeerbare pseudoniemen lijken op gespannen voet te staan met de CBP pseudonimisering richtlijnen.

2.2 Reikwijdte van de specificatie in dit memo

De reikwijdte van de specificatie in dit memo is beperkt tot de twee genoemde cryptografische operaties en betrokken data structuren bij de aanbieder en bij de pseudonimiseringsdienst. De genoemde 'uitdun' operatie en beheer van de pseudonimiseringsdienst buiten de reikwijdte van deze specificatie. Ook de bescherming van de uitgewisselde gegevens en de verbindingen tussen de partijen (Aanbieder, pseudonimiseringsdienst en Afnemer) vallen buiten de reikwijdte. In beperkte zin stelt deze specificatie eisen aan het beheer van het cryptografisch sleutelmateriaal van de pseudonimiseringsdienst in Sectie 4.5. Enerzijds is dit ingegeven vanuit de CBP richtlijnen en anderzijds om de portabiliteit onder pseudonimiseringsdienst te bevorderen.

Feitelijk is het niet nodig om de betrokken data structuren bij de aanbieder in deze memo te specificeren omdat deze alleen binnen de pseudonimiseringsdienst zullen worden gebruikt. Om redenen van duidelijkheid en volledigheid zijn deze data structuren wel opgenomen in dit memo.

Deze specificatie beschrijft slechts twee soorten pseudoniemen (BSN en Adres) maar biedt ruimte om pseudoniemen op basis van andere identificerende gegevens eenvoudig toe te voegen. Deze specificatie beschrijft geen omkeerbare pseudoniemen maar biedt wel ruimte deze eenvoudig alsnog toe te voegen.

---

[2] Ter illustratie, er zijn slechts 100 miljoen (10^8) mogelijkheden voor een Burgerservicenummer (BSN). Dat betekent dat op basis van een secure hash waarde van een BSN een geautomatiseerde zoektocht van 100 miljoen mogelijkheden volstaat om het BSN terug te vinden. Een dergelijke zoektocht duurt enkele minuten op een krachtige PC.

3 Eisen gesteld aan deze specificatie

In onderstaande tabel zijn de eisen opgesteld waaraan de cryptografische operaties en betrokken data structuren bij de aanbieder en bij de pseudonimiseringsdienst moeten voldoen.

#	Vereiste	Toelichting
1.	*Conformiteit met de CBP richtlijnen*	Pseudoniemen moeten, in lijn met CBP richtlijn a., gebaseerd zijn op een eerste 'encryptie' bij de aanbieder en op een tweede 'encryptie' bij de pseudonimiseringsdienst.
2.	*Pseudoniemen niet omkeerbaar*	Nadere explicitering van CBP richtlijn b., zie Sectie 2.1.
3.	*Compartimentering van afnemer domeinen*	Nadere explicitering van CBP richtlijnen a. en b.. Zie Sectie 2.1.
4.	*Compartimentering van domein typen*	Nadere explicitering van de CBP richtlijnen a. en b.. Zie Sectie 2.1.
5.	*Authenticiteit van pseudoniemen beschermd*	De authenticiteit van de pseudoniemen moet kunnen worden vastgesteld door de pseudonimiseringsdienst. Dit is met name van belang als de pseudonimiseringsdienst domein conversie gaat uitvoeren, daarbij moet de pseudonimiseringsdienst zekerheid hebben dat hij (gevoelige) cryptografische operaties gaat uitvoeren op pseudoniemen die van *hemzelf* afkomstig zijn en niet op mogelijk gemanipuleerde data.
6.	*Cryptografische stand der techniek*	Dit wordt ook geëist in Artikel 13 van de Wet bescherming persoonsgegevens. Hiervoor zullen wij [NIST] als basis hanteren.
7.	*Migreerbaarheid naar andere cryptografische sleutels*	Het moet mogelijk zijn om op een gegeven moment op andere cryptografische sleutels over te gaan, bijvoorbeeld omdat ze verouderd zijn of omdat er een beveiligingsincident is geweest bij de pseudonimiseringsdienst.
8.	*Migreerbaarheid naar andere algoritmen*	Het moet mogelijk zijn om op een gegeven moment op andere cryptografische algoritmen over te gaan, bijvoorbeeld omdat ze verouderd zijn en te weinig beveiliging bieden of omdat ze gebroken zijn.
9.	*Overdraagbaarheid ('portability')*	Het moet mogelijk zijn om op een gegeven moment de pseudonimiseringsdienst over te dragen aan een andere pseudonimiseringsdienst met minimale inspanning en impact voor de aanbieders en afnemers. Dit zou vorm gegeven kunnen worden doordat de huidige pseudonimiseringsdienst de betrokken

		cryptografische sleutels veilige overdraagt aan de nieuwe en dat de aanbieders en afnemers software van de nieuwe pseudonimiseringsdienst gaan gebruiken.
10.	*Compactheid* Pseudoniemen moeten zo min mogelijk data (‘karakters) benutten.	Gepseudonimiseerde bestanden omvatten vaak grote populaties met dus vele pseudoniemen. Een langer pseudoniem betekent een groter bestand.
11.	*Representatie in printbare vorm*	Pseudoniemen moeten ook manueel kunnen worden bewerkt, e.g. in Excel. Om ongewenste neveneffecten te voorkomen, gaan we slechts uit van karakters die printbaar zijn.
12.	*(Eenvoudig) interpreteerbaar*	Pseudoniemen moeten ook handmatig kunnen worden verwerkt. Het is belangrijk dat de afnemer snel kan vaststellen wat voor type pseudoniem (en voor welk domein) hij aan het bewerken is. Ook moet de pseudonimiseringsdienst in detail kunnen achterhalen welke sleutels en algoritmen zijn toegepast voor als er domeinconversie moet worden toegepast.

4 Specificatie van de cryptografische operaties en onderliggende datastructuren

4.1 Inleiding

In deze sectie zullen de volgende twee zaken in detail worden gespecificeerd:

- de ‘eerste encryptie’ gebaseerd op een Adres of BSN gevormd door de Aanbieder, en
- het feitelijke pseudoniem gebaseerd op een Adres of BSN gevormd door de pseudonimiseringsdienst en bedoeld voor de Afnemer.

Om redenen van flexibiliteit en toekomstvastheid zullen wij deze beschrijving maken middels een bovenliggende flexibele, generieke pseudonimiseer datastructuur. Deze opzet maakt het eenvoudig mogelijk om aanvullingen toe te voegen en maakt toekomstige aanpassingen eenvoudig mogelijk. Als aanvullingen kan men denken aan pseudoniemen gebaseerd op andere direct identificerende gegevens, e.g. geboorteplaats in combinatie met geboortejaar, en andere typen pseudonimiseer structuren zoals omkeerbare pseudoniemen.

Hiertoe introduceren wij het begrip pseudonimiseer datastructuur (PD). Geassocieerd met een PD zijn twee attributen:

- PD-type,
- PD-inputsoort.

Het PD-type geeft de rol aan die de PD speelt bij een pseudonimiseringsdienst. In deze specificatie zullen wij slechts twee PD typen definiëren:

- de ‘eerste encryptie’ gevormd door de Aanbieder (prematuur pseudoniemen, feitelijk secure hash waarden van identificerende gegevens aangegeven met PD-type ‘H’), en
- het feitelijke pseudoniem gevormd door de pseudonimiseringsdienst en bedoeld voor de Afnemer (feitelijk een symmetrisch versleutelde hash waarde aangegeven met PD-type ‘P’).

In aanvulling hierop zou een type ‘R’ kunnen worden geïntroduceerd corresponderende met een omkeerbaar (‘*reversible’*) pseudoniem. Dit type zullen wij niet verder specificeren.

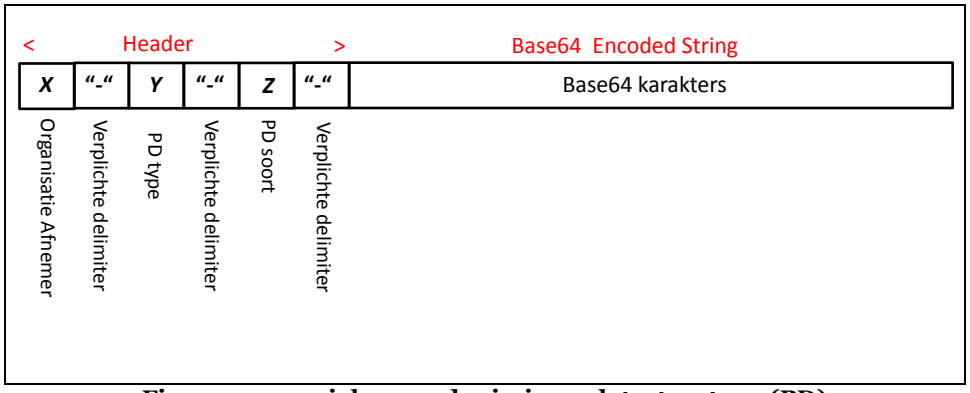
Het PD- inputsoort geeft aan welke identificerende gegevens van een betrokkene de basis vormen voor het PD. In deze specificatie zullen wij slechts twee PD inputsoorten definiëren:

- Inputsoort Adres ‘A’
  Gebaseerd op het woonadres en de postcode van betrokkene
- Inputsoort BSN ‘B’
  Gebaseerd op het Burgerservicenummer (BSN) van de betrokkene.

Voor elke combinatie van een soort en een type zullen wij de bijbehorende PD specificeren. Om redenen van toekomstvastheid suggereren wij ook dat bij elke uitbreiding van deze specificatie met een soort of een type ook alle combinaties worden gedefinieerd.

4.2 Generieke vorm van PD

De generieke vorm van een PD is een string aangegeven in onderstaande figuur. Een PD bestaat uit een Header en een Base64 string. De Header is opgebouwd uit drie parameters gescheiden door het delimiter symbool “-”. Merk op dat het symbool “-” geen onderdeel uitmaakt van de Base64 karakterset.



Figuur 1: generieke pseudonimiseer datastructuur (PD)

Ter illustratie, een voorbeeld van een PD string zou kunnen zijn:

"ZI-P-B-AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7".

In onderstaande tabel worden de parameters X, Y, Z en Base64 uitgelegd. Daarbij merken wij (ASCII) strings zoals als gebruikelijk zullen representeren met aanhalingstekens "" als delimiters. Voorbeelden: "", "abc", "064148737". Omdat geen van de string types die wij gebruiken 'white spaces' bevat (zoals spaties) zullen wij strings die over verschillende regels lopen zonder verdere signalering afbeelden. Strings bestaande uit hexadecimale karakters {0-A} zullen wij ook wel arrays noemen. Wij zullen deze onderscheiden van reguliere strings doordat wij geen aanhalingstekens "" gebruiken; wij zullen ook eisen dat hun lengte altijd een even getal is omdat wij met een array van hexadecimale karakters feitelijk een array van bytes representeren. En 1 byte wordt gerepresenteerd als twee hexadecimale karakters. Voorbeeld van een array van hexadecimale karakters: 012345678900

Parameter	Type	Uitleg	Voorbeelden
X	*Afnemer ID* • Alfabetische string van maximaal 64 karakters. • Indicatie van de Afnemer organisatie, i.e. van het Afnemer domein.	De pseudonimiseringsdienst is vrij om deze zelf te kiezen.	"ZI" van Zorg Instituut Nederland.
Y	*PD type* • Alfabetische string van maximaal 16 karakters. • Indicatie van PD type.	Pseudonimiseringsdiensten die deze specificatie volgen, kiezen hiervoor alleen typen die in de specificatie zijn vastgelegd. Op dit moment zijn dit "H" voor prematuur pseudoniem en "P"	"H", "P"

Parameter	Type	Uitleg	Voorbeelden
		voor pseudoniem.	
Z	*PD inputsoort* • Alfabetische string van maximaal 16 karakters. • Indicatie van de input waarop het PD is gebaseerd.	Pseudonimiseringsdiensten die deze specificatie volgen, kiezen hiervoor alleen soorten die in de specificatie zijn vastgelegd. Op dit moment zijn dit 'A' voor adres-string en 'B' voor BSN-string.	"A", "B"
Base64	*Base64 string* • String bestaande uit Base64 karakters, i.e. in {a-z, A-Z, 0-9, + en /}. • De lengte is afhankelijk van het PD type. • Maximale lengte is 1024 karakters.	• De eerste byte van de decoded Base64 string is altijd een versie nummer, i.e. een getal dat minimaal 1 is en maximaal 255. • Het versie nummer samen met de type indicatie moet vastleggen hoe de rest van de decoded Base64 string is opgebouwd. Vergelijk Secties 4.3.1en 4.3.2.	"AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7"

Tabel 1: Header specificatie

4.3 PD type soorten versie 1

In deze sectie zullen wij versie 1 van de twee PD typen 'H' (Hash) en 'P' (Pseudoniem) definiëren. Concatenatie van strings (en arrays van hexadecimale karakters) zullen wij aangeven met een plus ('+') teken. Voorbeelden:

- "abc" + "064148737"="abc064148737".
- 0A12+23BB=0A1223BB (dit wijkt dus af van de reguliere optelling).

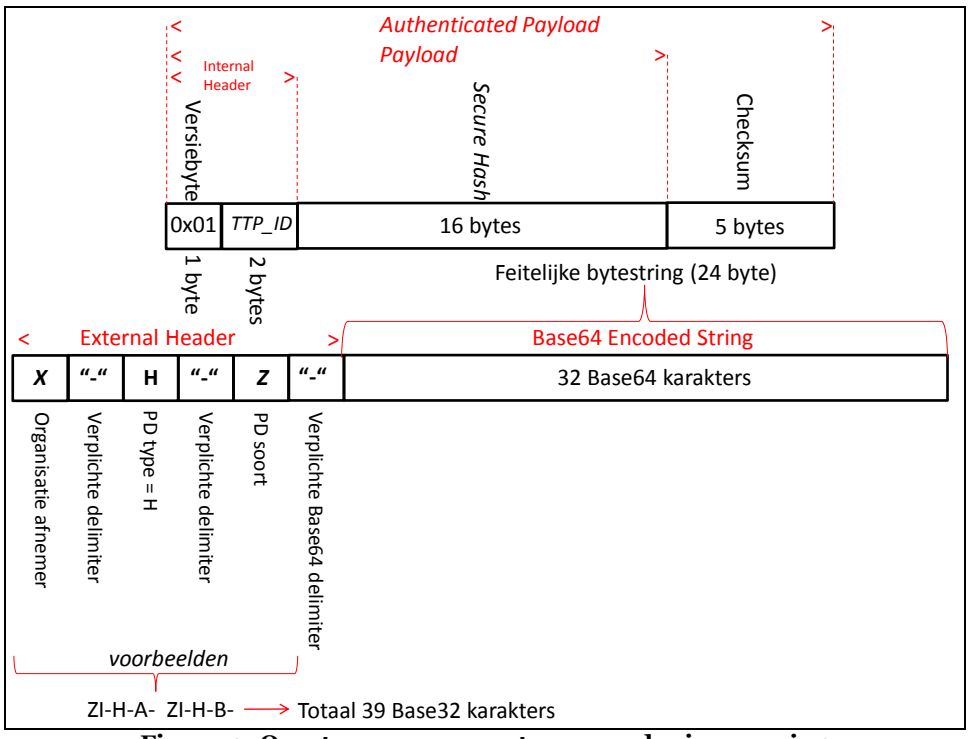
Verder zullen wij de volgende operaties/functies gebruiken.

Operatie	Uitleg	Voorbeeld
AES(K,M)	Deze operatie heeft als input: • een sleutel K die bestaat uit een array 32 hex karakters (16 bytes), • een plaintext M die bestaat uit een array 32 hex karakters (16 bytes). De output is de AES versleuteling C van M onder K in ECB mode. Dit is ook een array van 32 hex karakters (16 bytes). Zie [AES].	K=2B7E151628AED2A6ABF7158809CF4F3C M=6BC1BEE22E409F96E93D7E117393172A C=3AD77BB40D7A3660A89ECAF32466EF97
Base64E(A)	Deze operatie heeft als input een array A van bytes gerepresenteerd als hexadecimale karakters en levert als output de Base64 codering daarvan. Zie [BASE].	Base64E(0102FF)=“AQL/”
Base64E(S)	Deze operatie heeft als input een (correcte) Base64 als output een array van bytes gerepresenteerd als hexadecimale karakters. Zie [BASE].	Base64D(“AQL/”)=0102FF
GetH(PP)	Deze operatie heeft als input een alfanummerieke string PP. Deze operatie stelt vast dat de string opgebouwd is zoals in Figuur 1 en Tabel 1. Als dit niet zo is, wordt een foutmelding teruggegeven. Anders wordt als output de gebruikte header van de vorm X8Y8Z8 zoals aangegeven in Figuur 1.	GetH(“ZI-P-B-AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7”)= “ZI-P-B-”.
GetB64(PP)	Deze operatie heeft als input een alfanummerieke string PP. Deze operatie stelt vast dat de string opgebouwd is zoals in Figuur 1 en Tabel 1. Als dit niet zo is, wordt een foutmelding teruggegeven. Anders wordt als output de Base64 string teruggegeven zoals aangegeven in Figuur 1.	GetB64(“ZI-P-B-AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7”)=“AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7”.
SHA256(A)	Deze operatie heeft als input een array A van bytes gerepresenteerd als hexadecimale karakters en levert als output de SHA256 output zoals gespecificeerd in [SHA256], i.e. 32 bytes gerepresenteerd als hexadecimale karakters.	SHA256(616263)= BA7816BF8F01CFEA414140DE5DAE2223B00361A396177A9CB410FF61F20015AD
HMAC(K,M)	Deze operatie heeft als input: • een sleutel K die bestaat uit een array van 64 hex karakters (32 bytes), • een plaintext M die bestaat uit een array A van bytes gerepresenteerd als hexadecimale karakters. De output is de HMAC H onder K in ECB mode. Dit is een array van 64 hex karakters (32 bytes). Zie [HMAC].	K=000102030405060708090A0B0C0D0E0F 000102030405060708090A0B0C0D0E0F M=616263 H=10CC8E7C6473C85058703F0E0C7699BE A3B820DA3851D558E1D8DE1DAA6001CC
Tr(n,S)	Deze operatie heeft als input een natuurlijk getal n en een array A van hex karakters. De output bestaat uit de n meest significante array elementen.	Tr(4, 123456)=1234

4.3.1 PD soort ‘H’ versie 1

Vergelijk de generieke PD structuur aangegeven in Figuur 1. Zoals aangegeven in Sectie 4.2 is de pseudonimiseringsdienst vrij om de X parameter in te vullen in lijn met de specificaties in Tabel 1. Een PD van type “H”, versie 1 heeft parameter Y=“H”. De eerste byte in de decoded Base64 string is de versie byte, i.e. 01. De parameter Y kan in deze specificatie slechts de waarden “A” (input adres-inputstring) en “B” (input BSN-string) hebben. Een PD van dit type heet een *prematuur pseudoniem*.

De decoded Base64 string bestaat behalve uit de versie byte, uit een TTP_ID van twee bytes corresponderende met een unieke referentie van de TTP wiens software het prematuur pseudoniem heeft gevormd. Verder bestaat een secure hash waarde van 16 bytes en een checksum. De nadere betekenis en constructie hiervan zal onderstaand worden uitgelegd.



Figuur 2: Opzet van een prematuur pseudoniem versie 1

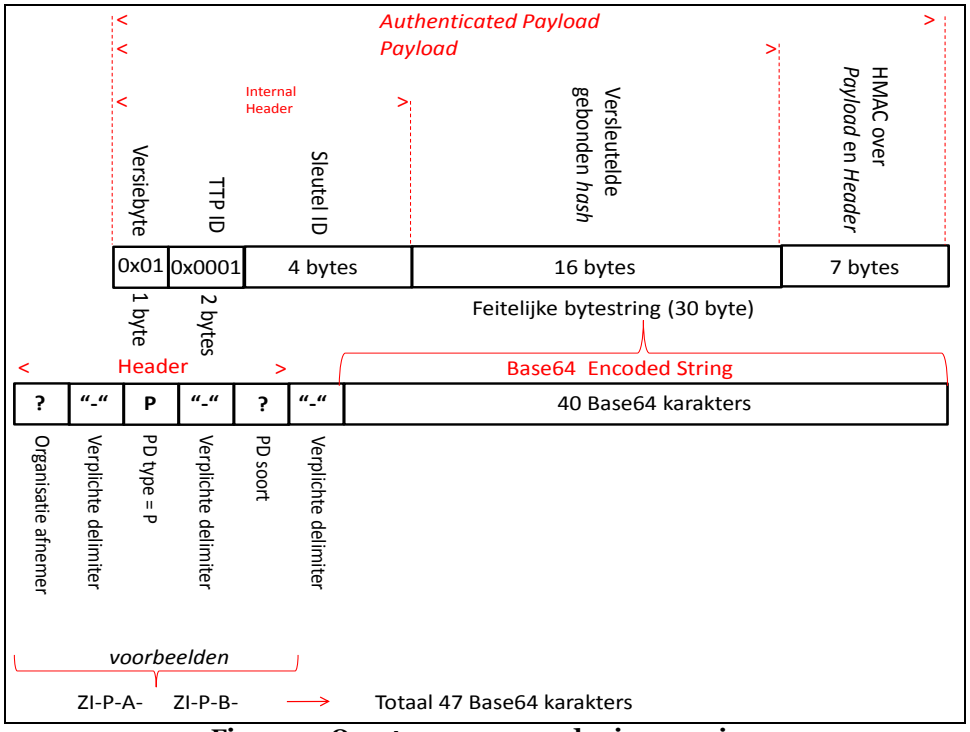
#	Input/bewerking	Toelichting
1.	S	Inputstring vanuit aanbieder bestand (BSN of Adres, zie Sectie 4.4). Er vindt een controle plaats of de input voldoet aan vorm vereisten, zie Sectie 4.4. Indien dit niet het geval is wordt de string ES+"1"+ minusk31 geretourneerd en verdere verwerking rond het specifieke pseudoniem wordt afgebroken. Hierbij is minusk31 de string bestaande uit 39 maal het teken "-". Merk op dat de string "1"+ minusk31 geen valide Base64 string is.
2.	SH=S2H(S)	String geconverteerd naar HEX
3.	ES	External Header volgens Tabel 1 en Figuur 2.
4.	EH=S2H(ES)	ES in Hex
5.	IH=Version_ID+TTP_ID	Internal Header volgens Figuur 2
6.	H1=SHA256(SH)	Hele SHA256 Hash van String S
7.	TH=Tr(32, H1)	Hash truncated op 32HEX(=16BYTE)
8.	PA=IH+TH	Payload volgens Figuur 2
9.	EH+PA	Voorbereiding voor *checksum*
10.	H2=SHA256(EH+PA)	Berekening voor *checksum*
11.	CH=Tr(12, H2)	*Checksum* Truncated op 12HEX(6byte)
12.	AP=IH+TH+CH	*Authenticated Payload* volgens Figuur 2
13.	BA=Base64E(AP)	AP Base64 *encoded*
14.	PP=ES+BA	Prematuur Pseudoniem volgens Figuur 2

Tabel 2: Constructie van prematuur pseudoniemen

4.3.2 PD soort 'P' versie 1

Vergelijk de generieke PD structuur aangegeven in Figuur 1. Zoals aangegeven in Sectie 4.2 is de pseudonimiseringsdienst vrij om de X parameter in te vullen in lijn met de specificaties in Tabel 1. Een PD van type "P", versie 1 heeft parameter Y="P" en heeft als eerste byte in de decoded Base64 string, i.e. de versie byte, 01. De parameter Y kan in deze specificatie slechts de waarden "A" (input adres-inputstring) en "B" (input BSN-string) hebben. Een PD van dit type heet *pseudoniem*.

De decoded Base64 string bestaat behalve uit de versie byte, uit een TTP_ID van twee bytes corresponderende met een unieke referentie van de TTP die het pseudoniem oorspronkelijk heeft gevormd alsmede een referentie naar een sleutel set die de TTP daarbij heeft gebruikt. De sleutel set verwijst naar een AES sleutel aangeven met *K* en een HMAC-SHA256 sleutel die wordt aangegeven met *L*. Deze drie velden vormen de internal header. Verder bestaat een versleutelde secure hash waarde van 16 bytes en een HMAC waarde waarmee de integriteit en authenticiteit kan worden geverifieerd. De nadere betekenis en constructie hiervan zal onderstaand worden uitgelegd.



Figuur 3: Opzet van een pseudoniem versie 1

#	Input/bewerking	Uitleg
1.	PP	Prematuur Pseudoniem aangeleverd. • Er wordt gecontroleerd of het PP afkomstig is van een exceptieproces bij de generatie van de prematuur pseudoniemen, i.e. van de vorm ES+"1"+ minusk31. Zie Tabel 2. In dit geval wordt de string ES+"1"+ minusk31 geretourneerd en verdere verwerking wordt afgebroken. Hierbij bestaat de string minusk39 uit 39 maal het karakter "-". • Bij overige validatie fouten wordt de string ES2+"2"+ minusk39 geretourneerd en verdere verwerking wordt afgebroken.
2.	E1=getH(PP)	Header in input PP, cf. Figuur 2.
3.	EH1=S2H(E1)	E1 in Hex
4.	T	PD Type (B of H)
5.	TH=S2H(T)	PD Type in HEX
6.	B=getB64(PP)	Base64 string, cf. Figuur 2.
7.	AP=Base64D(B)	Authenticated Payload (Base64 decoded)
8.	PA=AP[1-38]	Payload, zie Figuur 3.
9.	CHI=AP[39-48]	CHECKSUM opgegeven in PA
10.	EH1+PA	Voorbereiding voor *checksum*
11.	H=SHA256(EH1+PA)	Berekening voor *checksum*
12.	CH=Tr(10,H)	Controleer of Checksum gelijk is aan de opgegeven checksum.
13.	IHI=PA[1-6]	Internal Header in PA (input)
14.	SIO	SleutelID voor output.
15.	IHO=IHI+SIO	Internal Header voor output cf. Figuur 3.
16.	HI=PA[7-38]	Hash in PA (input)
17.	TH+HI	Binding met pseudoniem type
18.	HHI=SHA256(T+HI)	Hash van gebonden pseudoniem
19.	THHI=Tr(16,HHI)	Truncated op 32HEX(=16BYTE)
20.	E=AES(K,THHI)	Hash AES versleuteld met K
21.	P=IHO+E	Payload
22.	ES2	External Header voor output
23.	EH2=S2H(ES2)	ES2 in Hex
24.	HMACI=EH2+P	HMAC INPUT
25.	HM=HMAC(L, HMACI)	HMAC met sleutel L
26.	THM=Tr(14,HM)	Truncated op 14HEX(=7BYTE)
27.	AP=P+THM	Authenticated Payload cf. Figuur 3.
28.	BA=Base64E(AP)	AP Base64 encoded
29.	P=ES2+BA	BSN Pseudoniem

Tabel 3: Generatie van pseudoniemen

4.4 PD Inputsstrings

In deze sectie zullen wij de twee PD Inputsstrings definiëren:
Adresstring ('A') en BSNstring ('B').

4.4.1 Inputsoort 'A' (Adres-string)

De basis voor een Adresstring zijn drie alfanumerieke velden PC, WA en HT:

- Postcode woonadres PC, een string van 6 karakters waarvan de eerste 4 cijfers en de laatste twee letters zijn.
- Huisnummer woonadres WA, een string van maximaal 5 karakters bestaande uit (maximaal) 5 cijfers.
- Huisnummertoevoeging woonadres HT, een string van maximale 12 alfanumerieke karakters.

De Adresstring is de concatenatie van PC, WA en HT waarin alle letters in UPPER worden geplaatst (hoofdletters).

Voorbeeld

Veld/Bewerking	Voorbeeld
PC	"1234aa"
WA	"123"
HT	"boven"
PC+WA+HT	"1234aa123boven"
Adresstring=UPPER(PC+WA+HT)	"1234AA123BOVEN"

4.4.2 Inputsoort 'B' (BSN-string)

Een BSN wordt gepresenteerd als een numerieke string (BSNstring) van precies 9 negen cijfers, als het BSN uit minder dan 9 cijfers bestaat wordt het links aangevuld met nullen.
Voorbeelden:

- als b=564148738 dan wordt dit gepresenteerd als de string S_b="564148738".
- als b=64148737 dan wordt dit gepresenteerd als de string S_b="064148737".

Een BSNstring mag alleen worden toegepast in pseudonimisering operaties indien het aan de zogenaamde 11-proef voldoet.

Als we het BSN schrijven in decimale representatie B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 dan voldoet het aan de 11-proef indien $9*B_1 + 8*B_2 + 7*B_3 +6*B_4 + 5*B_5 +4*B_6 +3*B_7 +2*B_8 - B_9$ deelbaar door 11 is. Geldige voorbeelden van BSNs zijn: 64148737, 111222333, 123456782 en 999999990.

4.5 Minimale eisen aan sleutelbeheer

1. De cryptografische sleutels (AES en HMAC) dienen cryptografisch random gegenereerd te zijn en beheerd te worden in lijn met [NIST]. Verdere duiding van deze eisen zijn buiten de reikwijdte van deze specificatie.
2. Per combinatie van een afnemer en een pseudoniem type dient een unieke AES sleutel te worden gehanteerd. Eenzelfde AES sleutel mag dus ***niet*** worden gebruikt voor de generatie van BSN én Adres pseudoniemen, ook niet bij dezelfde afnemer. Het wordt aanbevolen ook unieke HMAC sleutels te hanteren voor elke combinatie en pseudoniem type. Minimaal wordt vereist dat HMAC sleutels niet worden gedeeld door de pseudonimiseringdienst over afnemers, dit met name om de portabiliteit niet te bemoeilijken. Met andere woorden: verschillende afnemers hebben verschillende HMAC sleutels. Opgemerkt wordt dat bovenstaande sleutel eigenschappen efficiënt kunnen worden ingericht middels sleutel diversificatie technieken, zie [KDF].

5 Vergelijking met de gestelde eisen

In onderstaande tabel hebben wij de eisen genomen uit Sectie 3 en deze vergelijken met de voorgestelde opzet.

#	Vereiste	Toelichting
1.	*Conformiteit met de CBP richtlijnen*	In lijn met de eisen voeren zowel de aanbieder als de pseudonimiseringsdienst een 'encryptie' uit in de vorm van een secure hash (SHA256). De pseudonimiseringsdienst voert daarbij nog een AES encryptie uit.
2.	*Pseudoniemen niet omkeerbaar*	Deze eis wordt ingevuld door de AES versleuteling die de pseudonimiseringsdienst uitvoert.
3.	*Compartimentering van afnemer domeinen*	Deze eis wordt ingevuld door de tweede eis in Sectie 4.5 dat de pseudonimiseringsdienst per combinatie van een afnemer en een pseudoniem type een unieke AES sleutel moet worden gehanteerd.
4.	*Compartimentering van domein typen*	Ook deze eis wordt ingevuld door de tweede eis in Sectie 4.5 dat de pseudonimiseringsdienst per combinatie van een afnemer en een pseudoniem type een unieke AES sleutel moet worden gehanteerd. Aanvullend hieraan biedt de binding van het pseudoniem type aan het pseudoniem (Stappen 17-19, Tabel 3) hier extra zekerheid bij. Zelf in het geval dat de pseudonimiseringsdienst – in strijd met de eisen in Sectie 4.5 – dezelfde AES sleutel zou gebruiken voor verschillende pseudoniem type dan nog zou aan de eis voldaan zijn.
5.	*Authenticiteit van pseudoniemen beschermd*	De toegevoegde 7 byte HMAC in de pseudoniemen voorziet hierin. De kans dat een willekeurig samengesteld pseudoniem door de HMAC test komt is 2^{-56} hetgeen als voldoende wordt gezien in deze context.
6.	*Cryptografische stand der techniek*	De methode maakt gebruik van SHA256, HMAC gebaseerd op SHA256 en AES. Dit zijn gangbare cryptografische technieken, die deel uit maken van de richtlijnen in [NIST].
7.	*Migreerbaarheid naar andere cryptografische sleutels*	Hiertoe dient de Sleutel ID in de interne header. Vergelijk Figuur 3.
8.	*Migreerbaarheid naar andere algoritmen*	Hiertoe dient de Versie byte in de interne header. Vergelijk Figuur 3.
9.	*Overdraagbaarheid ('portability')*	De specificatie ondersteunt dit in opzet omdat de specificatie publiek is. In praktische zin kan dit worden ondersteund door sleutelmateriaal op veilige wijze van de ene naar de andere

		pseudonimiseringsdienst te verplaatsen. De eisen in Sectie 4.5 stellen daarbij dat een pseudonimiseringsdienst zijn sleutelmateriaal niet mag delen over verschillende afnemers, hetgeen een noodzakelijke voorwaarde is voor een dergelijke verplaatsing.
10.	*Compactheid* Pseudoniemen moeten zo min mogelijk data (‘karakters) benutten.	Hiertoe is gebruik gemaakt van Base64 codering en is zo zuinig mogelijk gecodeerd.
11.	*Representatie in printbare vorm*	Hiertoe is gebruik gemaakt van Base64 codering.
12.	*(Eenvoudig) interpreteerbaar*	Hiertoe dient de Header van de pseudoniemen.

6 Referenties

#	Document
AES	National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), FIPS 140-2, November 26, 2001. Zie http://csrc.nist.gov.
BASE	The Internet Engineering Task Force, The Base16, Base32, and Base64 Data Encodings, RFC 4648, october 2006. Zie www.ietf.org.
HMAC	National Institute of Standards and Technology (NIST), The Keyed-Hash Message Authentication Code (HMAC), FIPS PUB 198-1, juli 2008. Zie http://csrc.nist.gov.
HSM	National Institute of Standards and Technology (NIST), Security Requirements for Cryptographic Modules, FIPS 140-2, May 25, 2001. See http://csrc.nist.gov.
KDF	ISO, ISO/IEC 18033-2:2006 Information technology - Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers, 2006.
NIST	National Institute of Standards and Technology (NIST), Recommendation for Key Management – Part 1: General (Revision 3), 800-57, juli 2012. Zie http://csrc.nist.gov.
SHA256	National Institute of Standards and Technology (NIST), Secure Hash Standard (SHS), FIPS 180-4, maart 2012. Zie http://csrc.nist.gov.

A. Voorbeelden

In deze bijlage berekenen wij voorbeelden van (prematuur) pseudoniemen van type Adres en BSN. In alle gevallen is het pseudonimiseer schema gelijk aan 1 (dus 01 in hex) waarvan de opzet is beschreven in Sectie 4. Verder is de TTP identifier gelijk aan 1 (dus 0001 in hex). Dit voorziet aldus in het gebruik van SHA256, AES en HMAC-SHA256.

Specifiek:

- de afnemer organisatie wordt door de TTP aangegeven met “ZI”, i.e. Afnemer_ID=“ZI”,
- de TTP_ID=01,
- worden twee sets gebruikt aangegeven met 00000001(in hex) en 00000002 (in hex):
  Set 1:
  - een AES sleutel K1=000102030405060708090A0B0C0D0E0F (16 byte in hex), en
  - een HMAC-SHA256 sleutel L1=000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F (32 byte in hex)

  Set 2:
  - een AES sleutel K2=F0E0D0C0B0A090807060504030201000 (16 byte in hex), en
  - een HMAC-SHA256 L2=sleutel 0F0E0D0C0B0A090807060504030201000F0E0D0C0B0A090807060504030201000 (32 byte in hex)

Om deze voorbeelden eenvoudig te kunnen narekenen maken wij gebruik van de volgende on-line tools:

- http://extranet.cryptomathic.com/ voor de conversie van ASCII naar HEX en voor de berekening van SHA256, HMAC-SHA256 en AES,
- http://tomeko.net/online_tools/hex_to_base64.php voor de conversie van HEX naar Base64 en omgekeerd,

A.1. Voorbeelden prematuur pseudoniemen (PD type="H")

In deze sectie illustreren we de generatie van prematuur pseudoniemen (PD type="H") versie 1, zoals beschreven in Sectie 4.3.1. Daarbij zullen we zowel een Adres prematuur pseudoniem (PD inputsoort="A") als een BSN prematuur pseudoniem (PD inputsoort="B") genereren. Verder gaan we uit van de volgende zaken (vergelijk alle parameters in Figuur 2):

- de afnemer organisatie wordt door de TTP aangegeven met "ZI", i.e. Afnemer_ID="ZI".
- de versie van het prematuur pseudoniem is 1, i.e. Version_ID=01 in hexadecimale representatie. Dit betekent dat de het pseudoniem is gebaseerd op de SHA256 secure hash functie, cf. Sectie INSERT.
- de TTP heeft TTP ID "1", i.e. TTP _ID=0001 in hexadecimale representatie.
- bij het BSN prematuur pseudoniem gaan we uit van BSN-string "064148737" en van BSN-string "564148738".
- bij het Adres prematuur pseudoniem gaan we uit van de Adres-string "1234AA123BOVEN".

BSN prematuur pseudoniem generatie

Parameter/bewerking	Voorbeeld Waarde	Toelichting
S	"064148737"	BSN-string vanuit aanbieder bestand (en gecontroleerd)
SH=S2H(SS)	303634313438373337	BSN-string geconverteerd naar HEX
ES	"ZI-H-B-"	External Header volgens Tabel 1
EH=S2H(ES)	5A492D482D422D	E in Hex
IH=Version_ID+TTP_ID	010001	Internal Header volgens Figuur 2
H1=SHA256(SH)	01CFA0E9347BB4C3E365DAE070C85D454FF0CC43809209E0CB21B1594770D445	Hele SHA256 Hash van BSN-string
Telraam	12345678901234567890123456789012345678901234567890123456789012345678901234 1 2 3 4 5 6	
TH=Tr(32,H1)	01CFA0E9347BB4C3E365DAE070C85D45	Hash truncated op 32HEX(=16BYTE)
PA=IH+TH	01000101CFA0E9347BB4C3E365DAE070C85D45	Payload volgens Figuur 2
EH+PA	5A492D482D422D01000101CFA0E9347BB4C3E365DAE070C85D45	Voorbereiding voor *checksum*
H2=SHA256(EH+PA)	7756F64A102195441EECA7363670892F6F423E05B5B4425B74CA67C3E2031744	Berekening voor *checksum*
CH=Tr(10, H2)	7756F64A10	*Checksum* Truncated op 10 HEX(5byte)
AP=IH+TH+CH	01000101CFA0E9347BB4C3E365DAE070C85D457756F64A10	*Authenticated Payload* volgens Figuur 2
BA=Base64E(AP)	"AQABAc+g6TR7tMPjZdrgcMhdRXdW9koQ"	AP Base64 *encoded*
PP=ES+BA	"ZI-H-B-AQABAc+g6TR7tMPjZdrgcMhdRXdW9koQ"	Prematuur Pseudoniem volgens Figuur 2 (39 Base64 karakters).

Adres prematuur pseudoniem generatie

Parameter/bewerking	Waarde	Toelichting
S	"1234AA123BOVEN"	Adres-string vanuit aanbieder bestand (en gecontroleerd)
SH	3132333441413132334 24F56454E	Adres-string geconverteerd naar HEX
ES	"ZI-H-A-"	External Header volgens Tabel 1
EH=S2H(ES)	5A492D482D412D	E in Hex
IH=Version_ID+TTP_ID	010001	Internal Header volgens Figuur 2
H1=SHA256(SH)	BCE52220DC12FC36B4CE4E48870242533A96AF1D749B88DB8DAB6D1375830117	Hele SHA256 Hash van Adres-string
Telraam	12345678901234567890123456789012345678901234567890123456789012345678901234 1 2 3 4 5 6	
TH=Tr(32,H1)	BCE52220DC12FC36B4CE4E4887024253	Hash truncated op 32HEX(=16BYTE)
PA=IH+TH	010001BCE52220DC12FC36B4CE4E4887024253	Payload volgens Figuur 2
EH+PA	5A492D482D412D010001BCE52220DC12FC36B4CE4E4887024253	Voorbereiding voor *checksum*
H2=SHA256(EH+PA)	4B0E1BE5F3195480E2372D732B0CA65195912C57A3CA5104871C15C19E4247C4	Berekening voor *checksum*
CH=Tr(10,H2)	4B0E1BE5F3	Checksum=Truncated op 10HEX(5byte)
AP=IH+TH+CH	010001BCE52220DC12FC36B4CE4E48870242534B0E1BE5F3	*Authenticated Payload* volgens Figuur 2
BA=Base64E(AP)	"AQABvOUiINwS/Da0zk5IhwJCU0sOG+Xz"	AP Base64 encoded
PP=ES+BA	"ZI-H-A-AQABvOUiINwS/Da0zk5IhwJCU0sOG+Xz"	Prematuur Pseudoniem volgens Figuur 2 (39 Base64 karakters).

A.2. Voorbeelden pseudoniemen (PD type="P")

BSN pseudoniem generatie

Parameter/bewerking	Waarde	Toelichting
PP	ZI-H-B-AQABAc+g6TR7tMPjZdrgcMhdRXdW9koQ	Prematuur BSN Pseudoniem aangeleverd (en gecontroleerd)
E1= getH(PP)	"ZI-H-B-"	External Header in input PP, cf. Figuur 2.
EH1=S2H(E1)	5A492D482D422D	E1 in Hex
T	"B"	PD Type
TH=S2H(T)	42	PD Type in HEX
B=getB64(PP)	"AQABAc+g6TR7tMPjZdrgcMhdRXdW9koQ"	Base64 string, cf. Figuur 2.
AP=Base64D(B)	01000101CFA0E9347BB4C3E365DAE070C85D457756F64A10	Auth. Payload (Base64 decoded)
PA=AP[1-38]	01000101CFA0E9347BB4C3E365DAE070C85D45	Payload, zie Figuur 3.
CHI=AP[39-48]	7756F64A10	CHECKSUM opgegeven in PA
EH1+PA	5A492D482D422D01000101CFA0E9347BB4C3E365DAE070C85D45	Voorbereiding voor *checksum*
H=SHA256(EH1+PA)	7756F64A102195441EECA7363670892F6F423E05B5B4425B74CA67C3E2031744	Berekening voor *checksum*
Telraam	12345678901234567890123456789012345678901234567890123456789012345678901234 1 2 3 4 5 6	
CH=Tr(10,H)	7756F64A10	Berekende Checksum=CHI!
IHI=PA[1-6]	010001	Internal Header in PA (input)
SIO	00000001	SleutelID voor output (=set 1)
IHO=IHI+SIO	01000100000001	Internal Header voor output cf. Figuur 3.
HI=PA[7-38]	01CFA0E9347BB4C3E365DAE070C85D45	Hash in PA (input)
TH+HI	4201CFA0E9347BB4C3E365DAE070C85D45	Binding met pseudoniem type
HHI=SHA256(T+HI)	9F7D67A8159EA295478CBF4B931B84BF81E3F91E190C33E5A8FAB65517B047FD	Hash van gebonden pseudoniem type
Telraam	12345678901234567890123456789012345678901234567890123456789012345678901234 1 2 3 4 5 6	
THHI=Tr(16,HHI)	9F7D67A8159EA295478CBF4B931B84BF	Truncated op 32HEX(=16BYTE)
E=AES(K1,THHI)	8F697DCBC6F07FF94469CF6B539D9B96	Hash AES versleuteld met K1
P=IHO+E	010001000000018F697DCBC6F07FF94469CF6B539D9B96	Payload
ES2	"ZI-P-B-"	External Header voor output
EH2=S2H(ES2)	5A492D502D422D	ES2 in Hex
HMACI=EH2+P	5A492D502D422D010001000000018F697DCBC6F07FF94469CF6B539D9B96	HMAC INPUT
HM=HMAC(L1, HMACI)	8CD4C7F973457BD4EE52EB59072BE4A009D023EE65511F78A1C471B4423FFB8F	HMAC met sleutel L1
THM=Tr(14,HM)	8CD4C7F973457B	Truncated op 14HEX(=7BYTE)
AP=P+THM	010001000000018F697DCBC6F07FF94469CF6B539D9B968CD4C7F973457B	Auth. Payload cf. Figuur 3.
BA=Base64E(AP)	"AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7"	AP Base64 encoded
P=ES2+BA	"ZI-P-B-AQABAAAAAY9pfcvG8H/5RGnPa1Odm5aM1Mf5c0V7"	BSN Pseudoniem (47 Base64 karakters).

Adres pseudoniem generatie

Parameter/bewerking	Waarde	Toelichting
PP	ZI-H-A-AQABvOUiINwS/Da0zk5IhwJCU0sOG+Xz	Prematuur Adres Pseudoniem (en gecontroleerd)
E1=getH(PP)	"ZI-H-A-"	External Header in input, cf. Figuur 2.
EH1=S2H(E1)	5A492D482D412D	E1 in Hex
T	"A"	PD Type
TH=S2H(T)	41	PD Type in HEX
B=getB64(PP)	"AQABvOUiINwS/Da0zk5IhwJCU0sOG+Xz"	Base64 string, cf. Figuur 2.
AP=Base64D(B)	010001BCE52220DC12FC36B4CE4E48870242534B0E1BE5F3	Auth. Payload (Base64 decoded)
PA=AP[1-38]	010001BCE52220DC12FC36B4CE4E4887024253	Payload
CHI=AP[39-48]	4B0E1BE5F3	CHECKSUM INSIDE
EH1+PA	5A492D482D412D010001BCE52220DC12FC36B4CE4E4887024253	
H=SHA256(EH1+PA)	4B0E1BE5F3195480E2372D732B0CA65195912C57A3CA5104871C15C19E4247C4	Hele Hash
Telraam	12345678901234567890123456789012345678901234567890123456789012345678901234	
	1 2 3 4 5 6	
CH=Tr(10,H)	4B0E1BE5F3	Berekende Checksum=CHI!
IHI=PA[1-6]	010001	Internal Header in Input
SIO	00000002	SleutelID voor output (=set 2)
IHO=IHI+SIO	01000100000002	Internal Header voor output
HI=PA[7-38]	BCE52220DC12FC36B4CE4E4887024253	Hash in PA (input)
TH+HI	41BCE52220DC12FC36B4CE4E4887024253	Binding met pseudoniem type
HHI=SHA256(T+HI)	7EDE18FB972FC7268AFC4EFC4F52F2A889A342FC894197F59EA81EB1F4D29FAB	Hash van gebonden pseudoniem type
Telraam	12345678901234567890123456789012345678901234567890123456789012345678901234	
	1 2 3 4 5 6	
THHI=Tr(16,HHI)	7EDE18FB972FC7268AFC4EFC4F52F2A8	Truncated op 32HEX(=16BYTE)
E=AES(K2,HI)	727013A3C0B4C1F5A5DEEFD166224FD8	HI AES versleuteld met K2
P=IHO+E	01000100000002727013A3C0B4C1F5A5DEEFD166224FD8	Payload
ES2	"ZI-P-A-"	External Header voor output
EH2=S2H(ES2)	5A492D502D412D	ES2 in Hex
HMACI=EH2+P	5A492D502D412D01000100000002727013A3C0B4C1F5A5DEEFD166224FD8	HMAC INPUT
HM=HMAC(L2, HMACI)	40AB887E7DAA1B9BA3F176E1606C3C59F2DBC532DFEE42B36785AE77CFE7907C	HMAC met sleutel L2
THM=Tr(14,HM)	40AB887E7DAA1B	Truncated op 14HEX(=7BYTE)
AP=P+THM	01000100000002727013A3C0B4C1F5A5DEEFD166224FD840AB887E7DAA1B	Auth. Payload cf. Figuur 3.
BA=Base64E(AP)	"AQABAAAAAnJwE6PAtMH1pd7v0WYiT9hAq4h+faob"	AP Base64 encoded
P=ES2+BA	"ZI-P-A-AQABAAAAAnJwE6PAtMH1pd7v0WYiT9hAq4h+faob"	Adres Pseudoniem (47 Base64 karakters).

B. Nadere informatie

De technische beveiliging van de pseudonimiseringsdienst kan worden verhoogd door de inzet van een zogenaamde Hardware Security Module (HSM), zie [HSM]. In een HSM kunnen cryptografische sleutels beschermd worden opgeslagen: ze kunnen daarbinnen wel worden gebruikt maar kunnen er niet uit worden geëxporteerd.

Idealiter zouden de AES en HMAC sleutels in een niet exporteerbare vorm gegenereerd en opgeslagen worden in een HSM en daarbinnen ook worden gebruikt voor de pseudonimiseringsdienst.

Een alternatieve, en technisch minder veilige oplossing dan het geschetste gebruik van een HSM bestaat eruit de sleutels AES en HMAC sleutels te laten genereren door de HSM middels een sleutel diversificatie mechanisme [KDF]. In deze opzet zouden de resulterende sleutels dan worden geëxporteerd naar een beschermde server omgeving waarbinnen ze kunnen worden gebruikt voor een pseudonimisering opdracht en daarna veilig worden gewist.

Dit sleutel diversificatie mechanisme zou onder een “vier ogen” principe kunnen worden beschermd. Dat wil zeggen, slechts als twee geautoriseerde beheerders een wachtwoord of PIN ingeven laat de HSM de vereiste sleutel diversificatie en export toe naar de applicatie toe. In feit volstaat een smartcard oplossing voor deze opzet.

Een algemeen aandachtpunt bij het gebruik van een HSM zijn mitigerende maatregelen voor het geval deze uitvalt en waardoor de continuïteit van de pseudonimiseringsdienst in gevaar zou kunnen komen. HSMs beschikken vaak over de mogelijkheid om een backup te maken die onder controle staat van meerdere personen.