

# Bescherming van persoonsgegevens met geavanceerde cryptografie

Kristof Verslype  
PhD, Smals Research



## κρυπτός γράφειν

(kryptós gráfein = verborgen schrijven)

Het beschermen van gegevens  
m.b.v. wiskundige principes

Oorspronkelijk bedoeld om data  
geheim te houden  
(tegenwoordig veel ruimer)

Gebruikt door Oude Egyptenaren,  
Grieken & Romeinen

# Caesarcijfer (aka ROT-3)



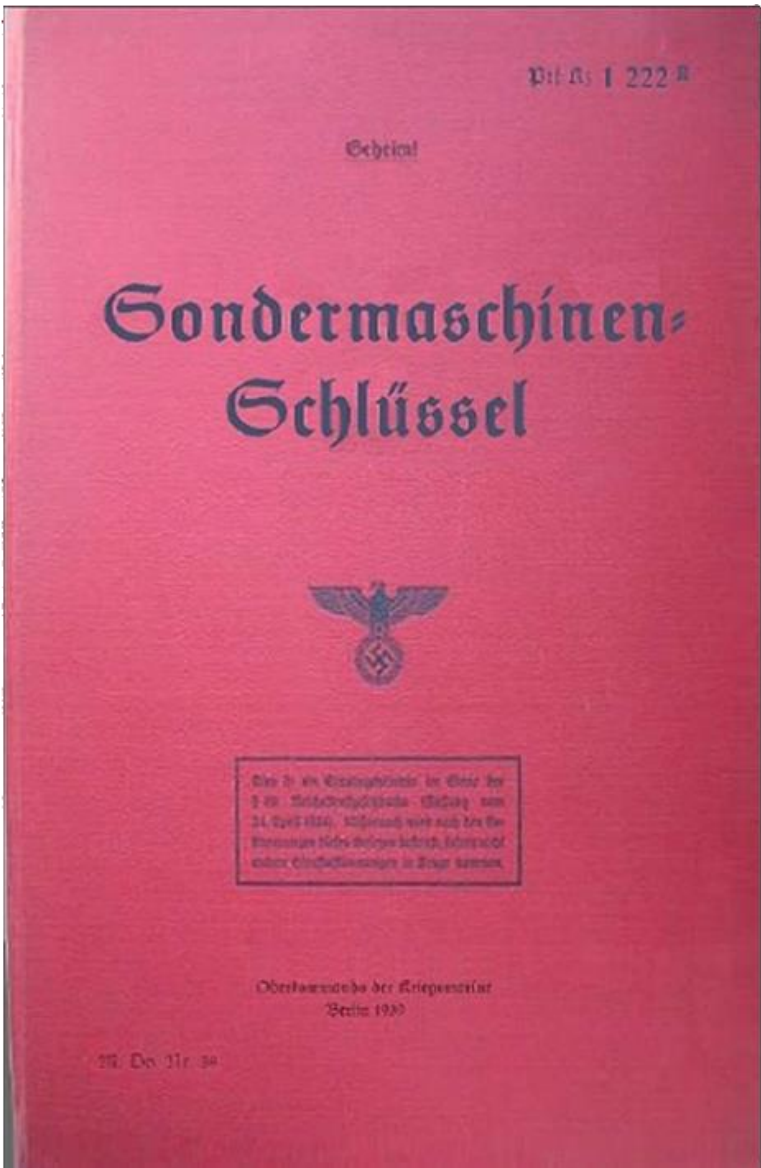
Klare tekst: **D I T I S Z E E R G E H E I M**

Cijfertekst: **A F Q F P W B B O D B E B F J**

Door Caesar en Augustus om met de veldheren te communiceren

‘Voldoende’ gezien ongeletterdheid vijand





SPECIAL OPERATIONS SET NO. 200 2

AUGUST 1939

Tag	Wochenlage	(Sonntagsruhe)	Stundenschiebungen	Zeitsgruppen
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30
31	31	31	31	31
32	32	32	32	32
33	33	33	33	33
34	34	34	34	34
35	35	35	35	35
36	36	36	36	36
37	37	37	37	37
38	38	38	38	38
39	39	39	39	39
40	40	40	40	40
41	41	41	41	41
42	42	42	42	42
43	43	43	43	43
44	44	44	44	44
45	45	45	45	45
46	46	46	46	46
47	47	47	47	47
48	48	48	48	48
49	49	49	49	49
50	50	50	50	50
51	51	51	51	51
52	52	52	52	52
53	53	53	53	53
54	54	54	54	54
55	55	55	55	55
56	56	56	56	56
57	57	57	57	57
58	58	58	58	58
59	59	59	59	59
60	60	60	60	60
61	61	61	61	61
62	62	62	62	62
63	63	63	63	63
64	64	64	64	64
65	65	65	65	65
66	66	66	66	66
67	67	67	67	67
68	68	68	68	68
69	69	69	69	69
70	70	70	70	70
71	71	71	71	71
72	72	72	72	72
73	73	73	73	73
74	74	74	74	74
75	75	75	75	75
76	76	76	76	76
77	77	77	77	77
78	78	78	78	78
79	79	79	79	79
80	80	80	80	80
81	81	81	81	81
82	82	82	82	82
83	83	83	83	83
84	84	84	84	84
85	85	85	85	85
86	86	86	86	86
87	87	87	87	87
88	88	88	88	88
89	89	89	89	89
90	90	90	90	90
91	91	91	91	91
92	92	92	92	92
93	93	93	93	93
94	94	94	94	94
95	95	95	95	95
96	96	96	96	96
97	97	97	97	97
98	98	98	98	98
99	99	99	99	99
100	100	100	100	100

SPECIAL OPERATIONS SET NO. 200 2

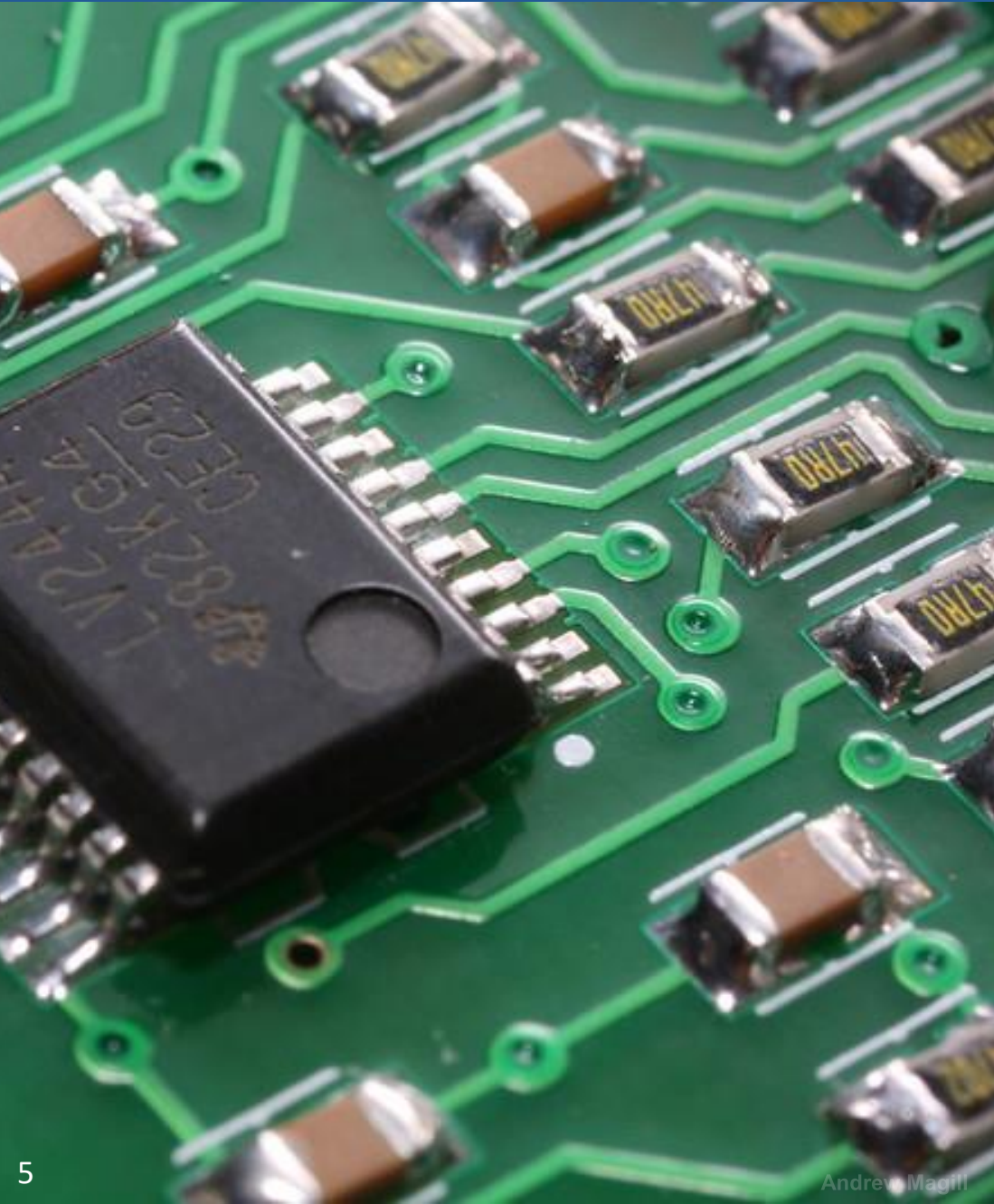
SEPTEMBER 1939

Tag	Wochenlage	(Sonntagsruhe)	Stundenschiebungen	Zeitsgruppen
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30
31	31	31	31	31
32	32	32	32	32
33	33	33	33	33
34	34	34	34	34
35	35	35	35	35
36	36	36	36	36
37	37	37	37	37
38	38	38	38	38
39	39	39	39	39
40	40	40	40	40
41	41	41	41	41
42	42	42	42	42
43	43	43	43	43
44	44	44	44	44
45	45	45	45	45
46	46	46	46	46
47	47	47	47	47
48	48	48	48	48
49	49	49	49	49
50	50	50	50	50
51	51	51	51	51
52	52	52	52	52
53	53	53	53	53
54	54	54	54	54
55	55	55	55	55
56	56	56	56	56
57	57	57	57	57
58	58	58	58	58
59	59	59	59	59
60	60	60	60	60
61	61	61	61	61
62	62	62	62	62
63	63	63	63	63
64	64	64	64	64
65	65	65	65	65
66	66	66	66	66
67	67	67	67	67
68	68	68	68	68
69	69	69	69	69
70	70	70	70	70
71	71	71	71	71
72	72	72	72	72
73	73	73	73	73
74	74	74	74	74
75	75	75	75	75
76	76	76	76	76
77	77	77	77	77
78	78	78	78	78
79	79	79	79	79
80	80	80	80	80
81	81	81	81	81
82	82	82	82	82
83	83	83	83	83
84	84	84	84	84
85	85	85	85	85
86	86	86	86	86
87	87	87	87	87
88	88	88	88	88
89	89	89	89	89
90	90	90	90	90
91	91	91	91	91
92	92	92	92	92
93	93	93	93	93
94	94	94	94	94
95	95	95	95	95
96	96	96	96	96
97	97	97	97	97
98	98	98	98	98
99	99	99	99	99
100	100	100	100	100

SPECIAL OPERATIONS SET NO. 200 2

OCTOBER 1939

Tag	Wochenlage	(Sonntagsruhe)	Stundenschiebungen	Zeitsgruppen
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30
31	31	31	31	31
32	32	32	32	32
33	33	33	33	33
34	34	34	34	34
35	35	35	35	35
36	36	36	36	36
37	37	37	37	37
38	38	38	38	38
39	39	39	39	39
40	40	40	40	40
41	41	41	41	41
42	42	42	42	42
43	43	43	43	43
44	44	44	44	44
45	45	45	45	45
46	46	46	46	46
47	47	47	47	47
48	48	48	48	48
49	49	49	49	49
50	50	50	50	50
51	51	51	51	51
52	52	52	52	52
53	53	53	53	53
54	54	54	54	54
55	55	55	55	55
56	56	56	56	56
57	57	57	57	57
58	58	58	58	58
59	59	59	59	59
60	60	60	60	60
61	61	61	61	61
62	62	62	62	62
63	63	63	63	63
64	64	64	64	64
65	65	65	65	65
66	66	66	66	66
67	67	67	67	67
68	68	68	68	68
69	69	69	69	69
70	70	70	70	70
71	71	71	71	71
72	72	72	72	72
73	73	73	73	73
74	74	74	74	74
75	75	75	75	75
76	76	76	76	76
77	77	77	77	77
78	78	78	78	78
79	79	79	79	79
80	80	80	80	80
81	81	81	81	81
82	82	82	82	82
83	83	83	83	83
84	84	84	84	84
85	85	85	85	85
86	86	86	86	86
87	87	87	87	87
88	88	88	88	88
89	89	89	89	89
90	90	90	90	90
91	91	91	91	91
92	92			



- ▶ Sinds intrede computer (Jaren 1970)
- ▶ Bewijsbaar veilig gebaseerd op wiskundige assumpties (rigoreuze wetenschap)
- ▶ Meer dan geheim houden van communicatie

## CRYPTO WERKPAARDEN

### Encryptie

DES, AES, ElGamal, RSA, ...

### Digitale handtekeningen

RSA, DSA, Schnorr, ...

### Authenticatie

SSH, CHAP, ...

### Hashing

MD5, RipeMD, SHA-1, SHA-2,  
SHA-3

### Key exchange

Diffie–Hellman, ...

### Message authentication code

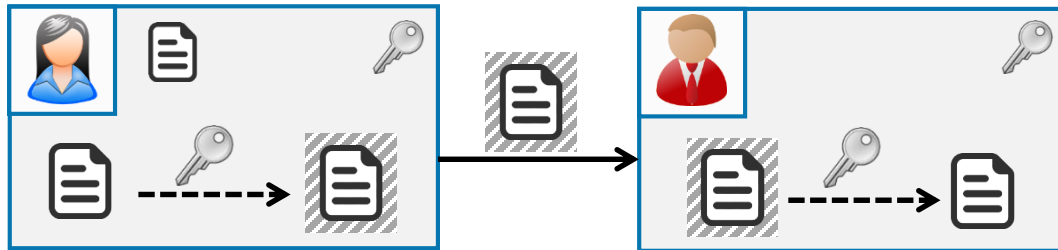
**code**  
HMAC, ...

## Doel

Confidentialiteit = vertrouwelijkheid

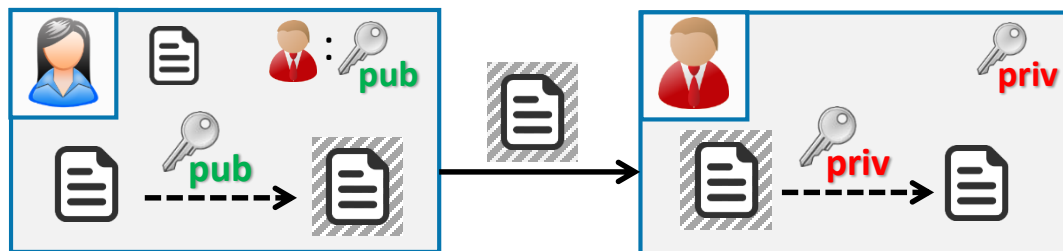
### Symmetrische encryptie

- ▶ Sneller
- ▶ vb. AES



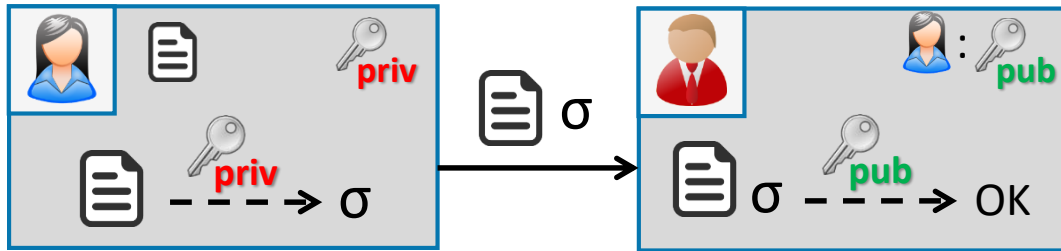
### Publieke sleutel encryptie

- ▶ Trager
- ▶ Vb. RSA, ElGamal

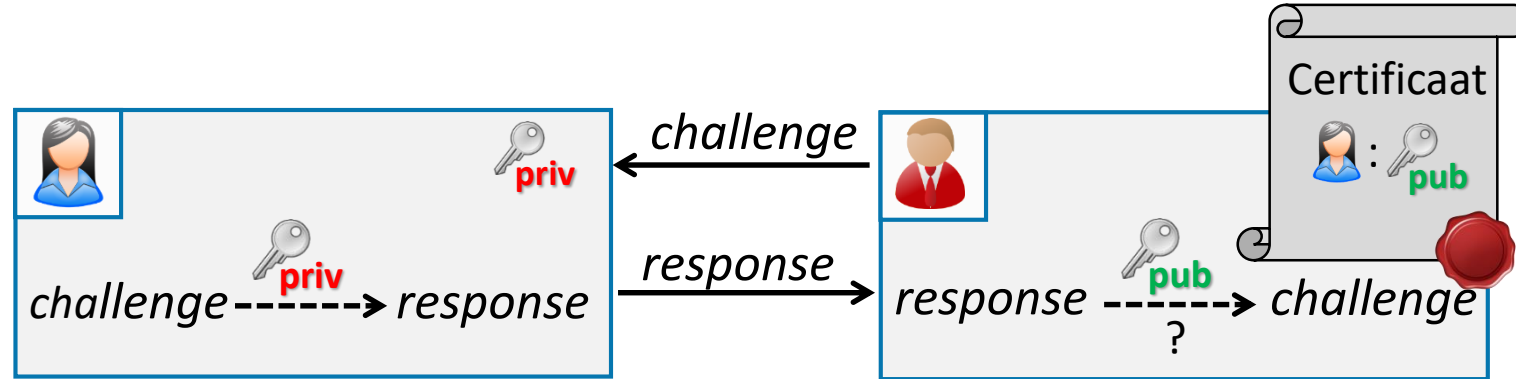


## Doel

Onweerlegbaarheid, integriteit, data authenticiteit



# Challenge-response authenticatie



## Trapdoor function

Met publieke sleutel kun je *challenge* uit *response* berekenen, maar niet omgekeerd  
Dat kan enkel met de private sleutel



CA

(Certificate authority)

## Toegepast door o.a.



# Cryptografische sleutels

## RSA-2048 public key

```

30 82 01 0a 02 82 01 01 00 c0 a7 e5 3a cf ea 93 df e0 ef fc fd 34 64 42 e1 65 2c ed 60 70 86 ed
47 18 64 53 d1 b1 84 b2 4a 98 96 89 de 54 c4 cd ef ad 59 a2 04 b6 53 14 6e dc 6f 90 0f 17 26 4f
3a 4e ad 7f 51 4d 00 fe b4 57 86 43 1c 53 14 08 24 af b6 da ae 22 83 31 9a 0b 0b 79 b1 51 34 9c
0e 51 d4 d4 6c e5 71 f9 9f bb e5 12 a5 15 68 1c 1b e6 dd 39 79 cb 6f 40 a3 fa 32 21 bc ee fc 30
31 d0 b7 22 97 59 52 9e 40 9c b6 f5 1e 9c f6 9e f6 8e 88 fc 10 37 ee 84 bc ad e3 bb 87 54 a9 88
e4 70 27 da 4f 02 e8 95 5f b6 ae 30 56 b3 b8 5c 18 8a 8e 1b 47 d7 94 f9 1d 1c 7c b4 23 16 10 2c
80 ca a6 7f 9c e6 2d 84 b7 0e 9d 9b 8d c6 b1 1d a9 b4 05 e0 73 25 8a 99 c2 9e c5 81 3c 96 33 7a
75 83 fe 5e 8c 0f 02 30 29 eb 28 fd 8c 9b 0d 48 34 df eb 3c b9 7c fe 9e dc bc 44 2a 9e 3c 27 82
    
```

Symmetric	DH or RSA	ECC
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

### Near-term protection

*Security for at least ten years (2019-2028)*

### Long-term protection

*Security for thirty to fifty years (2019-2068)*

## Bronnen

NIST, <https://csrc.nist.gov/Projects/Key-Management/publications>  
 Algorithms, Key Size and Protocols Report (2018), H2020-ICT-2014 – Project 645421, D5.4, ECRYPT-CSA, 02/2018.

Ook interessant: <https://www.keylength.com/>

Veel meer is mogelijk -

- ▶ Sinds intrede computer (Jaren 1970)
- ▶ Bewijsbaar veilig gebaseerd op wiskundige assumpties (rigoreuze wetenschap)
- ▶ Meer dan geheim houden van communicatie

## CRYPTO WERKPAARDEN

### Encryptie

AES, ElGamal, RSA, ...

### Digitale handtekeningen

RSA, DSA, Schnorr, ...

### Authenticatie

SSH, CHAP, ...

### Hashing

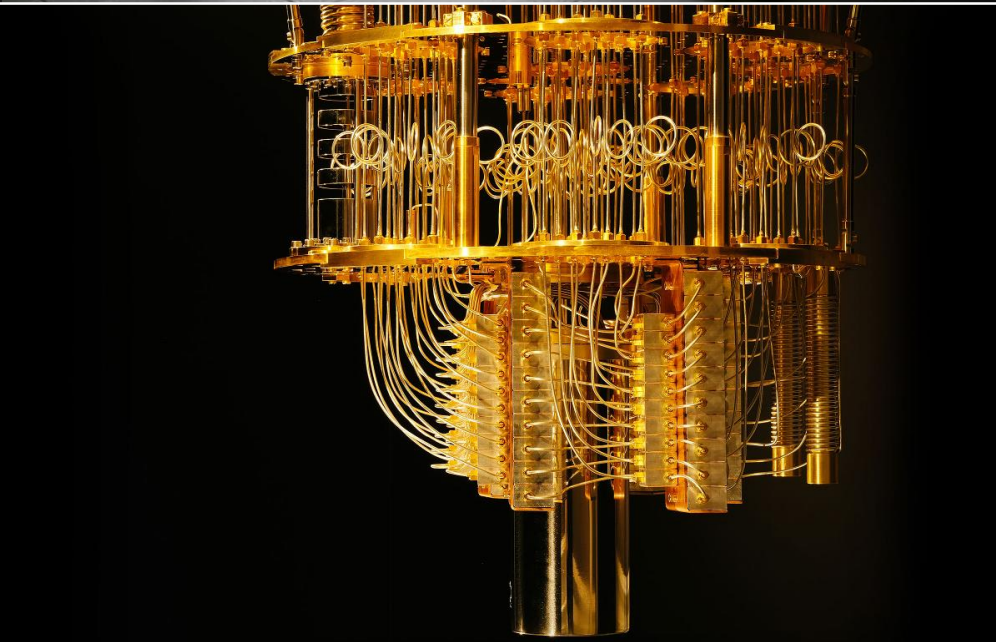
MD5, MD, SHA-1, SHA-2,

SHA-3

### Key exchange

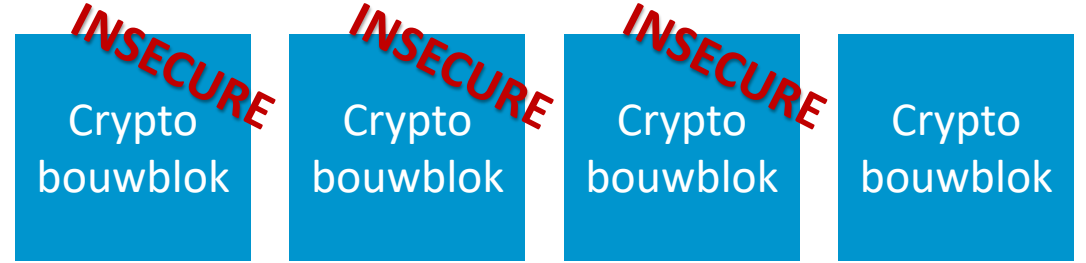
Diffie-Hellman, ...

Use cases?

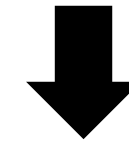


## TRADITIONELE CRYPTOGRAFIE

Mogelijkheden:

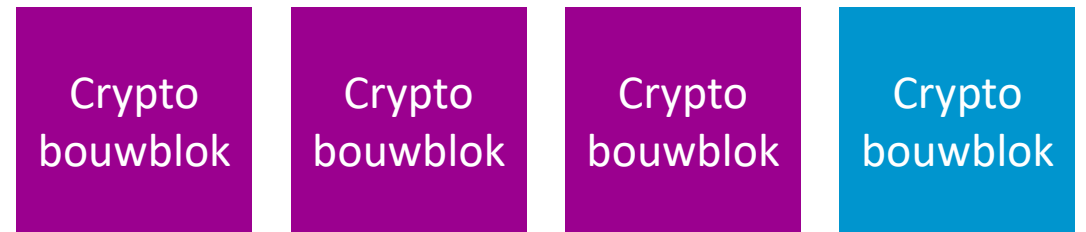


Fundamenten:



## KWANTUMRESISTENTE CRYPTOGRAFIE

Mogelijkheden:



Fundamenten:



# Geavanceerde cryptografie

## Speciale cijfertekst

- 1 Threshold encryption
- 2 Format-preserving encryption
- 3 Proxy reencryption

## Authenticatie

- 4 Secure remote password protocol
- 5 Attribute-based credentials

---

## Privacyvriendelijke opvraging

- 6 Secure multiparty computation
- 7 Oblivious transfer
- 8 Private set intersection
- 9 Oblivious join

Er is veel meer!



**WORK IN PROGRESS**

**GEEN WISKUNDE**

**VUUR VRAGEN!**



# Threshold encryption

Theorie:



Eigen code:



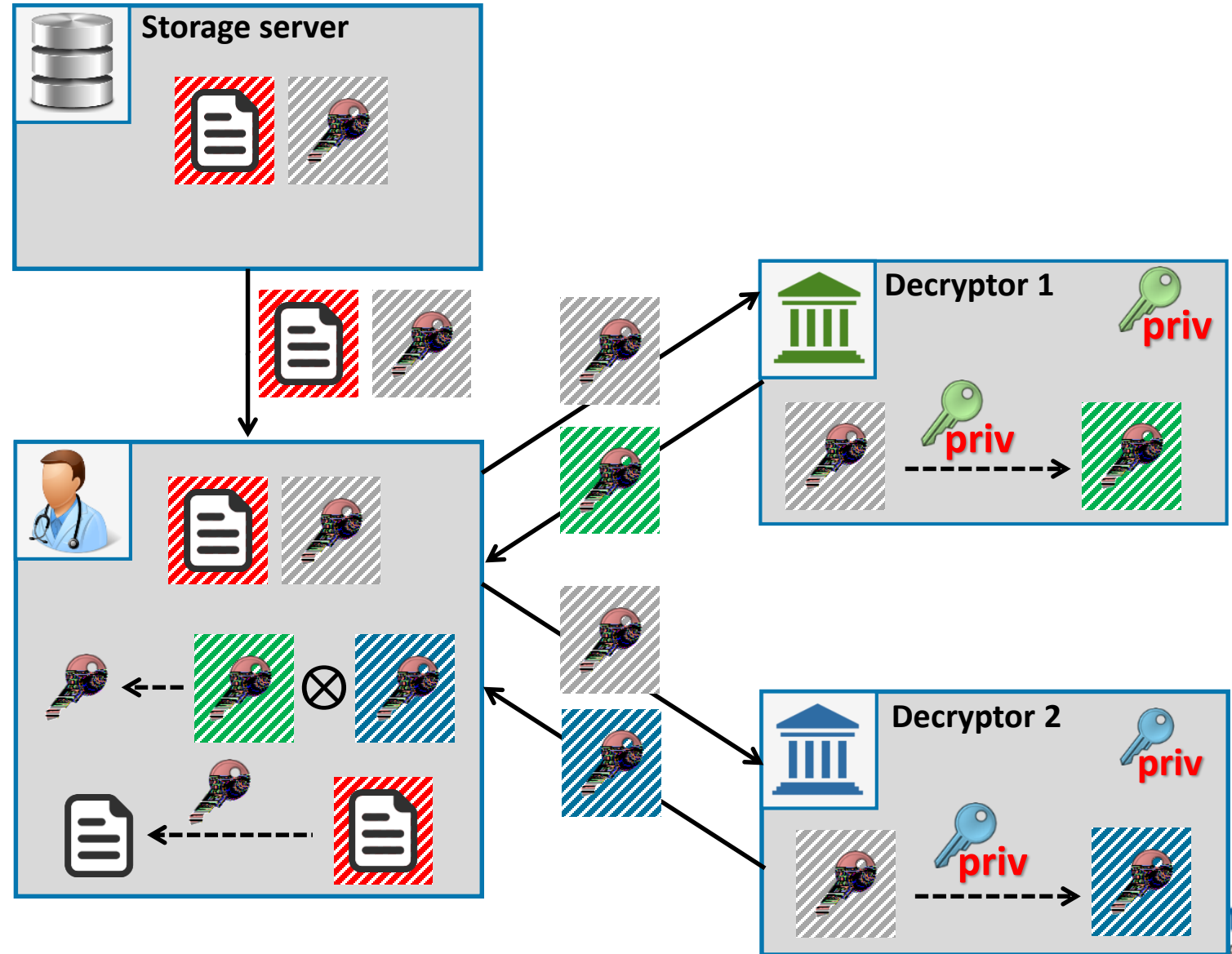
Getest:



Use case:



# Threshold encryption in Vitalink



- ▶ Zorgverleners kunnen digitale gegevens over hun patiënten met elkaar delen.
- ▶ Gegevens over vaccinaties, medicatie, bevolkingsonderzoeken en een samenvatting van het patiëntendossier van de huisarts.
- ▶ Hoge bescherming confidentialiteit dankzij threshold encryption.

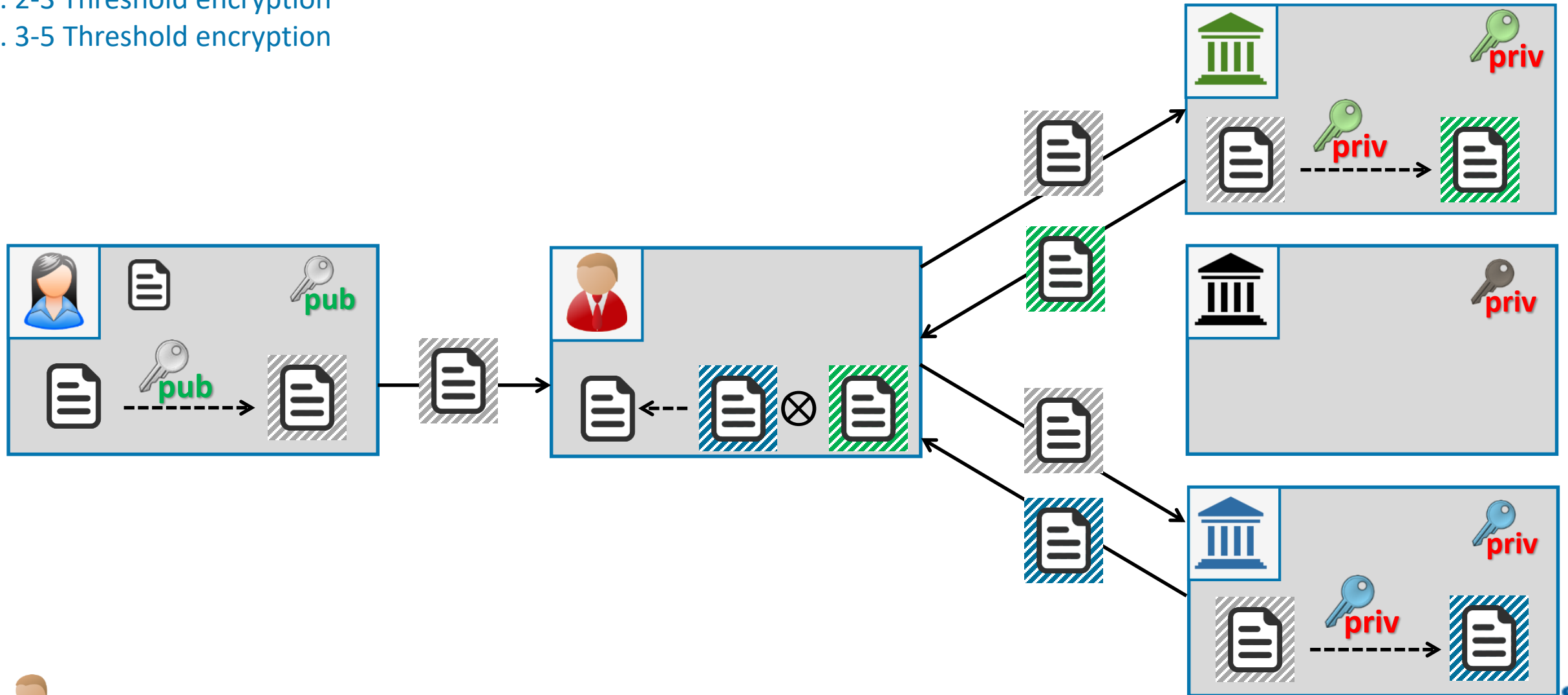
# Threshold encryption

## Opzet

Een quorum aan partijen vereist voor decryptie

Vb. 2-3 Threshold encryption

Vb. 3-5 Threshold encryption

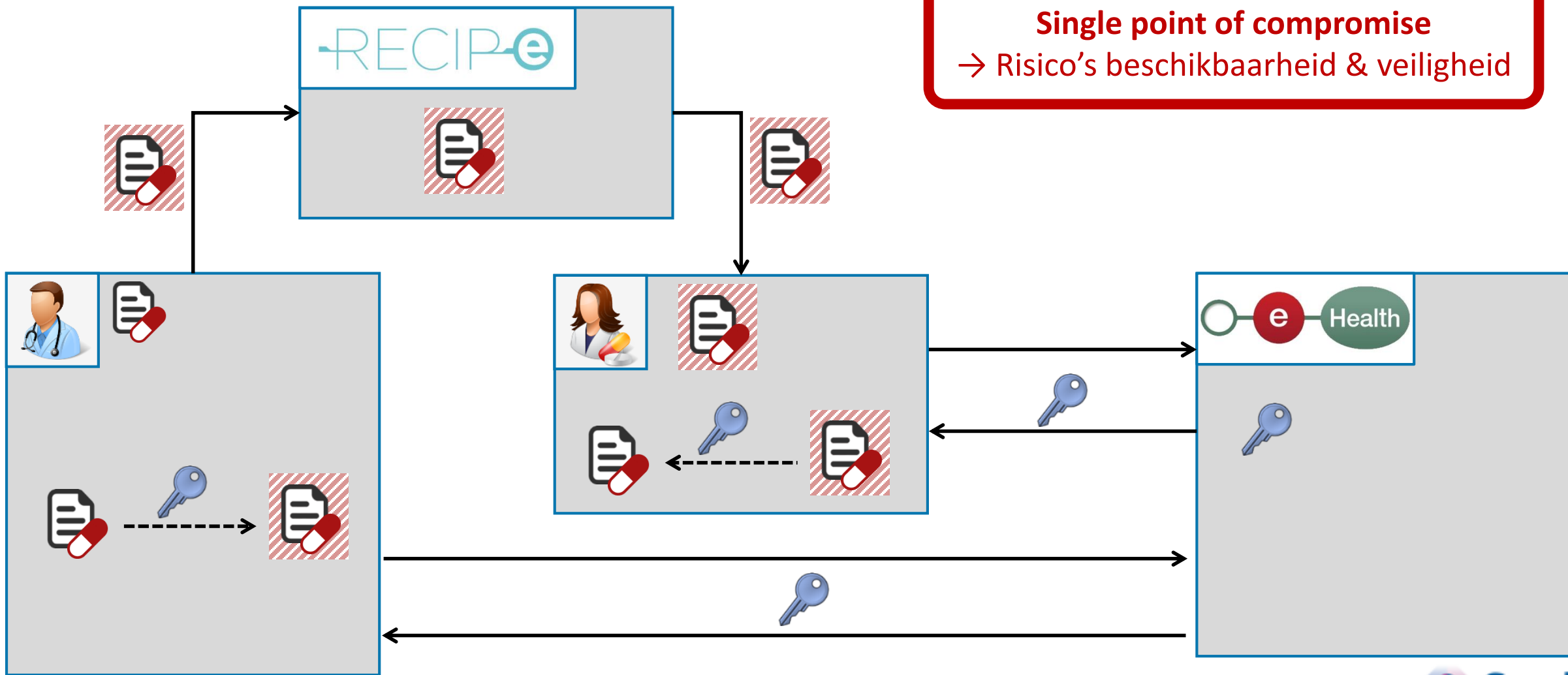


heeft medewerking nodig van twee van de drie decryptoren

# Threshold encryption – Medische voorschriften

Opzet  
Confidentialiteit & beschikbaarheid elektronische medische voorschriften

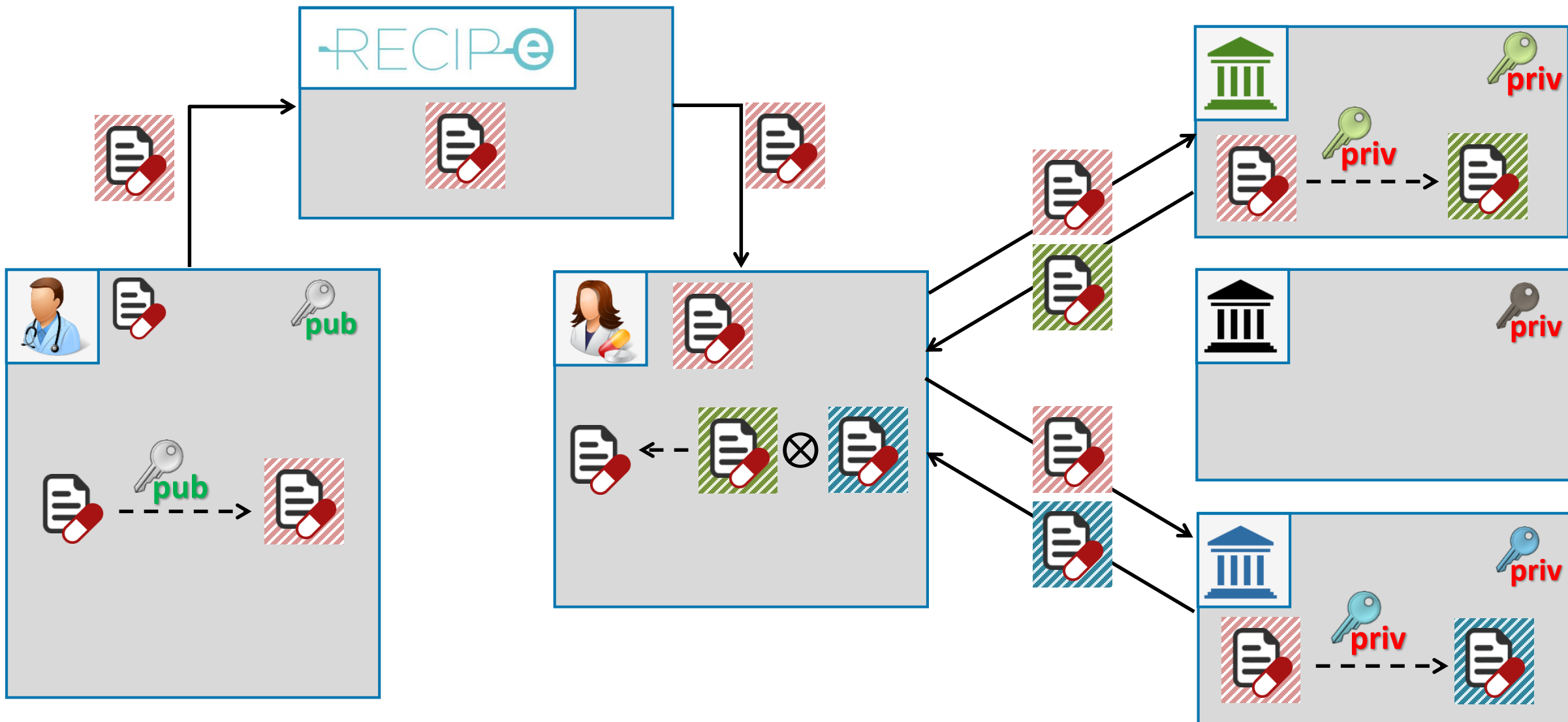
**Single point of failure**  
**Single point of compromise**  
→ Risico's beschikbaarheid & veiligheid



# Threshold encryption – Medische voorschriften

**Opzet**  
Confidentialiteit & beschikbaarheid elektronische medische voorschriften  
(vereenvoudigd)

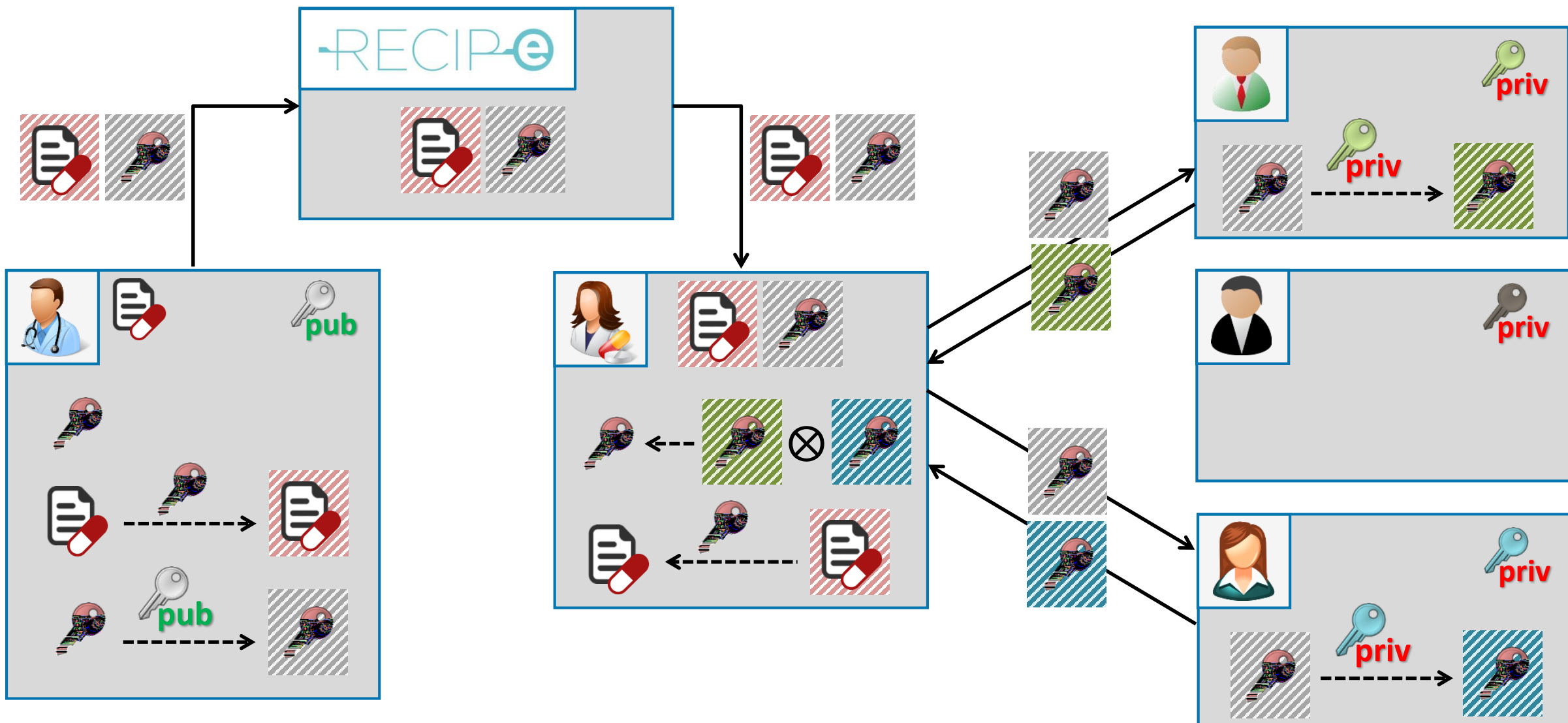
**Geen single point of failure / compromise**  
→ Hogere veiligheid & beschikbaarheid



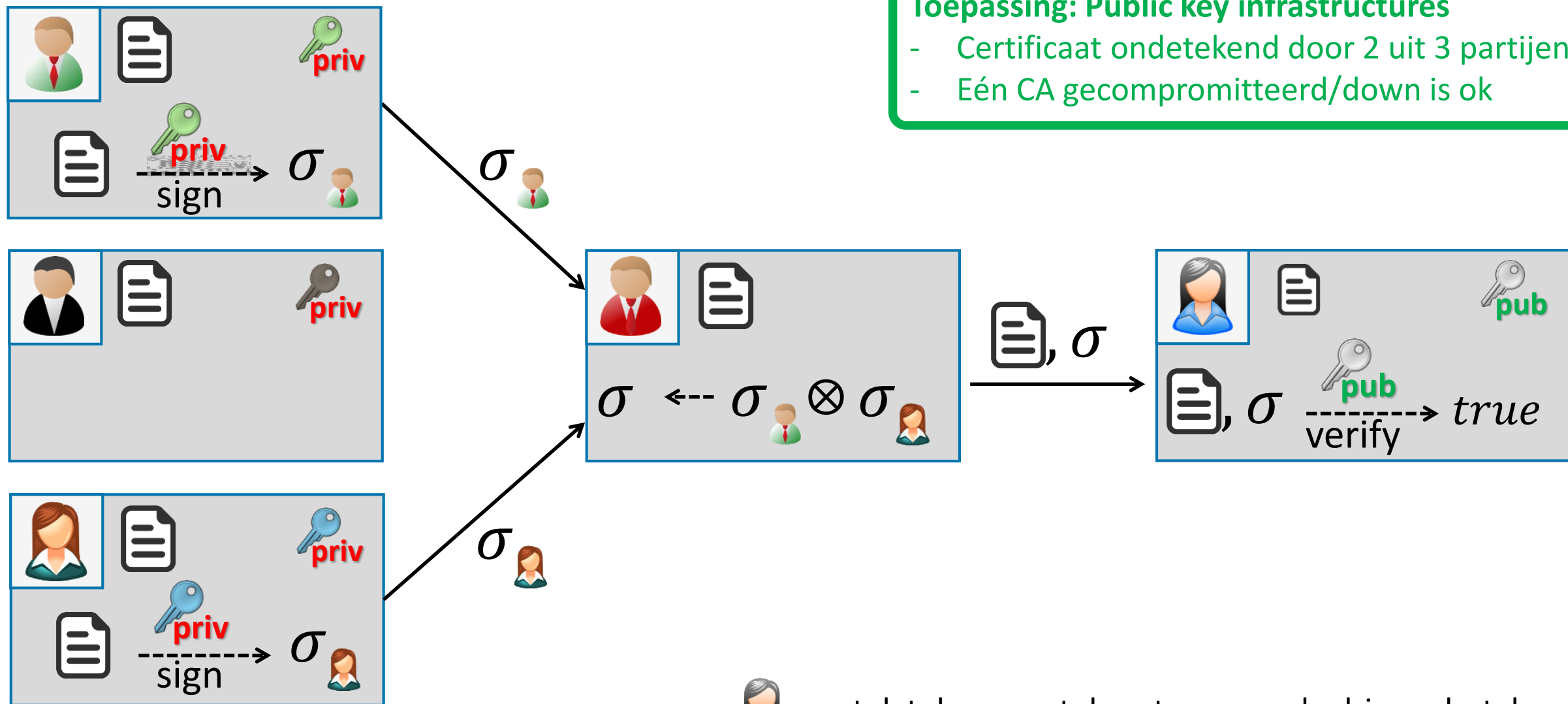
# Threshold encryption – Medische voorschriften

Opzet  
Confidentialiteit & beschikbaarheid elektronische medische voorschriften

Minder werk voor decryptoren  
Samenspannen wordt moeilijker




# Threshold signatures



**Toepassing: Public key infrastructures**

- Certificaat ondetekend door 2 uit 3 partijen
- Eén CA gecompromitteerd/down is ok

 weet dat document door twee van de drie ondertekend is  
Handtekening ziet er uit als normale handtekening

# Threshold encryption

Een quorum uit een set van partijen moet hun medewerking verlenen om een cijfertekst te decrypteren (of om een digitale handtekening te plaatsen)

Theorie:



Eigen code:

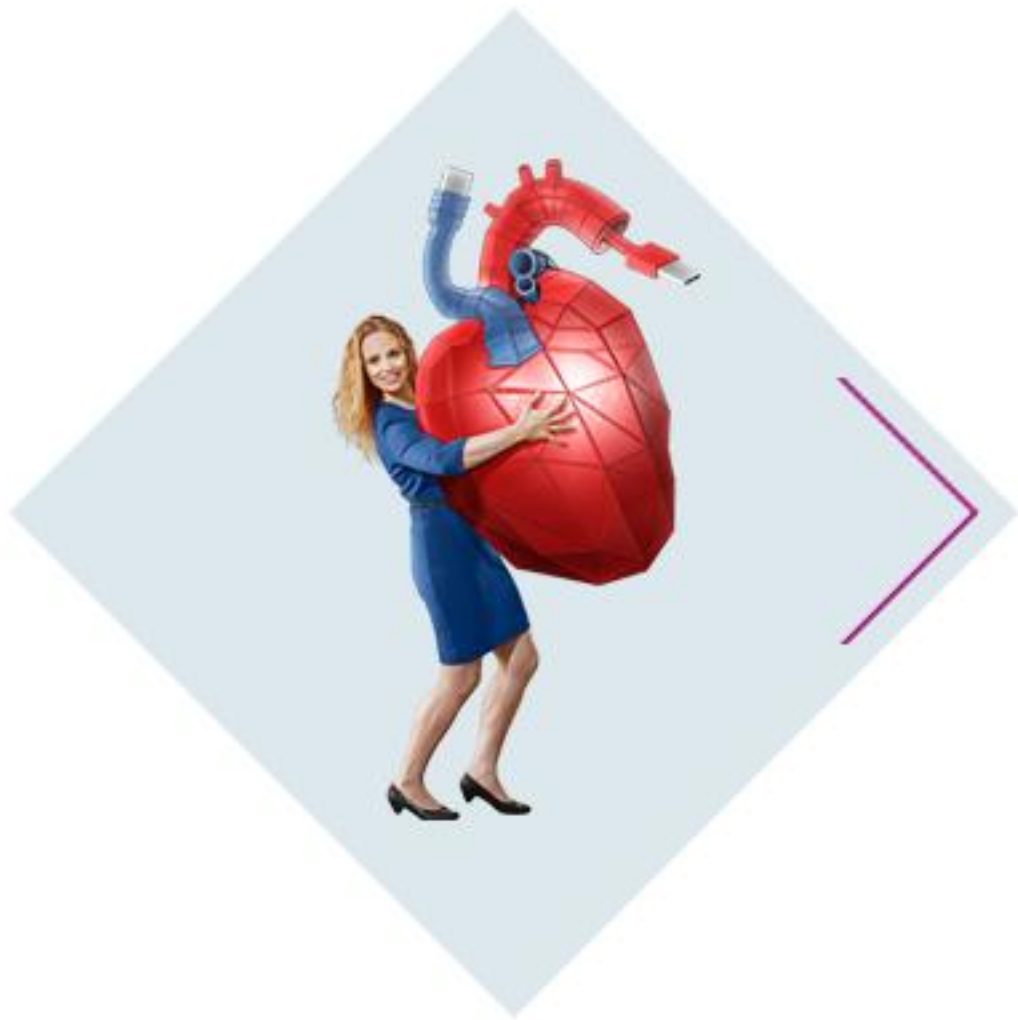


Getest:



Use case:





# Format-preserving encryption

Theorie:



Eigen code:



Getest:



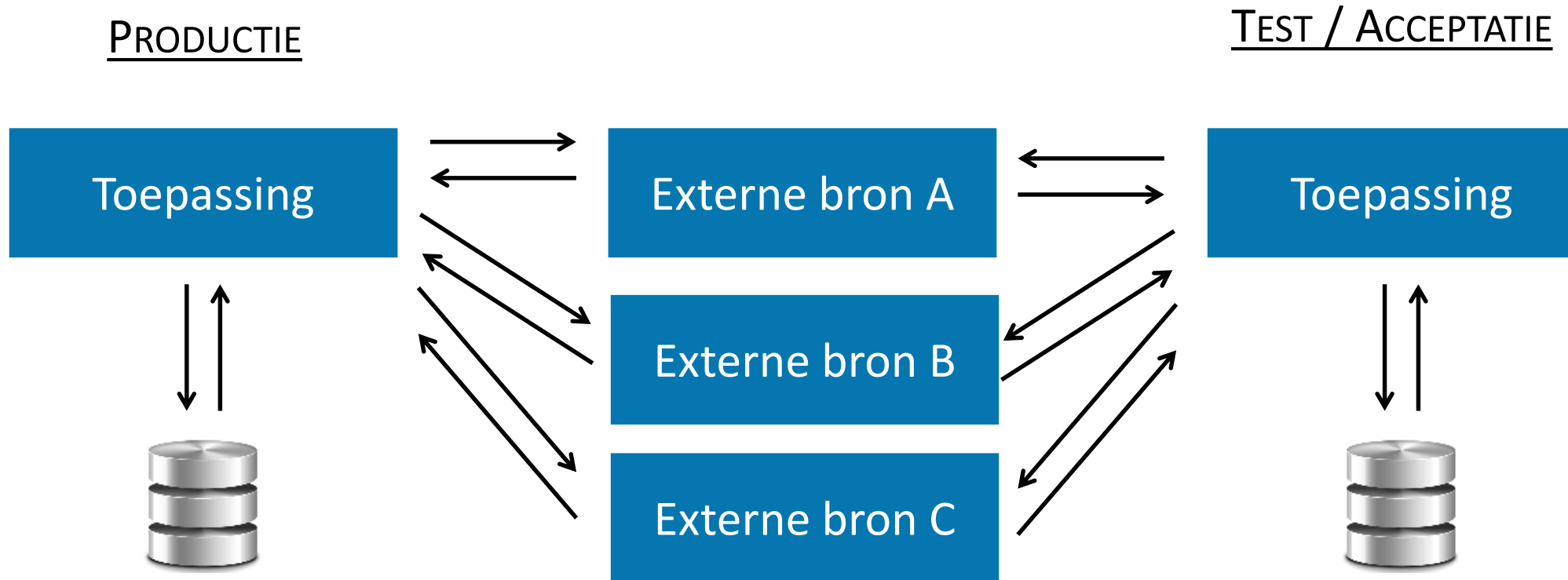
Use case:



## Probleemstelling

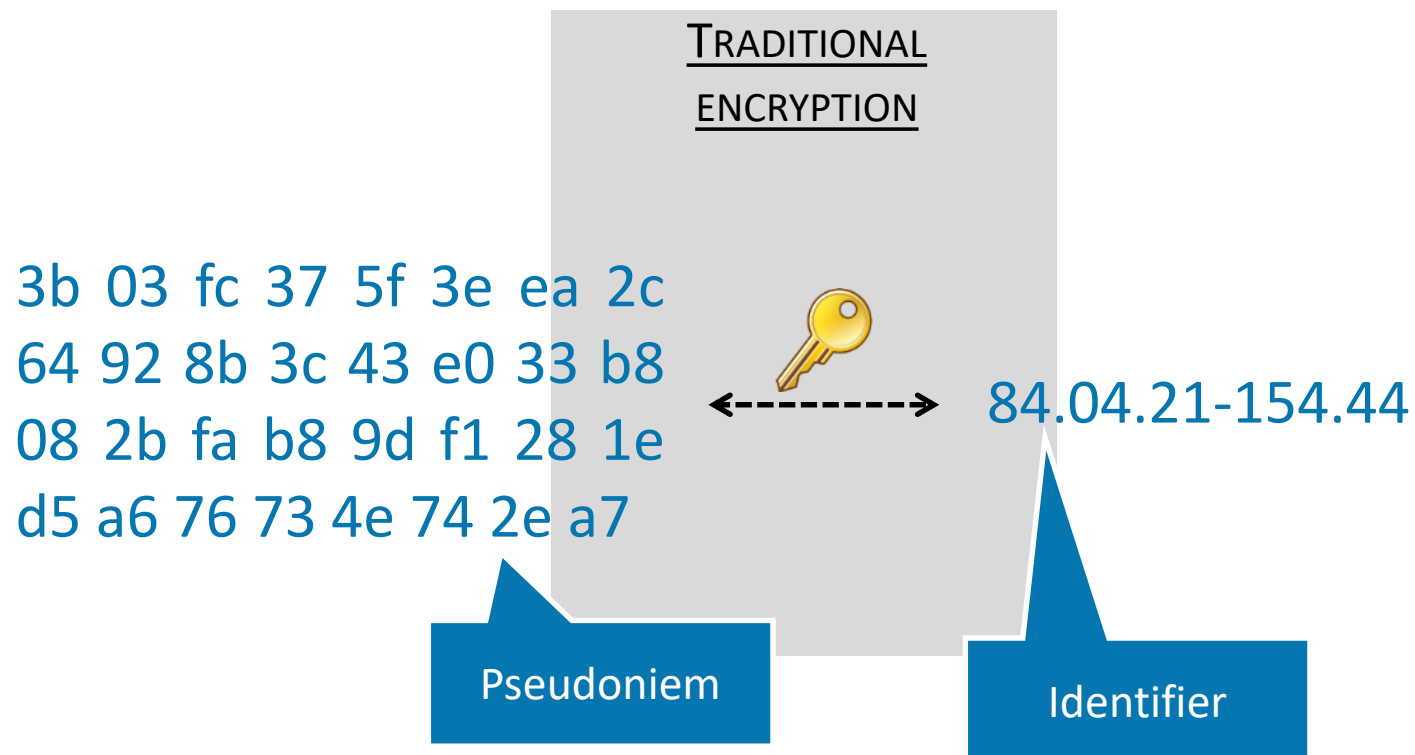
Vermijden identificeerbare persoonsgegevens in test & acc omgeving

Maar toch nog kunnen debuggen/testen



## Doel

Beschermen privacy burger  
(gestimuleerd door GDPR)



Identifier	Disease
84.04.21-154.44	Interstitial cystitis
92.11.07-087.63	Crohn's disease
55.07.27-231.66	Typhus
71.01.30.146.64	Cryptosporidiosis

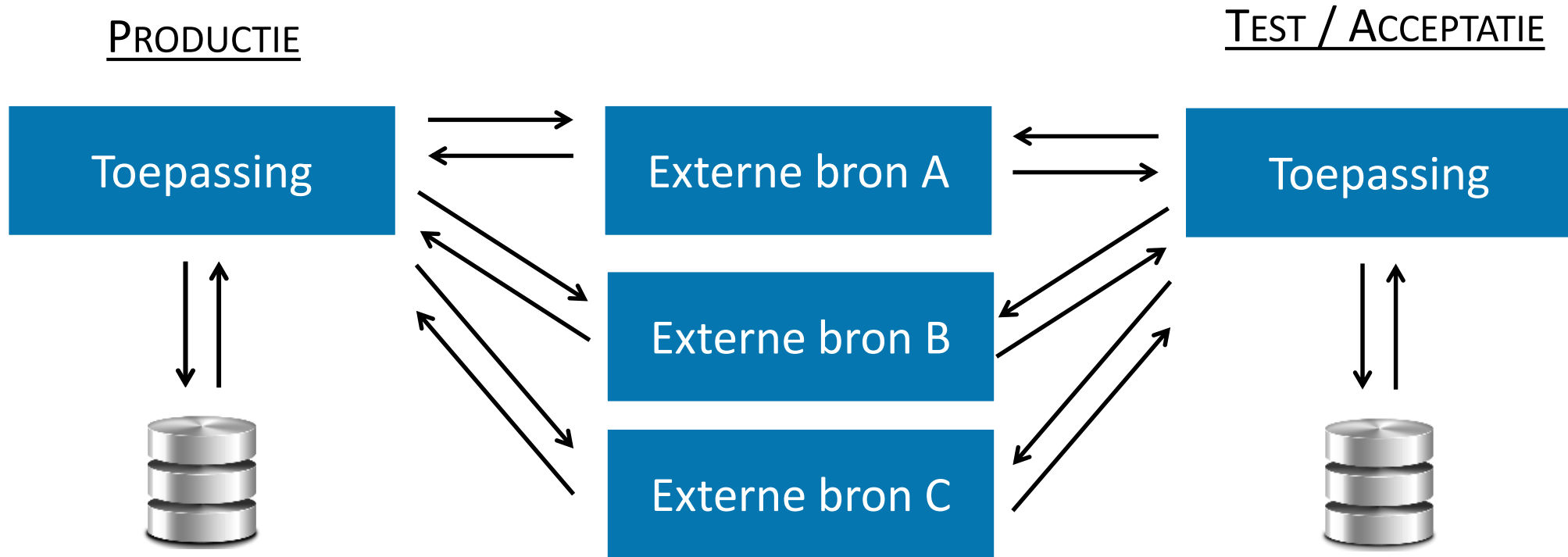
Identifier	Disease
3b 03 fc ... 2e a7	Interstitial cystitis
fb da 6c ... 7e 4f	Crohn's disease
1d 01 a8 ... 17 13	Typhus
52 a7 53 ... a4 2d	Cryptosporidiosis



# Format-preserving encryption

## Doel

Vermijden identificeerbare persoonsgegevens in test & acc omgeving



## **Pseudonimiseren d.m.v. encryptie onvoldoende**

- Databasevelden moeten aangepast worden
- Applicatie verwacht structuur RRN en zal niet langer correct werken

## **Gewenste vorm pseudoniem**

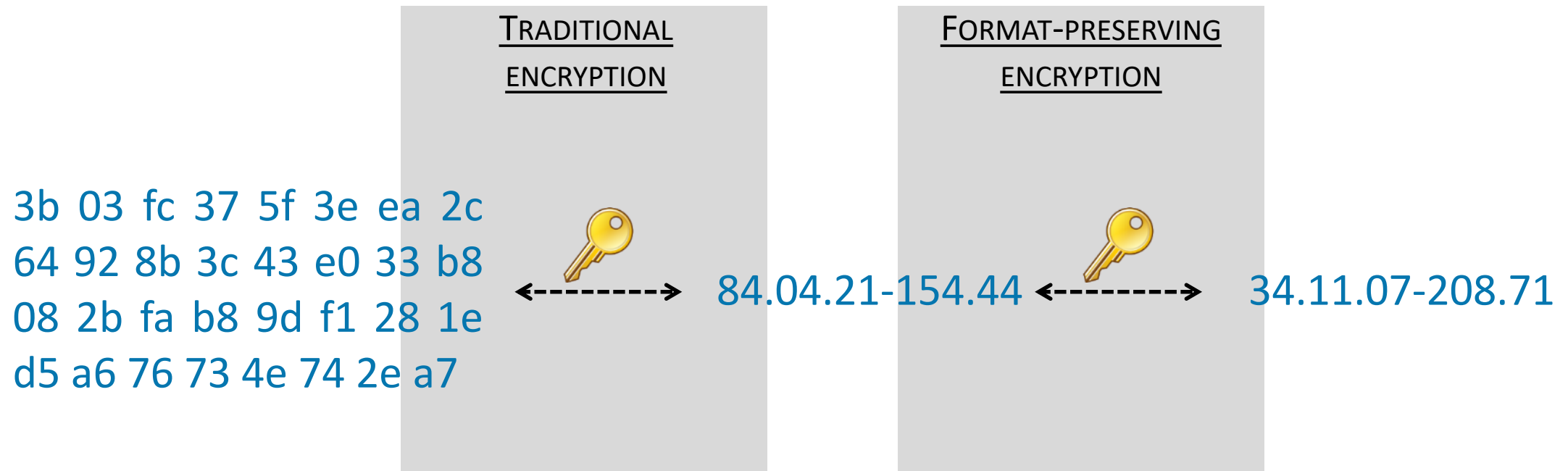
- ▶ Behoudt structuur rijksregisternummer
- ▶ Behoudt eventueel deel van informatie

# Format-preserving encryption

## Opzet

Bewaren structuur / formaat van originele data na vercijfering

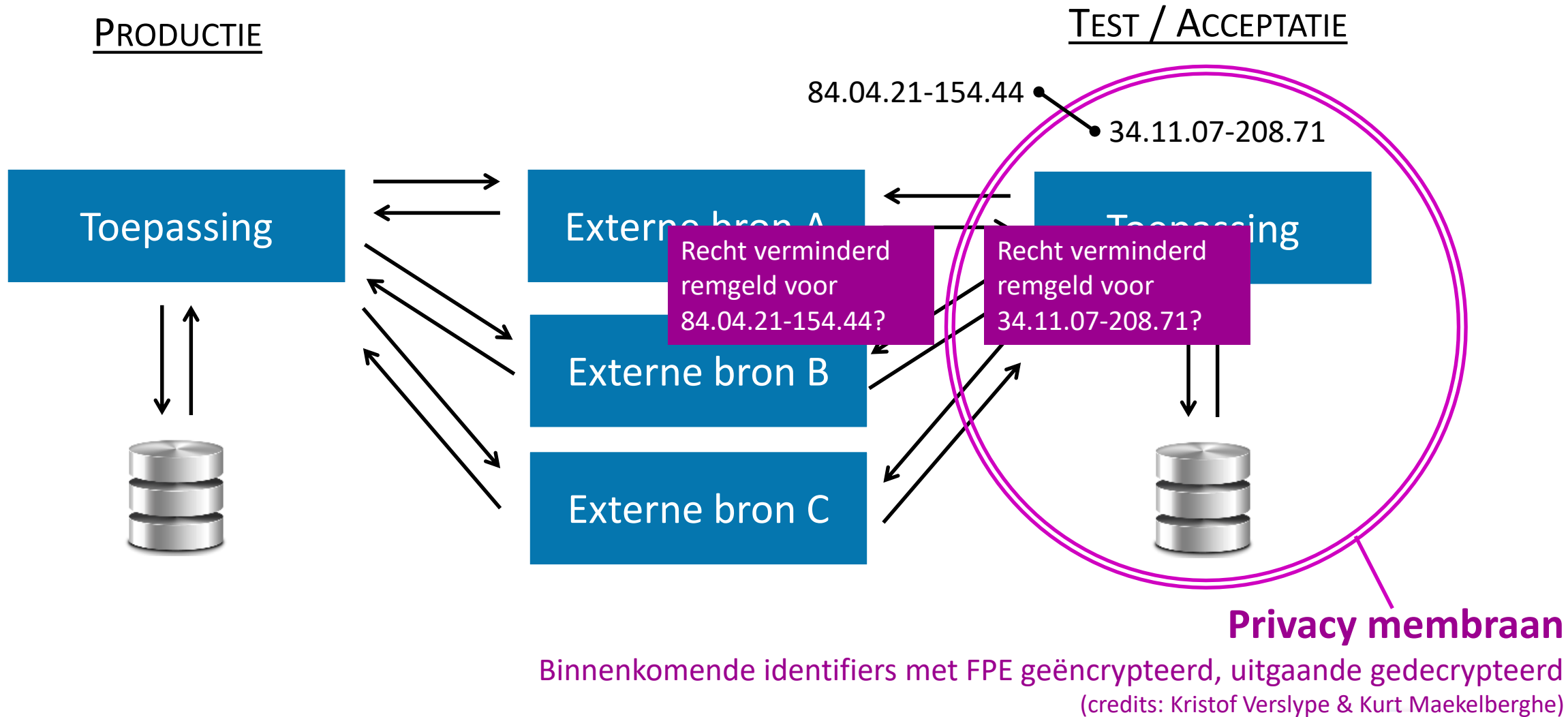
[optioneel] bewaren informatie (vb. correcte checksum, geslacht, leeftijdscategorie)



# Format-preserving encryption

## Doel

Vermijden identificeerbare persoonsgegevens in test & acc omgeving

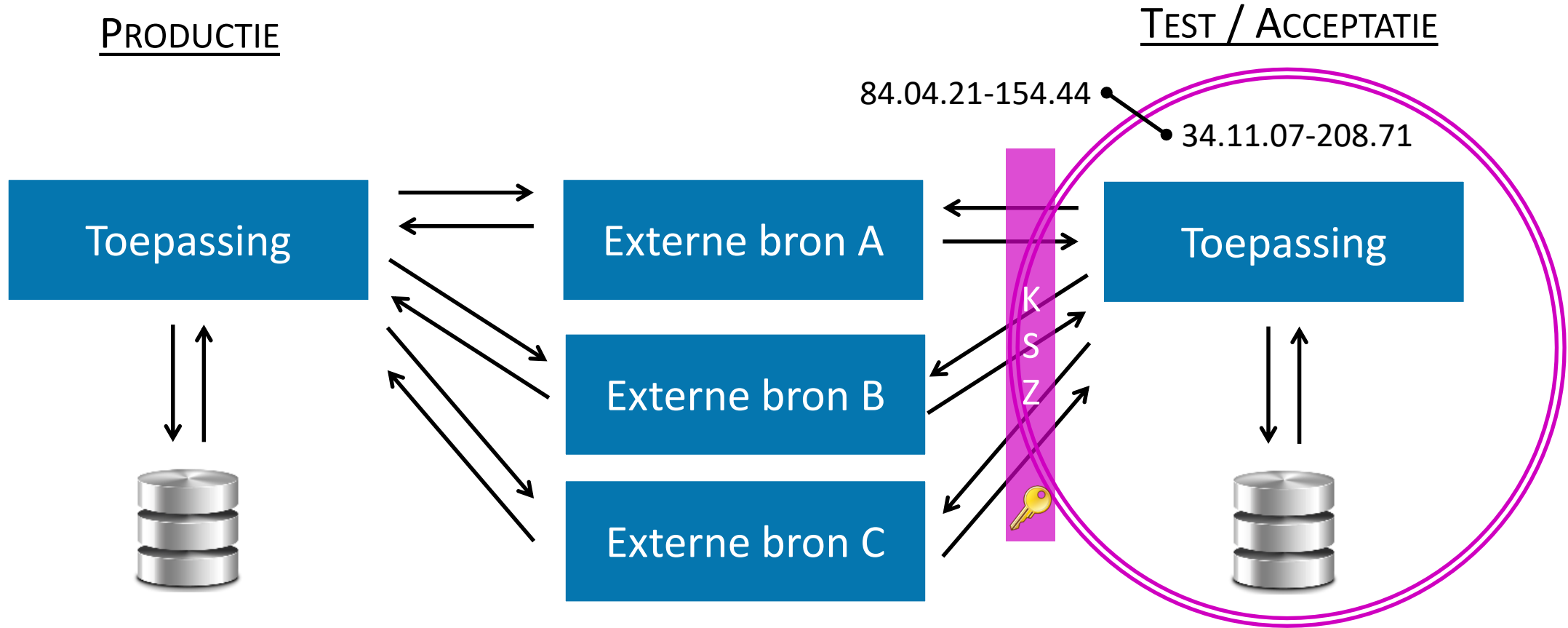


Binnenkomende identifiers met FPE geëncrypteerd, uitgaande gedecrypteerd  
(credits: Kristof Verslype & Kurt Maekelberghe)

# Format-preserving encryption

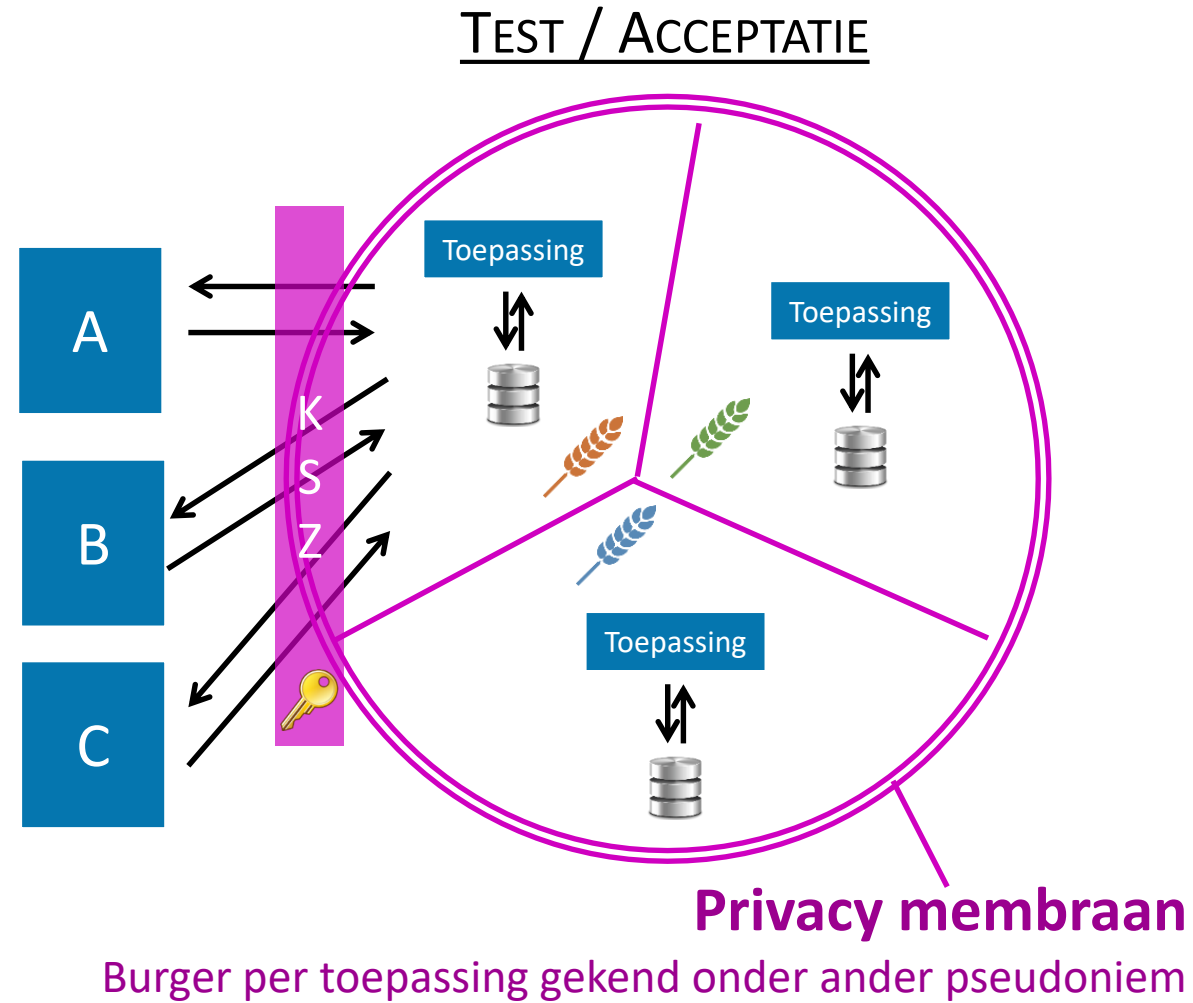
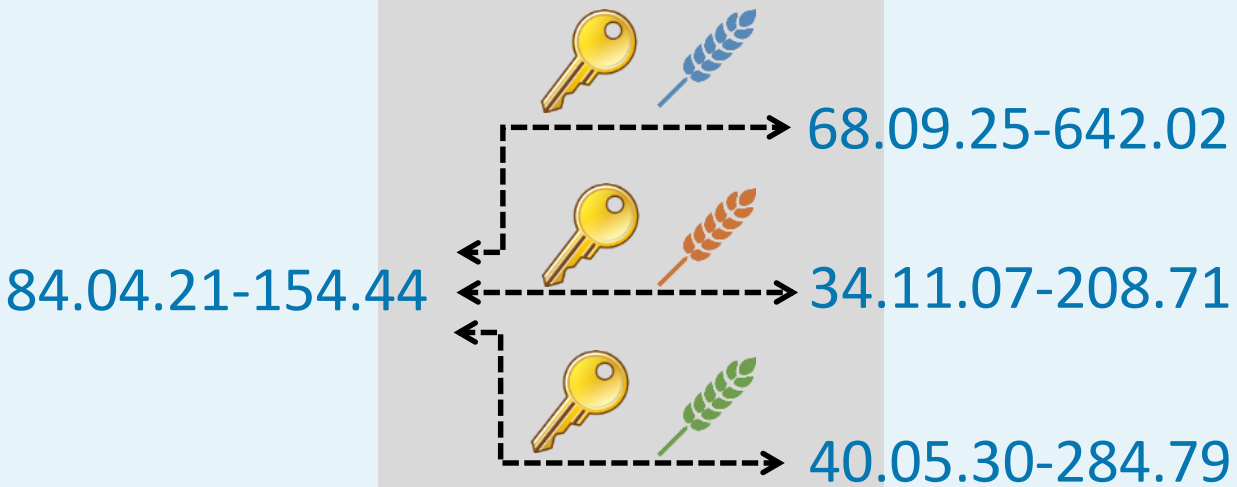
## Doel

Vermijden identificeerbare persoonsgegevens in test & acc omgeving



# Format-preserving encryption

## FORMAT-PRESERVING ENCRYPTION



## Implementatie

Structuur rijksregisternummers (Smals Research)

Format-preserving encryption (Github)

Crypto werkpaard (AES of HMAC)



## Performantie

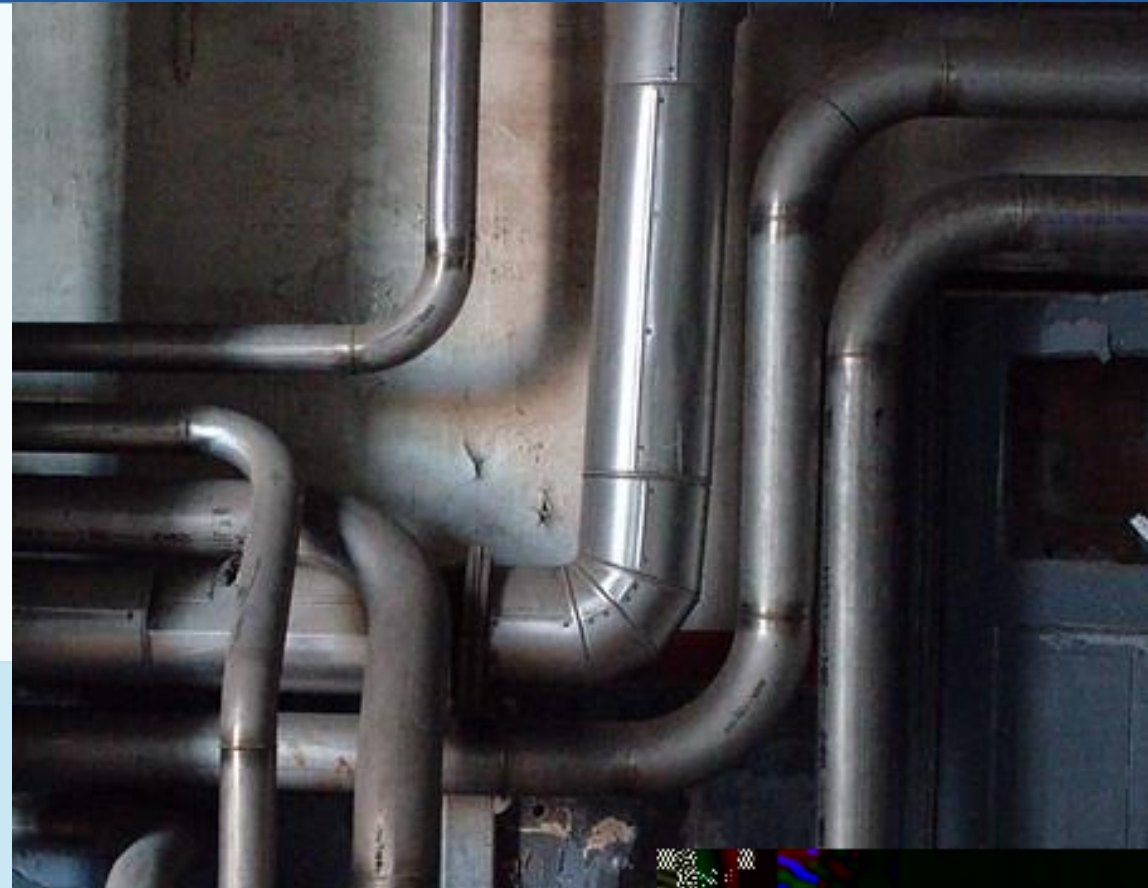
100 000 rijksregisternummers

256 bit security

► **Encryption: 23s**

► **Decryption: 27s**

Data in-memory, Lenovo Thinkpad L570, Windows 10, Intel Core i5-6300 CPU @ 2,40Ghz, 16GB





## Toepassingen

- ▶ Pseudonimiseren data voor dev / testen / bug-fixing
- ▶ Pseudonimiseren data in legacy applicatie zonder aanpassen DB

## Maturiteit

- ▶ Sinds 2002
- ▶ Gestandaardiseerd in NIST 800-38G (2016)
- ▶ Op roadmap BouncyCastle
- ▶ Beperkte ondersteuning HSMs

## Overleg

- ▶ Privacy membraan

# Format-preserving encryption

De cijfertekst behoudt de structuur van de oorspronkelijke data

Theorie:



Eigen code:



Getest:



Use case:





# Proxy re-encryption

Theorie:



Eigen code:



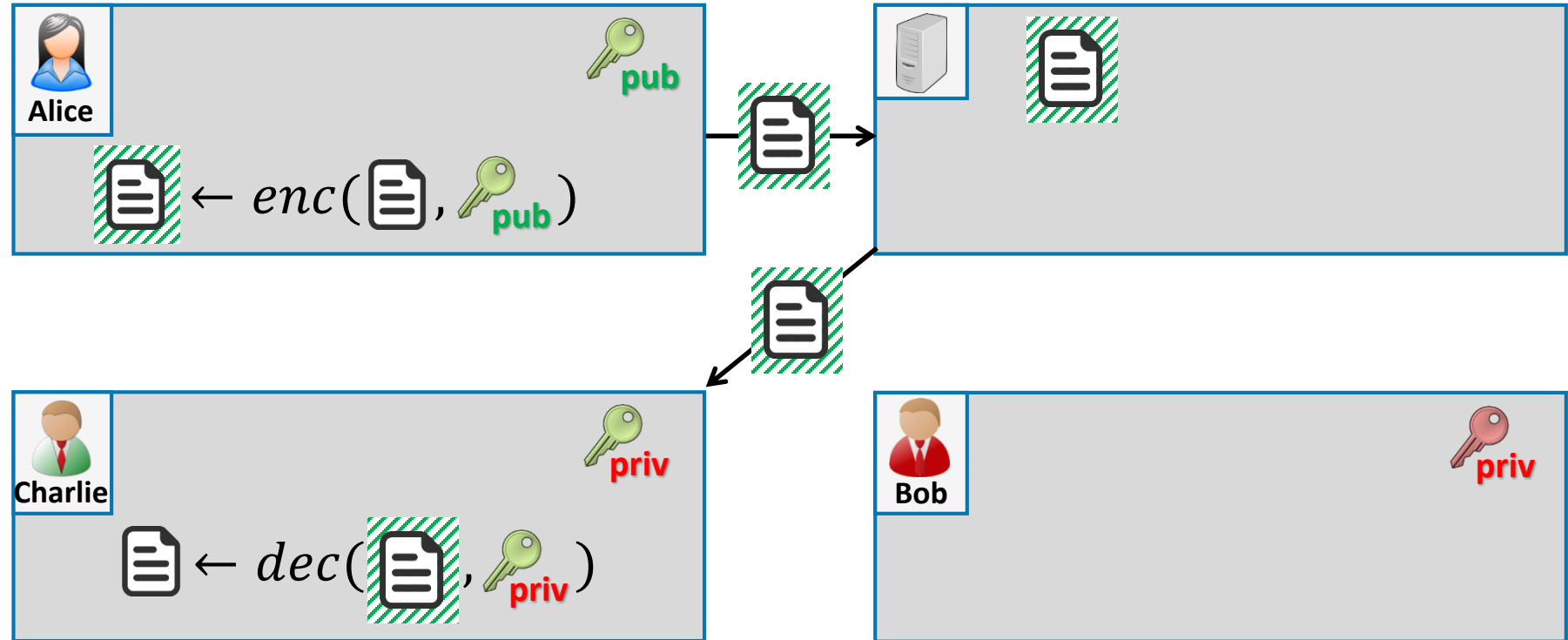
Getest:



Use case:



# Proxy re-encryption



## Probleemstelling

 gaat op reis en wil berichten forwarden naar  .

# Proxy re-encryption

## Doel

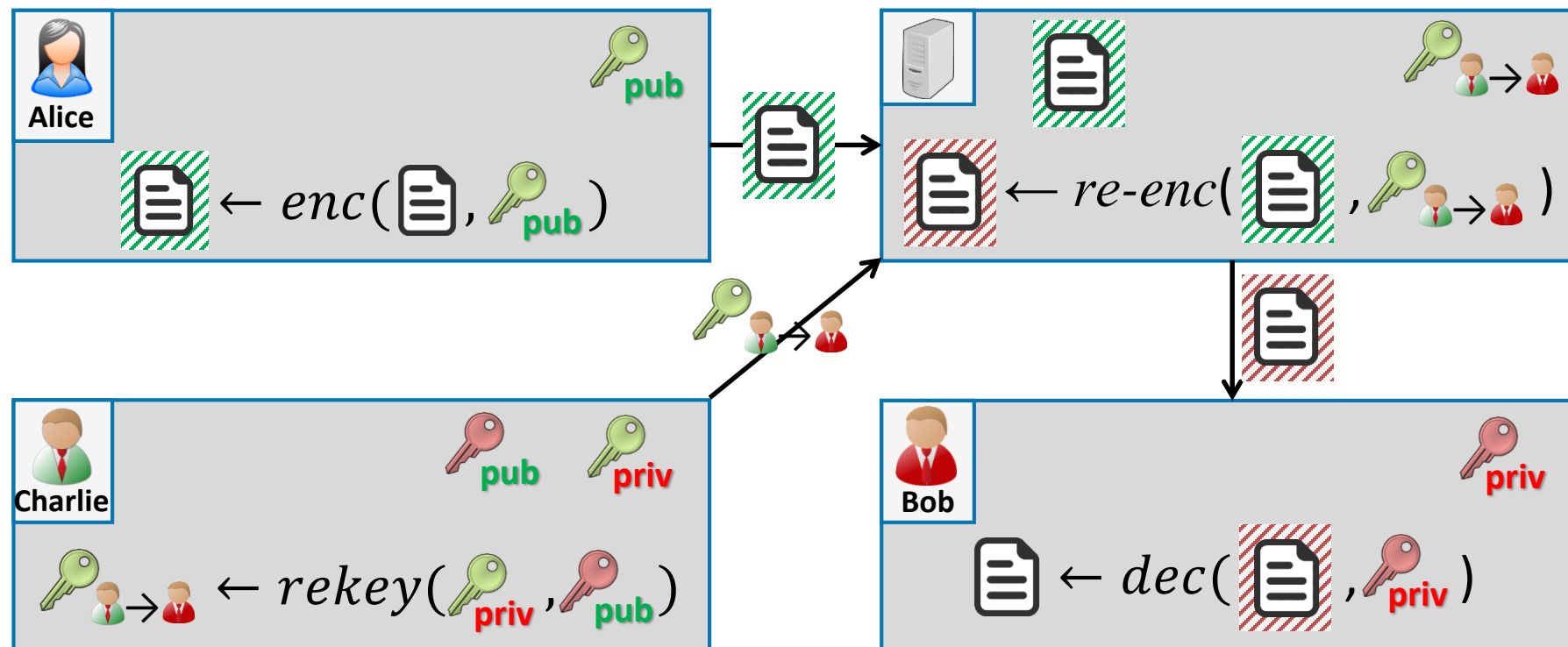
Conversie cijfertekst die decrypteerbaar is door een andere entiteit, zonder inhoud te zien

## Eigenschappen

ziet inhoud niet

komt niet te weten

Voor  verandert er niets



## Delegeren toegangsrechten

vb. e-mail, eHealth platform,  
KSZ

## File sharing (dropbox)

Eigenaar data geeft server  
reencryption key om anderen  
toegang te geven

## Toezicht

Autoriteit krijgt toegang tot  
vercijferd dataverkeer na  
toestemming ontvanger

# Proxy re-encryption

Omzetten cijfertekst,  
zonder de inhoud te zien,  
zodat de cijfertekst decrypteerbaar is  
door een andere partij

Theorie:



Eigen code:



Getest:



Use case:



# Geavanceerde cryptografie

## Speciale cijfertekst

- 1 Threshold encryption
- 2 Format-preserving encryption
- 3 Proxy reencryption

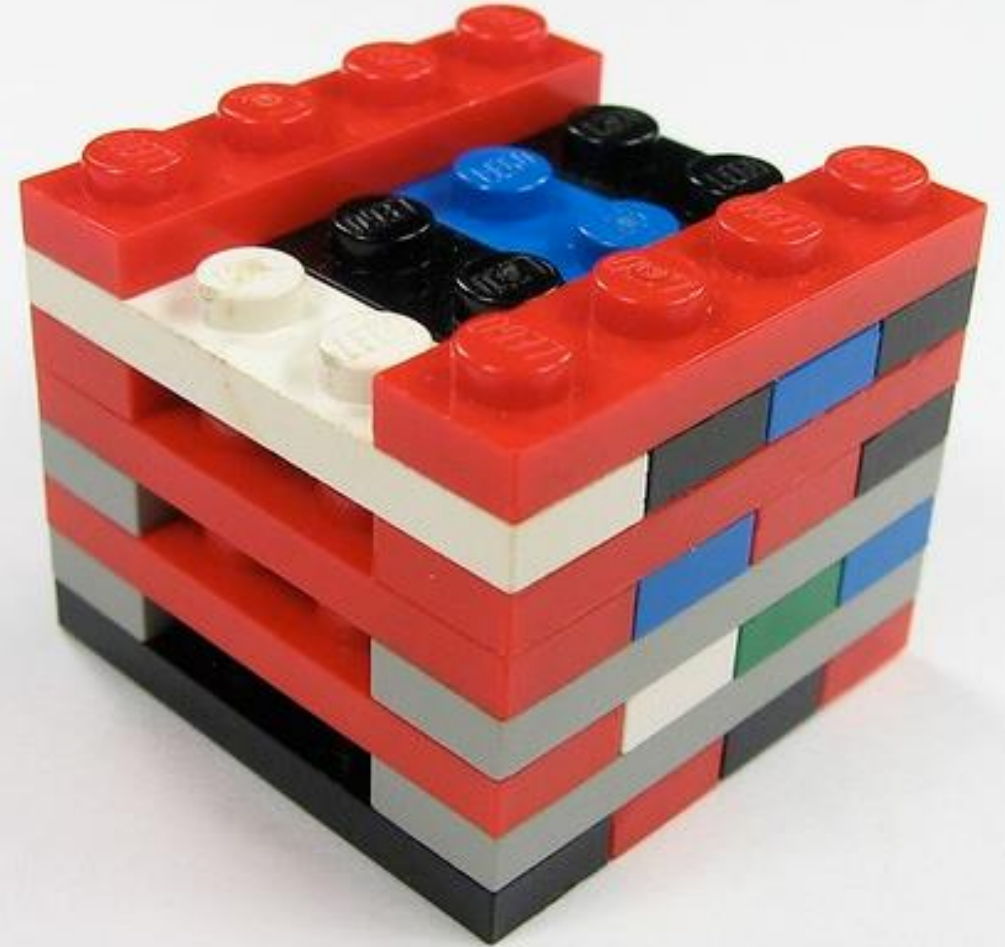
## Authenticatie

- 4 Secure remote password protocol
- 5 Attribute-based credentials

---

## Privacyvriendelijke opvraging

- 6 Secure multiparty computation
- 7 Oblivious transfer
- 8 Private set intersection
- 9 Oblivious join



Er is veel meer!



# Secure remote password protocol

Theorie:



Eigen code:



Getest:



Use case:



# Thousands of hacked Disney+ accounts are already for sale on hacking forums

Hackers began hijacking accounts hours after Disney+ launched earlier this week.





By [Catalin Cimpanu](#) for [Zero Day](#) |  
November 16, 2019 -- 08:00 GMT (08:00  
GMT) | Topic: [Security](#)

Hackers didn't waste any time and have started hijacking Disney+ user accounts hours after the service launched.

Many of these accounts are now being offered for free on hacking forums, or available for sale for prices varying from \$3 to \$11, a *ZDNet* investigation has discovered.

## A STREAM OF USER COMPLAINTS

The Disney+ video streaming service launched this week, on November 12. The service, although being available only in the US, Canada, and the Netherlands, has already amassed more than 10 million customers in its first 24 hours.

[Home](#)[Notify me](#)[Domain search](#)[Who's been pwned](#)[Passwords](#)[API](#)[About](#)[Donate](#)  

## Pwned Passwords

Pwned Passwords are 551,509,767 real world passwords previously exposed in data breaches. This exposure makes them unsuitable for ongoing use as they're at much greater risk of being used to take over other accounts. They're searchable online below as well as being downloadable for use in other online systems. [Read more about how HIBP protects the privacy of searched passwords.](#)

Oh no — pwned!

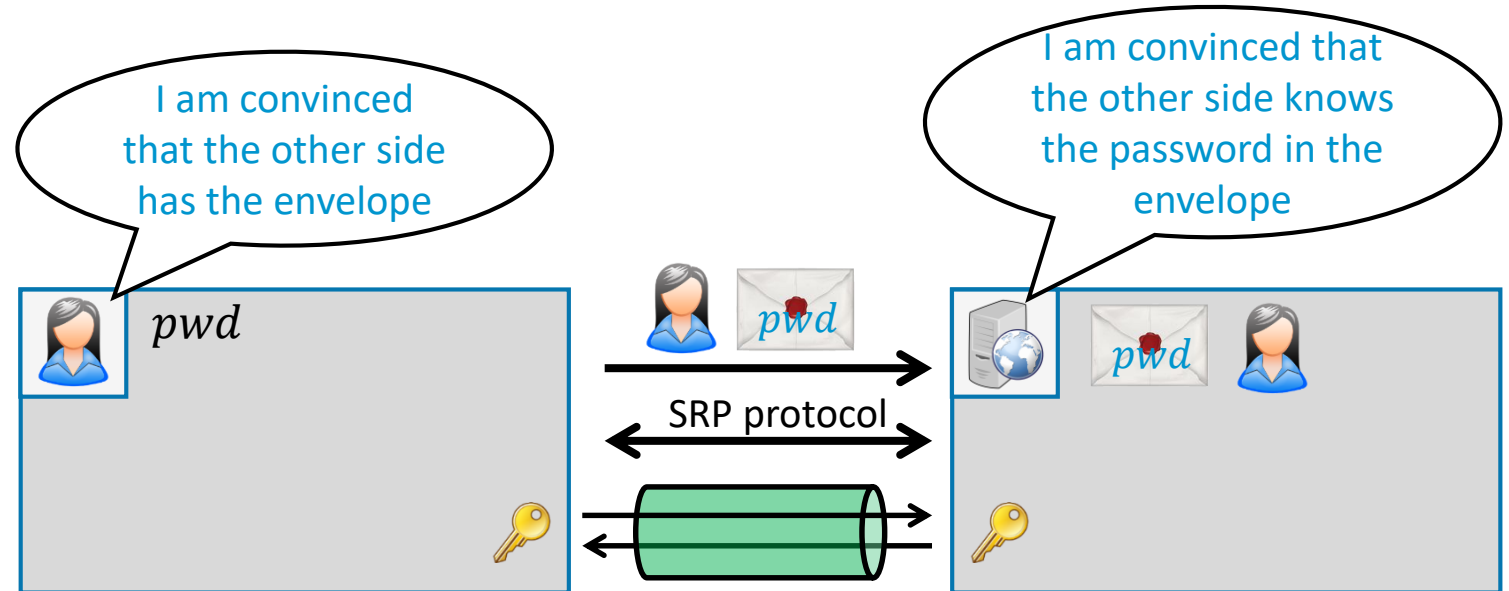
This password has been seen 36.320 times before

This password has previously appeared in a data breach and should never be used. If you've ever used it anywhere before, change it!

## Maximale veiligheid op basis van paswoord

- ▶ Paswoord verlaat nooit toestel eindgebruiker
- ▶ Raden paswoord moeilijk
- ▶ Server weet niet of twee verschillende users zelfde paswoord gebruiken
- ▶ Wederzijds geauthenticeerd kanaal

SRP is meest populaire PAKE  
(Password authenticated key exchange)



# Secure remote password protocol - Performantie

## Java / Bouncy Castle

Key length	Registratie client	Authenticatie client-side	Authenticatie server-side
2048 bit (112 bit security)	10ms	60ms	60ms
3072 bit (128 bit security)	20ms	200ms	200ms
4096 bit	30ms	350ms	350ms
6144 bit	40ms	1100ms	1100ms
8192 bit	70ms	3000ms	3000ms

Symmetric	DH or RSA	ECC
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

## Opmerkingen

- ▶ Performantie Javascript te testen
- ▶ OPAQUE is efficiënter dan SRP



## Hoge maturiteit

- ▶ Sinds 2001, momenteel versie 6
- ▶ Ondersteund door SSL/TLS (TLS-SRP), EAP, SAML, ...
- ▶ In diverse libraries zoals BouncyCastle & OpenSSL
- ▶ Standaardizatieprojecten: IEEE P1363 en ISO/IEC 11770-4
- ▶ Patentvrij

## Nauwelijks adoptie

- ▶ Gebruikt door o.a. Apple iCloud Key Vault, protonmail, 1Password, ...
- ▶ Complexer dan klassiek paswoord formulier
- ▶ Beperkte bekendheid (Cryptograaf ≠ marketing boy).

<https://blog.cryptographyengineering.com/2018/10/19/lets-talk-about-pake/>

[https://protonmail.com/blog/encrypted\\_email\\_authentication/](https://protonmail.com/blog/encrypted_email_authentication/)

<https://1password.com/files/1Password%20for%20Teams%20White%20Paper.pdf>

## Veiligheidsniveau

- ▶ Hoogst mogelijke beveiliging op basis van paswoord
- ▶ Factor in multi-factor authenticatie

Auth. level	Authentication means	Tech. level
<b>High</b> ↑	eID	500
	Itsme	450
	Mobile app / SMS OTP	400
	Token	300
<b>Low</b>	Username/password	200
<i>Without identification</i>	Self-registration without usage NRN	100



Sterke bescherming van  
server-side paswoorden

Veilig kanaal na  
geslaagde authenticatie  
op basis van paswoord

## Secure remote password protocol

Theorie:



Eigen code:



Getest:



Use case:





# Attribute-based credentials

Theorie:



Eigen code:



Getest:



Use case:



## Scenario

Burger wil alcohol/vuurwerk aankopen en moet bewijzen dat hij/zij ouder is dan 18 jaar

## Traditionele oplossing

Belgische eID kaart

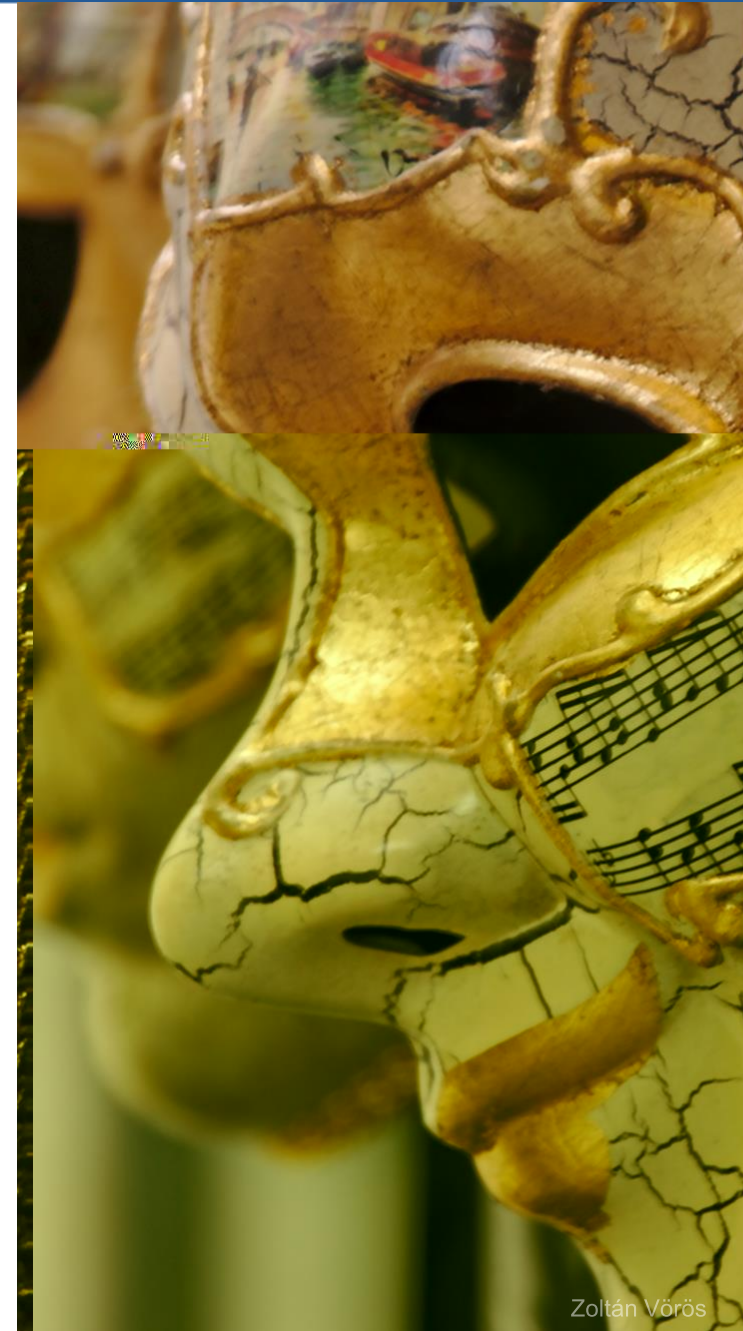


## Privacy

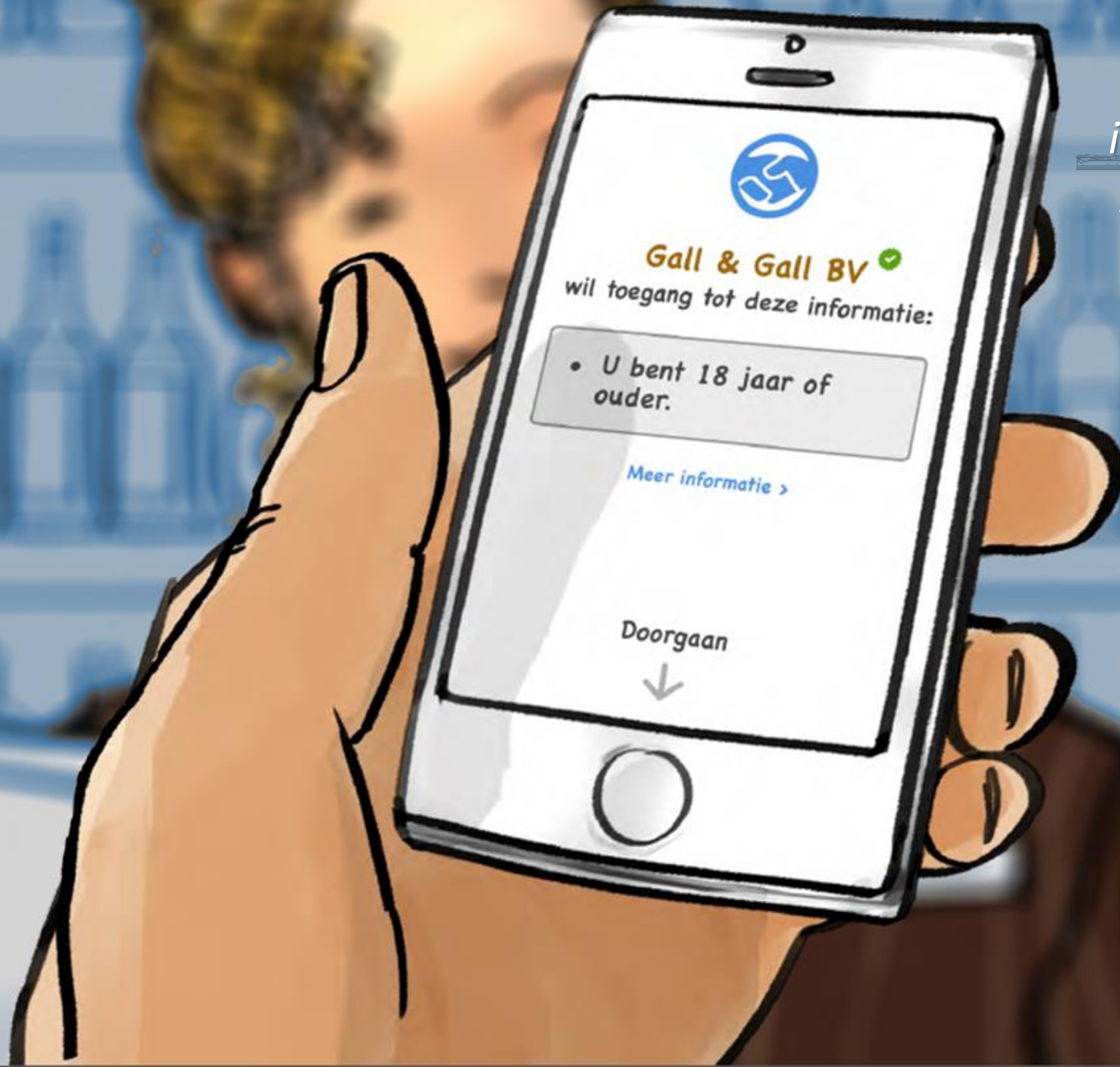
Handelaar komt ook je volledige naam, rijksregisternummer, geslacht, exacte geboortedatum, etc. te weten

## Probleemstelling

Kan de burger ENKEL bewijzen dat hij 18+ is, waarbij de afhankelijkheid van een derde partij geminimaliseerd wordt?



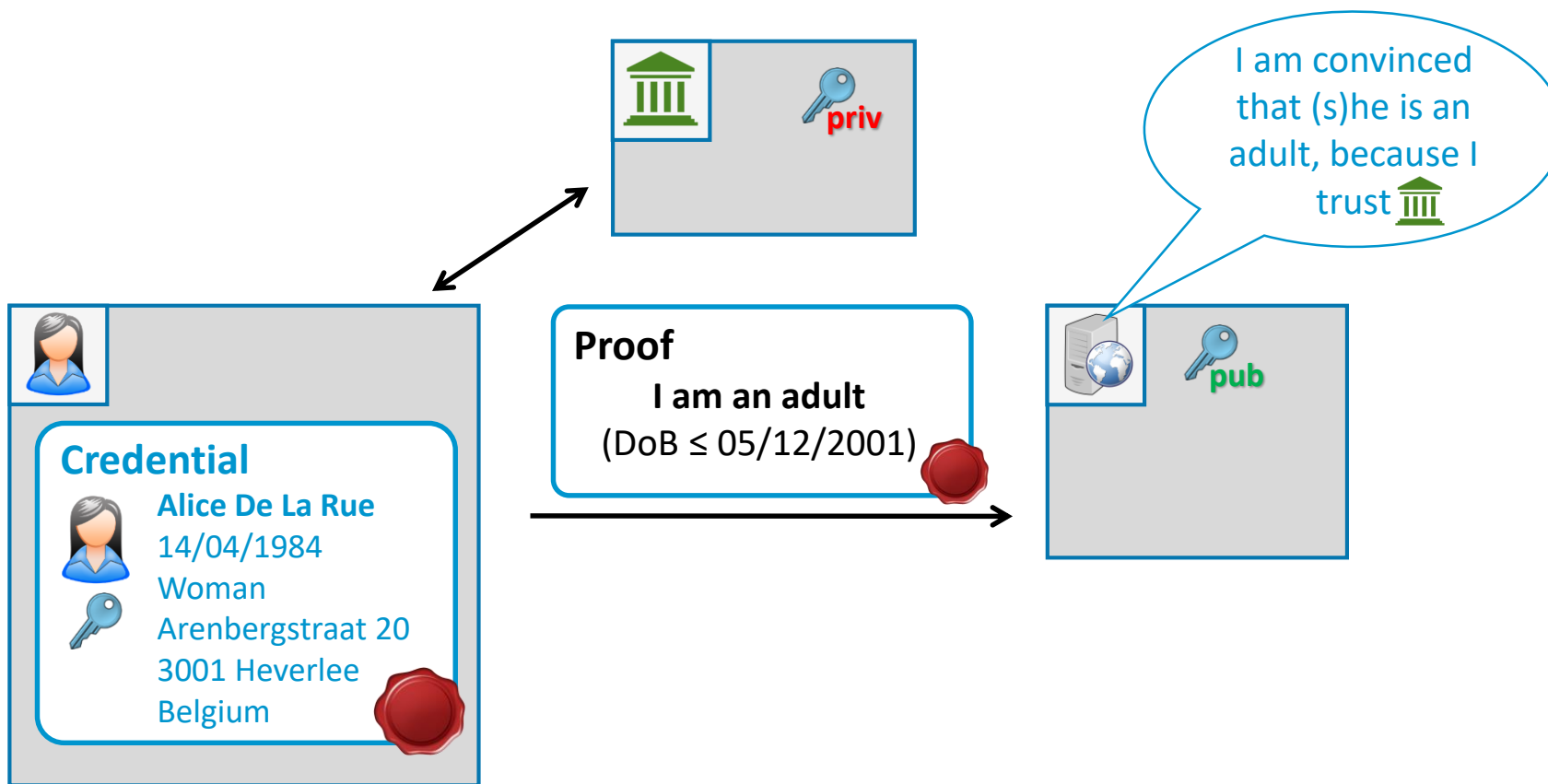
*“de combinatie van alle informatie die op een identiteitsbewijs zichtbaar is kan misbruikt worden.”*



### **Voordelen**

- *Veiligheid & privacy voor burger*
- *Burger bepaalt zelf met wie gegevens gedeeld worden*

# Attribute-based credentials



## Selectieve prijsgave

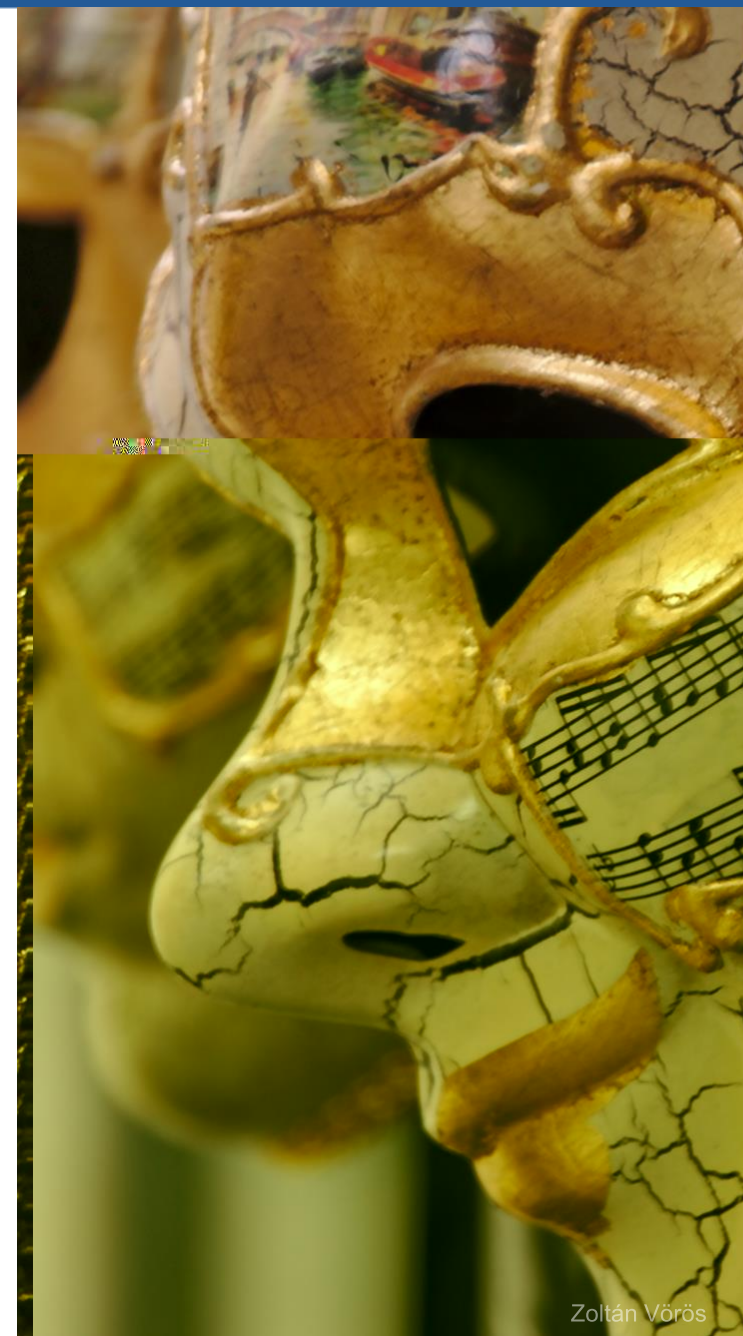
Burger kiest welke informatie hij/zij toont

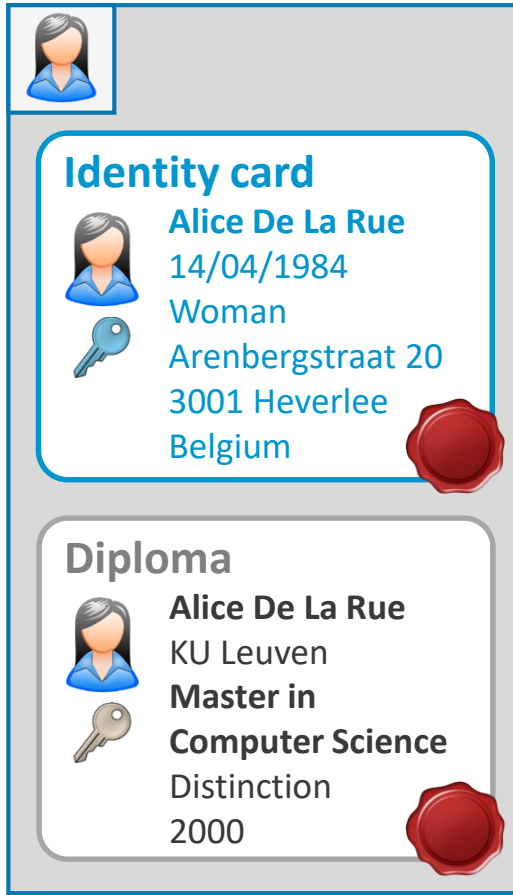
## Onlinkbaarheid

Server kan twee authenticaties zelfde persoon niet linken

## Conditionele privacy

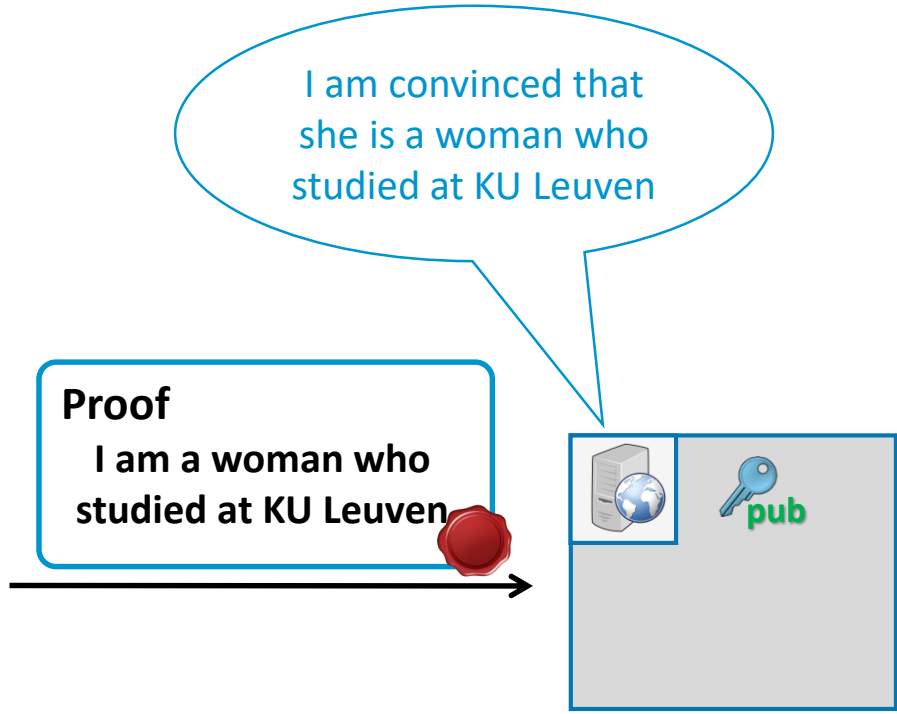
Eventueel kan een vertrouwde partij Alice identificeren (vb. bij misbruik)

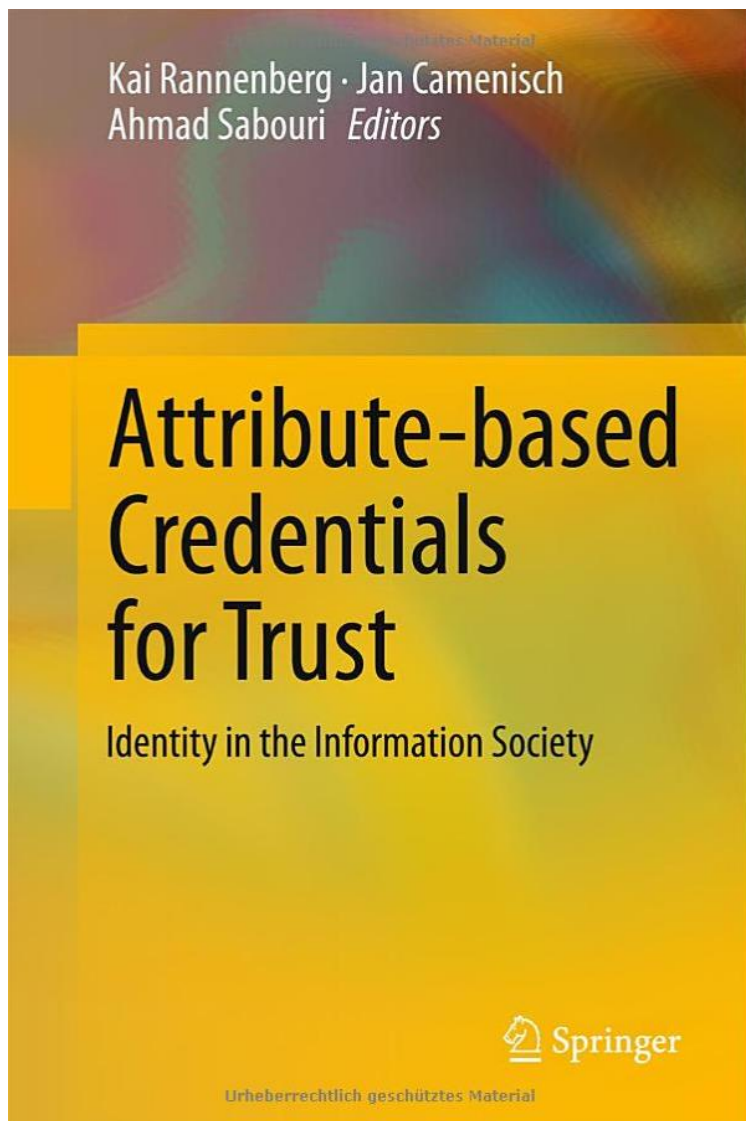




**Identity card**  
Alice De La Rue  
14/04/1984  
Woman  
Arenbergstraat 20  
3001 Heverlee  
Belgium

**Diploma**  
Alice De La Rue  
KU Leuven  
Master in  
Computer Science  
Distinction  
2000





Identity Mixer / Idemix



Microsoft



# Attribute-based credentials

De burger bepaalt selectief welke eigenschappen hij/zij over zichzelf prijsgeeft

Theorie:



Eigen code:



Getest:



Use case:



# Geavanceerde cryptografie

## Speciale cijfertekst

- 1 Threshold encryption
- 2 Format-preserving encryption
- 3 Proxy reencryption

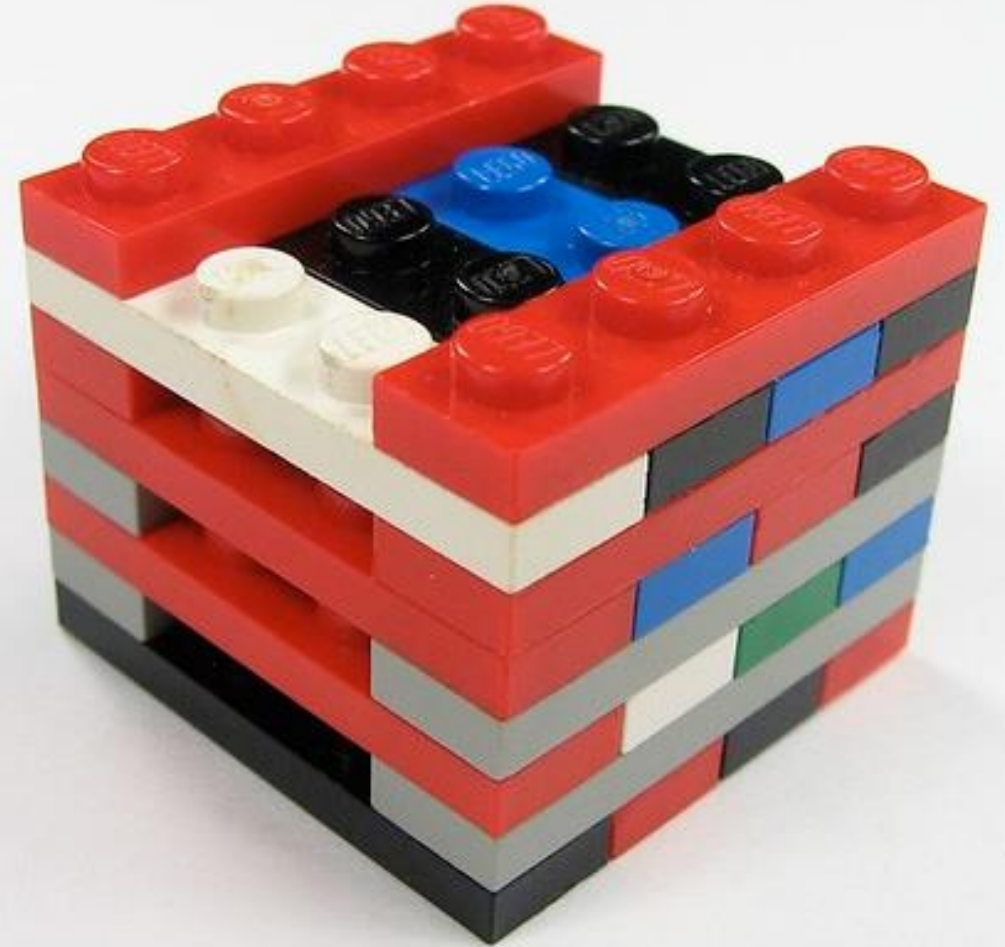
## Authenticatie

- 4 Secure remote password protocol
- 5 Attribute-based credentials

---

## Privacyvriendelijke opvraging

- 6 Secure multiparty computation
- 7 Oblivious transfer
- 8 Private set intersection
- 9 Oblivious join



Er is veel meer!

PAUZE



# Geavanceerde cryptografie

## Speciale cijfertekst

- 1 Threshold encryption
- 2 Format-preserving encryption
- 3 Proxy reencryption

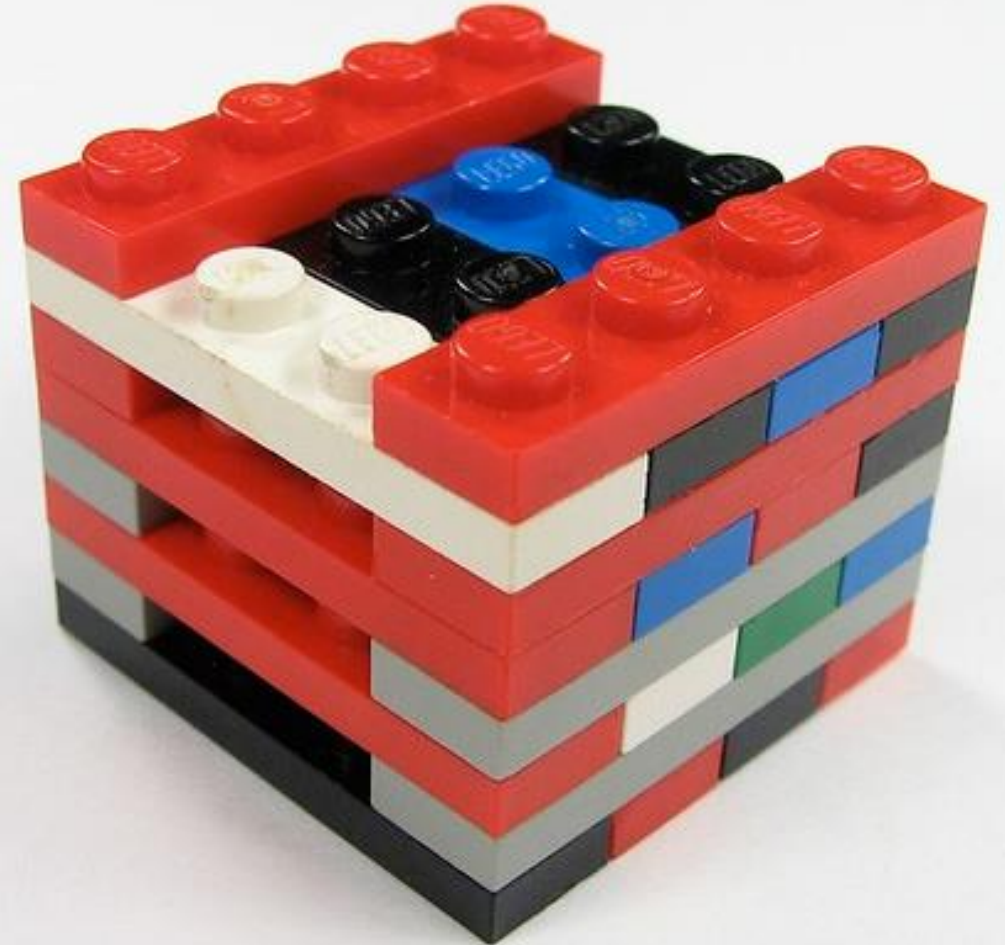
## Authenticatie

- 4 Secure remote password protocol
- 5 Attribute-based credentials

---

## Privacyvriendelijke opvraging

- 6 Secure multiparty computation
- 7 Oblivious transfer
- 8 Private set intersection
- 9 Oblivious join



Er is veel meer!



# Secure multiparty computation

Theorie:



Eigen code:



Getest:



Use case:



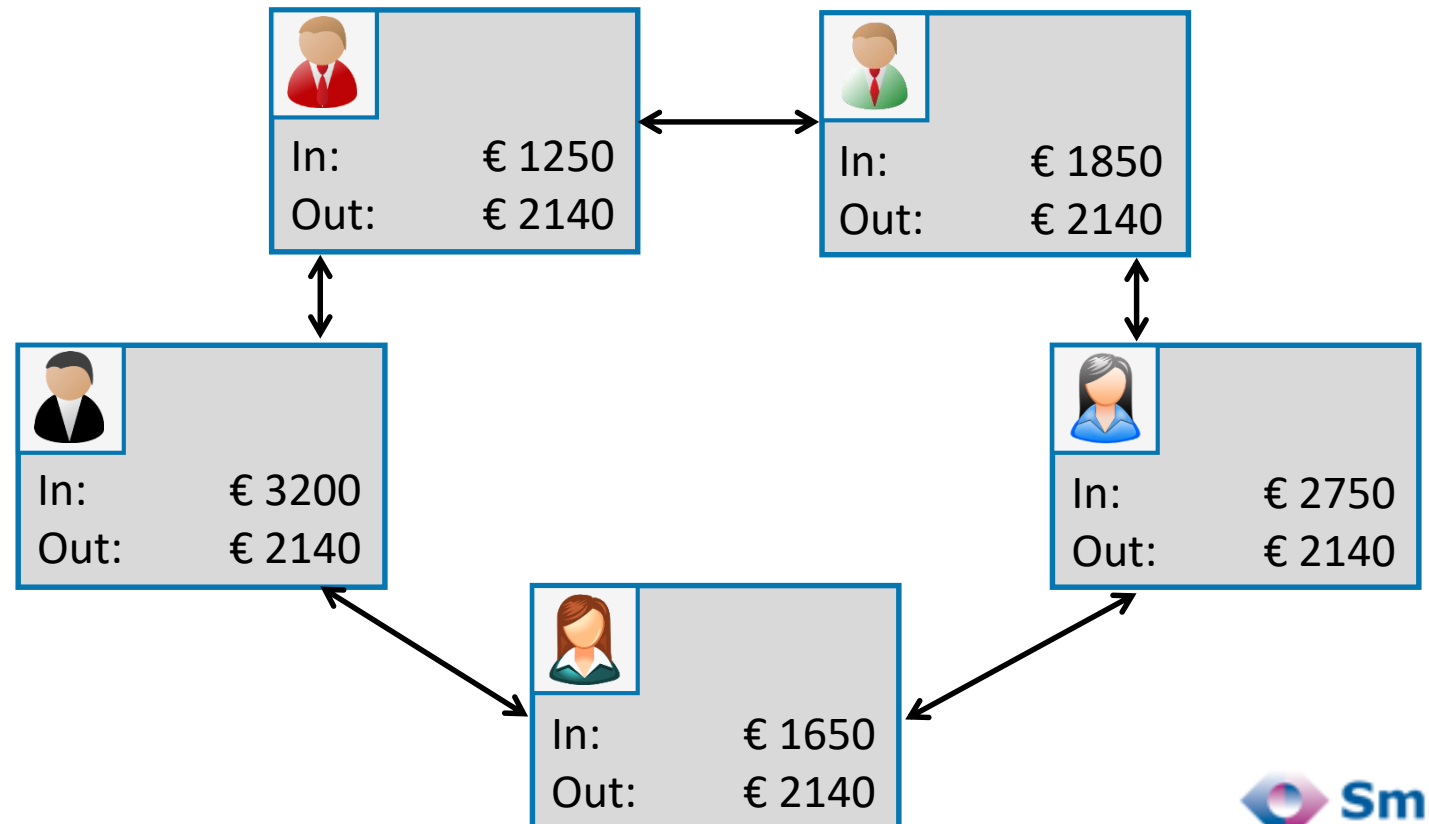


## Probleemstelling

Kunnen we collectief code uitvoeren, dus zonder vertrouwde partij, waarbij meerdere participanten invoer aanleveren die toch confidentieel blijft?

## Voorbeeld

Het berekenen van het gemiddelde loon, zonder het individuele loon prijs te geven.





## PARTISIA

- ▶ 1e grootschalige toepassing en commercieel gebruik van SMC (2009)
- ▶ Context: Deense suikerindustrie
- ▶ Dubbele veiling (kopers en verkopers bieden)
- ▶ Veiling verloopt zonder centrale partij
- ▶ De biedingen blijven voor altijd vertrouwelijk

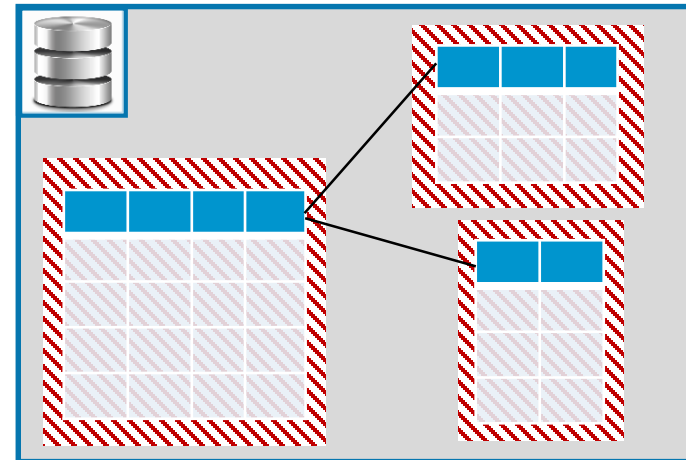
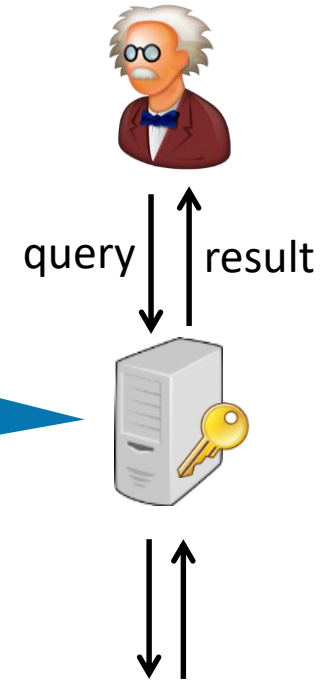


<https://partisia.com/mpc-goes-live/>

# Secure multiparty computation – Private Data as a Service (PDaaS)



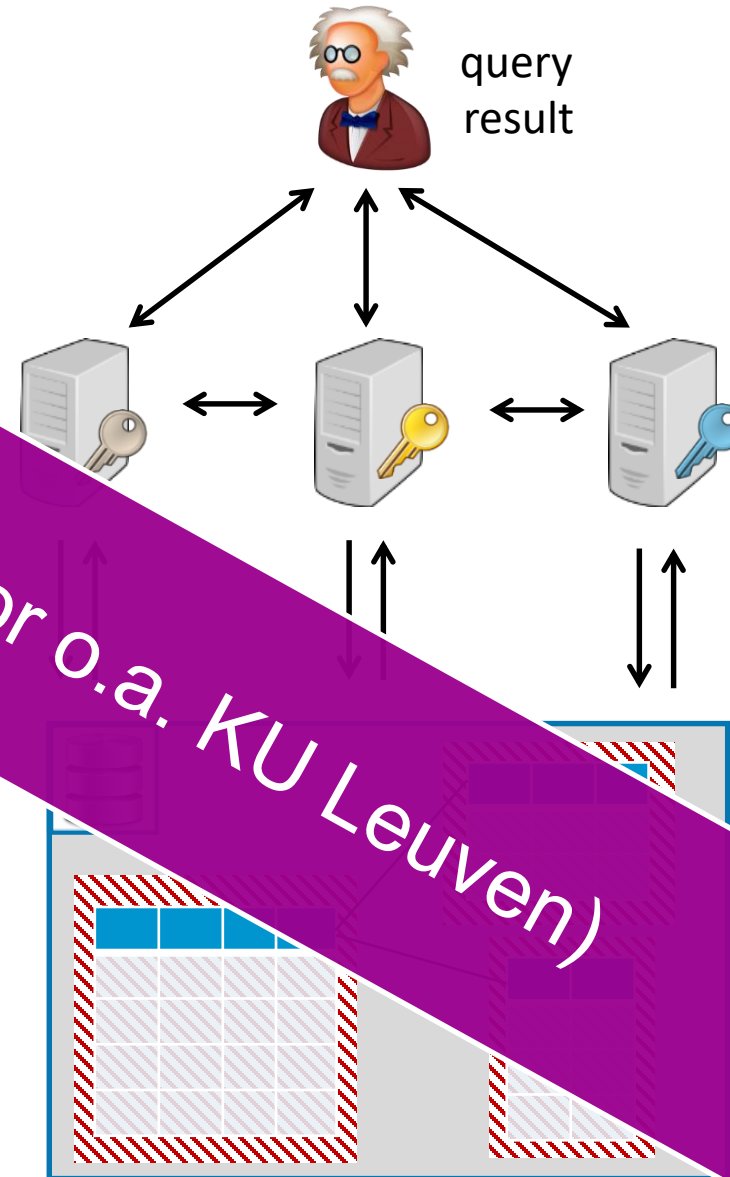
**Vertrouwde partij**  
Kent resultaat  
Heeft in principe toegang tot DB  
(=risico)



<https://eprint.iacr.org/2018/450.pdf>

<https://www.esat.kuleuven.be/cosic/running-projects/?project=242f>

Ongoing research (door o.a. KU Leuven)



- ▶ Vertrouwde partij is verdwenen
- ▶ Inhoud DB en resultaat query blijven verborgen voor servers.

<https://eprint.iacr.org/2018/450.pdf>

<https://www.esat.kuleuven.be/cosic/running-projects/?project=242f>



Alle code die door een centrale vertrouwde partij kan uitgevoerd worden, kan ook zonder die partij uitgevoerd worden, met bescherming van de confidentialiteit van de invoer.

→ **Verregaande theoretische implicaties**

In de praktijk niet altijd haalbaar (o.a. performantie)

Specifiek uitgedachte protocols doorgaans efficiënter

# Secure multiparty computation

Het collectief uitvoeren van code  
waarbij de invoer van de verschillende  
partijen confidencieel blijft

Theorie:



Eigen code:



Getest:



Use case:





# Oblivious transfer (vergeetachtige verzending)

Theorie:



Eigen code:



Getest:

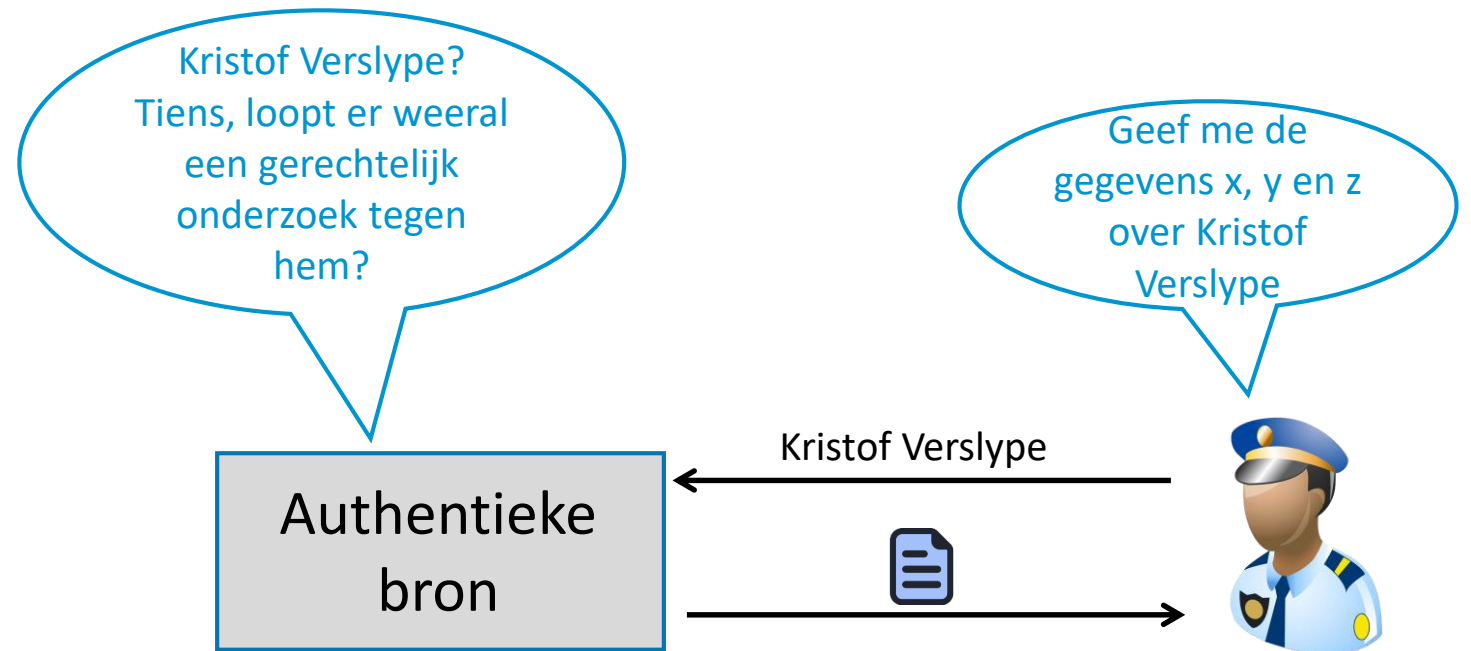


Use case:



## Scenario

Onderzoek vanuit justitie naar een specifieke burger



**Er wordt voldaan aan de informatievereiste,  
ten koste van privacy burger & confidentialiteit van het onderzoek**

## Scenario

Onderzoek vanuit justitie naar een specifieke burger

Die slaan door ginder!

Geef me de gegevens x, y en z  
over alle burgers geboren tussen  
1980 en 1990

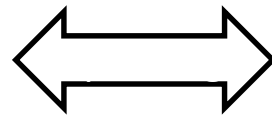
Authentieke  
bron

[1980, 1990]



**Disproportionele verwerking persoonsgegevens door justitie**

**Privacy burger &  
confidentialiteit onderzoek  
t.a.v. authentieke bron**



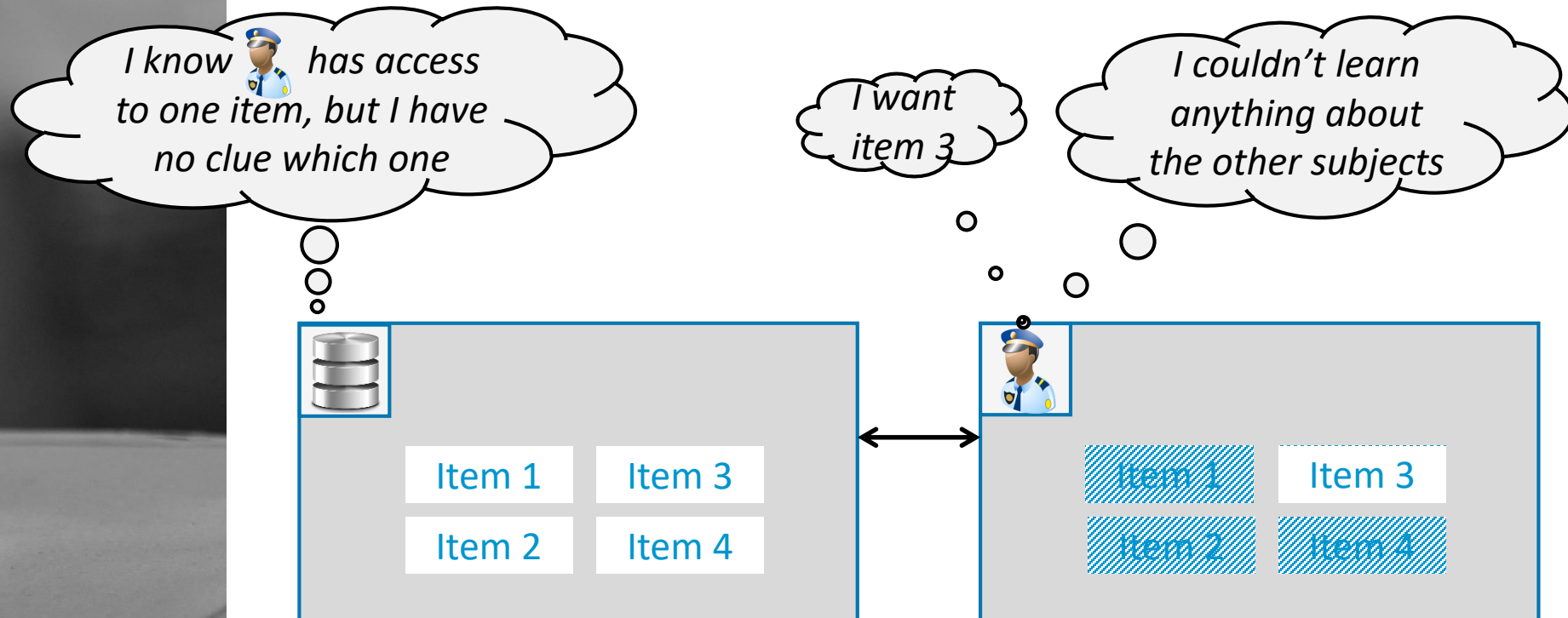
**Proportionele  
gegevensverwerking  
door justitie**




# Oblivious transfer - Concept

Een zender geeft aan een ontvanger toegang tot exact één data-item uit een verzameling data-items.

De zender weet echter niet over welk data item het gaat.



# Oblivious transfer – Scenario 1

I know  has access to one odd item, but I have no clue which one

I want info about 67.3.23-940.89  
It's item 3

**Group 1: odd SSNs**

Item 1 27.10.20-443.79	Item 2 51.07.11-127.81
Item 3 67.03.23-940.89	Item 4 97.01.10-729.45


**Group 2: even SSNs**

Item 5 62.01.25-81.90	Item 6 62.04.05-156.54
Item 7 73.02.23-520.46	Item 8 81.06.21-267.70

Odd

i	SSN
1	27.10.20-443.79
2	51.07.11-127.81
3	67.03.23-940.89
4	97.01.10-729.45

OT



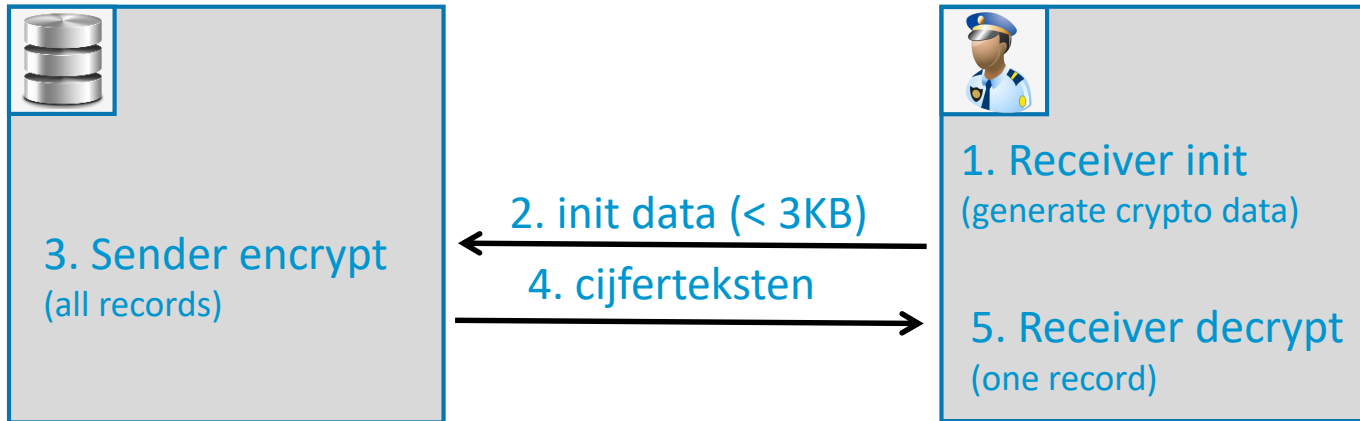
I couldn't learn anything about the other subjects

Item 1 27.10.20-443.79	Item 2 51.07.11-127.81
Item 3 67.03.23-940.89	Item 4 97.01.10-729.45

**In realiteit (1)**  
Wellicht in combinatie met juridische verplichting irrelevante cijferteksten te verwijderen

**In realiteit (2)**

- ▶ Niet 1-uit-4 oblivious transfer,
- ▶ Eerder 1-uit-1000 oblivious transfer



Zwaartepunt ligt bij zender

## 256 bit security, standaard laptop

	1 uit 1000 records			1 uit 10 000 records			1 uit 25 000 records		
	Receiver init $O(\log n)$	Sender encrypt $O(\log n * s)$	Receiver decrypt $O(s + \log n)$	Receiver init $O(\log n)$	Sender encrypt $O(\log n * s)$	Receiver decrypt $O(s + \log n)$	Receiver init $O(\log n)$	Sender encrypt $O(\log n * s)$	Receiver decrypt $O(s + \log n)$
20,7KB/rec.	213ms	925ms	30ms	299ms	4896ms	27ms	289ms	11544ms	37ms
103,4KB/rec	214ms	2604ms	33ms	270ms	22 146ms	36ms	282ms	54586ms	40ms

### Setup

Data in-memory / Lenovo Thinkpad L570, Windows 10, Intel Core i5-6300 CPU @ 2,40Ghz, 16GB

Measurements only of crypto calculations, not of storage IO or communication

P-521 curve, Average of 10 runs is taken

On average, less than 40% of total CPU used by the program

## Academische wereld: Actief onderzoeksdomein

- 1-out-of-2
- 1-out-of-N
- M-out-of-N
- Traditioneel of kwantumresistent
- ...

## Implementatie door Smals Research Library voor testdoeleinden

1-out-of-N oblivious transfer [1]	Unit test
1-out-of-2 oblivious transfer [1]	Unit test
Dual Mode Encryption [2]	Unit test



[1] M Byali, A Patra, D Ravi, P Sarkar. *Fast and Universally-Composable Oblivious Transfer and Commitment Scheme with Adaptive Security*. IACR Cryptology ePrint Archive, 2017

[2] C. Peikert, V. Vaikuntanathan, B. Waters. *A Framework for Efficient and Composable Oblivious Transfer*. CRYPTO 2008: Advances in Cryptology – CRYPTO 2008 pp 554-571

# Oblivious transfer

(vergeetachtige verzending)

Het opvragen aan een zender van één record uit een verzameling records, zonder dat de zender weet hetwelke

Theorie:



Eigen code:



Getest:



Use case:





# Private set intersection

Theorie:



Eigen code:



Getest:



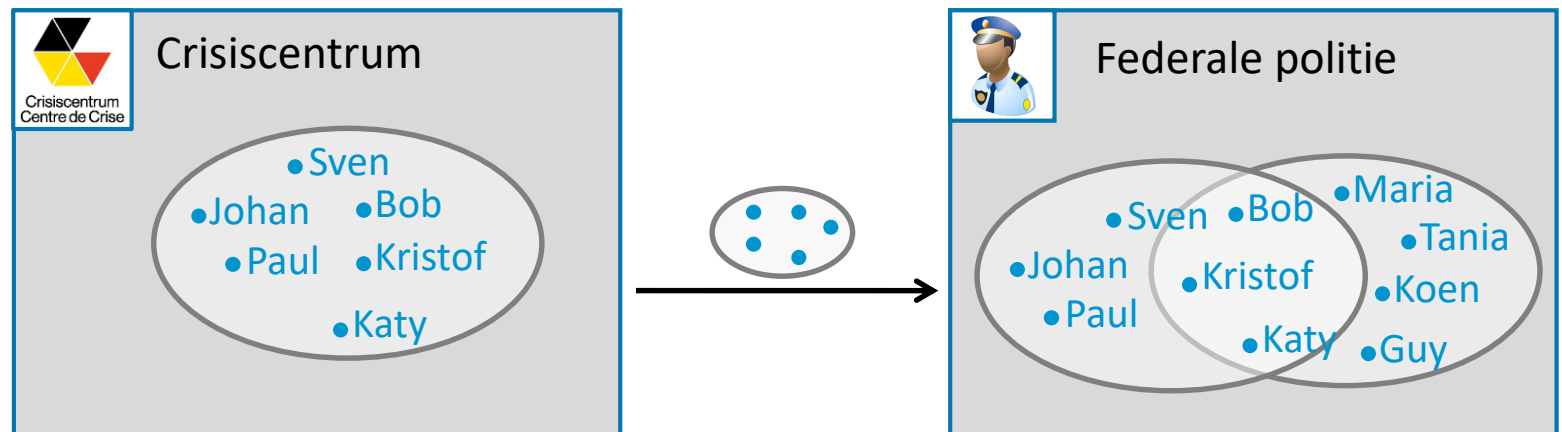
Use case:



# Private set intersection (PSI)

## Toepassing

Wetshandhavingdiensten willen weten of ze gemeenschappelijke verdachten volgen. (anders geformuleerd: over welke personen hebben ze beiden een dossier?)



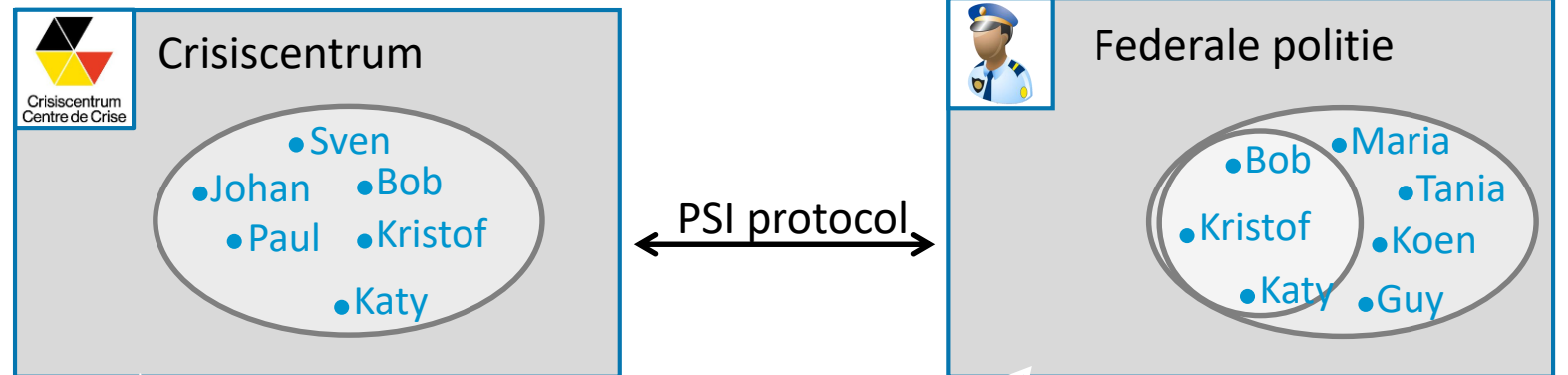
## Probleemstelling

Crisiscentrum geeft aan Federale Politie de volledige lijst van individuen met dossier  
-> De ene partij geeft veel te veel gevoelige persoonsgegevens aan de andere partij

# Private set intersection (PSI)

## Toepassing

Wetshandhavingdiensten willen weten of ze gemeenschappelijke verdachten volgen. (anders geformuleerd: over welke personen hebben ze beiden een dossier?)



Leert enkel dat er protocoluitvoering was met federale politie

- Komt enkel te weten voor welke individuen beide organisaties een dossier hebben
- Komt niet te weten over welke andere personen het crisiscentrum een dossier heeft

## **Andere potentiële toepassingen**

- ▶ Welke bedrijven ook achterstallige bijdragen bij andere overheidsinstelling
- ▶ Welke burgers aangesloten bij meerdere ziekenfondsen (dubbel verzekerd)



## Resultaten uit de literatuur (\*)

- ▶ 128 bit security
- ▶ Elementen 32 bits lang (voldoende voor rijksregisternummers)
- ▶ 2 PCs (Intel Haswell i7-4770K CPU with 3.5 GHz and 16 GB RAM)

### Gelijke set grootte $2^{20}$ (1 000 000)

Runtime	5,6 sec.
Communicatie	107 MB

### Ongelijke set grootte $2^{24}$ (16 000 000) en $2^{12}$ (4000)

Runtime	35,1 sec.
Communicatie	362 MB

#### Opmerking:

Meerdere PSI protocols werden voorgesteld. Bovenstaande is een recente en efficiënte.

(\*) Pinkas, B., Schneider, T., & Zohner, M. (2016). Scalable private set intersection based on OT extension. *ACM Transactions on Privacy and Security (TOPS)*, 21(2), 7.

Twee entiteiten hebben elke een set met elementen.

Een van die entiteiten leert de doorsnede.

Voor de rest lekt er geen informatie

## Private set intersection

Theorie:



Eigen code:

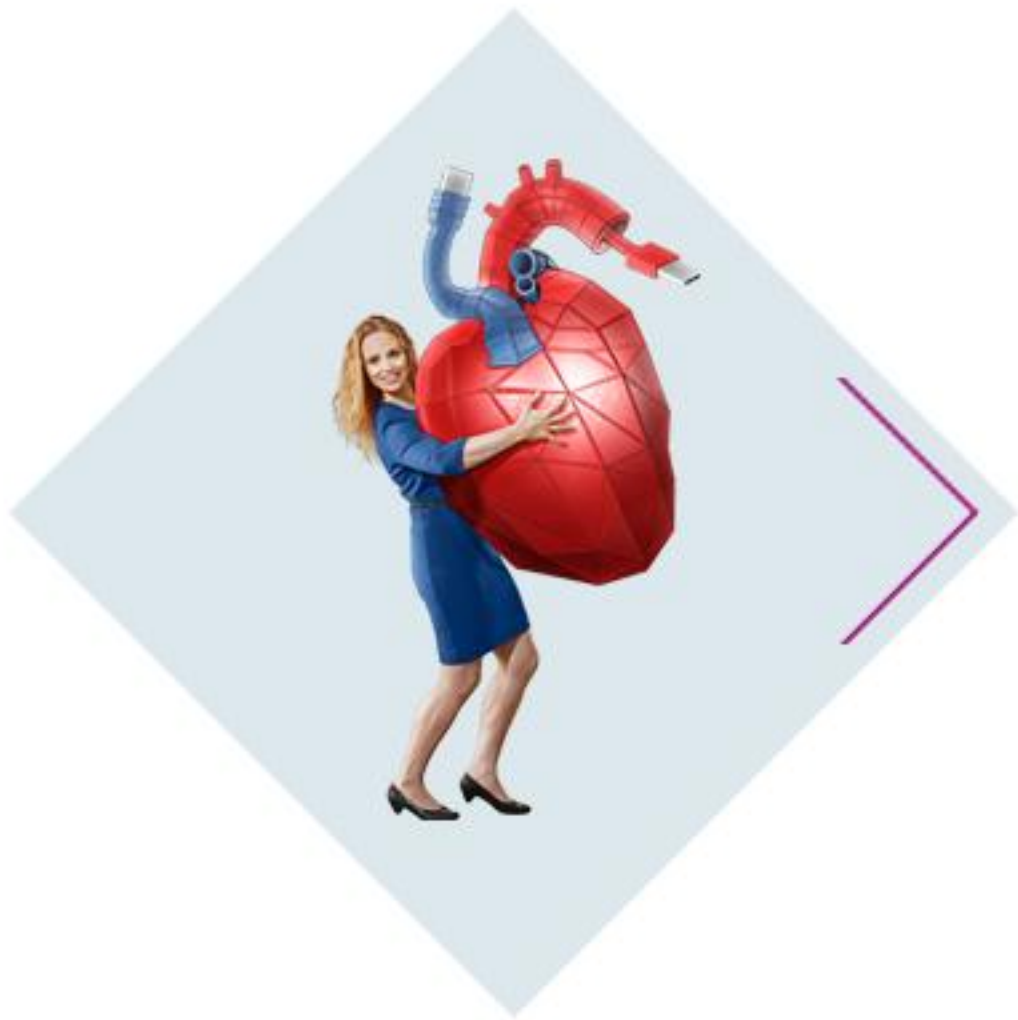


Getest:



Use case:





# Oblivious join

Concept door Smals Research

Theorie:



Eigen code:



Getest:



Use case:



## Context

Een onderzoeker wil voor wetenschappelijk onderzoek toegang tot bepaalde (gepseudonimizeerde) persoonsgegevens

## Welke burgers?

- ▶ Alle zelfstandigen in bijberoep
- ▶ Die meer dan € 50 000 per jaar verdienen in hoofdberoep

## Welke data?

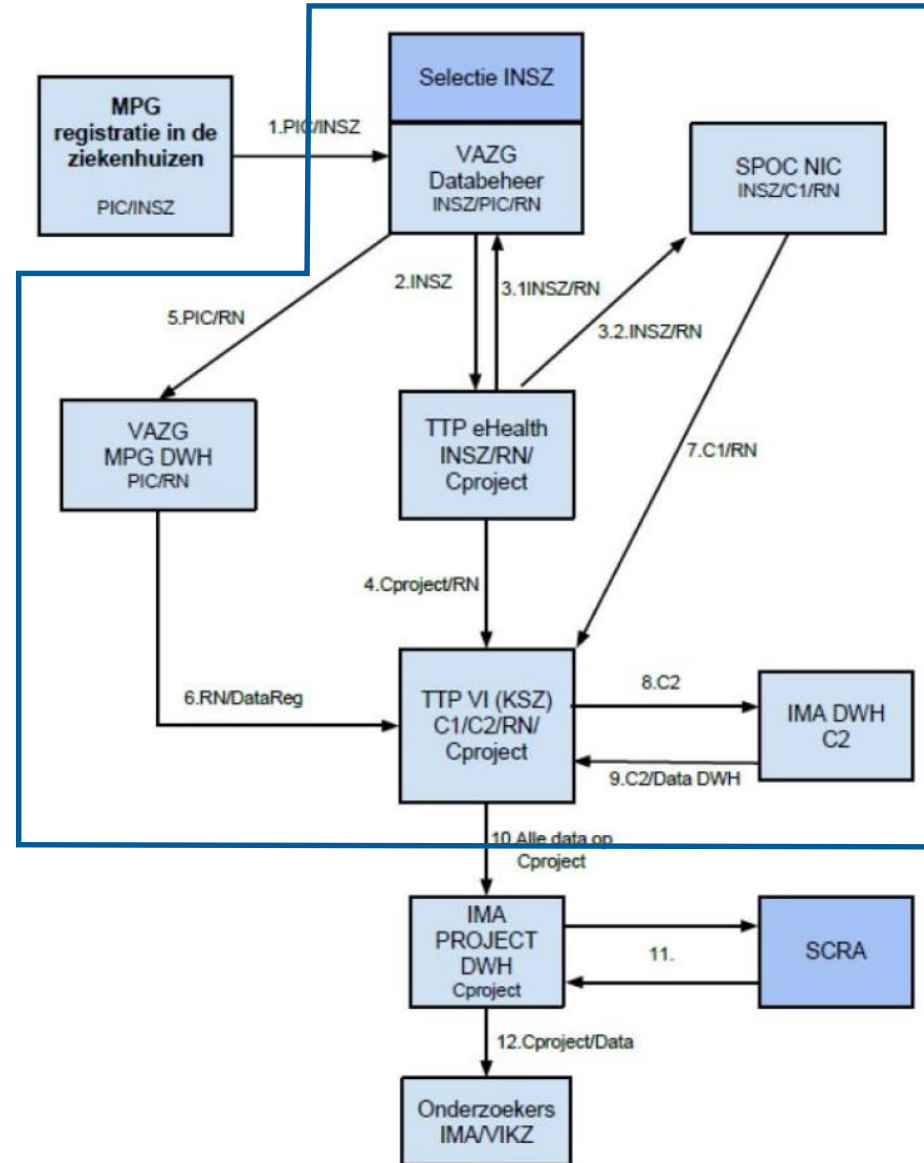
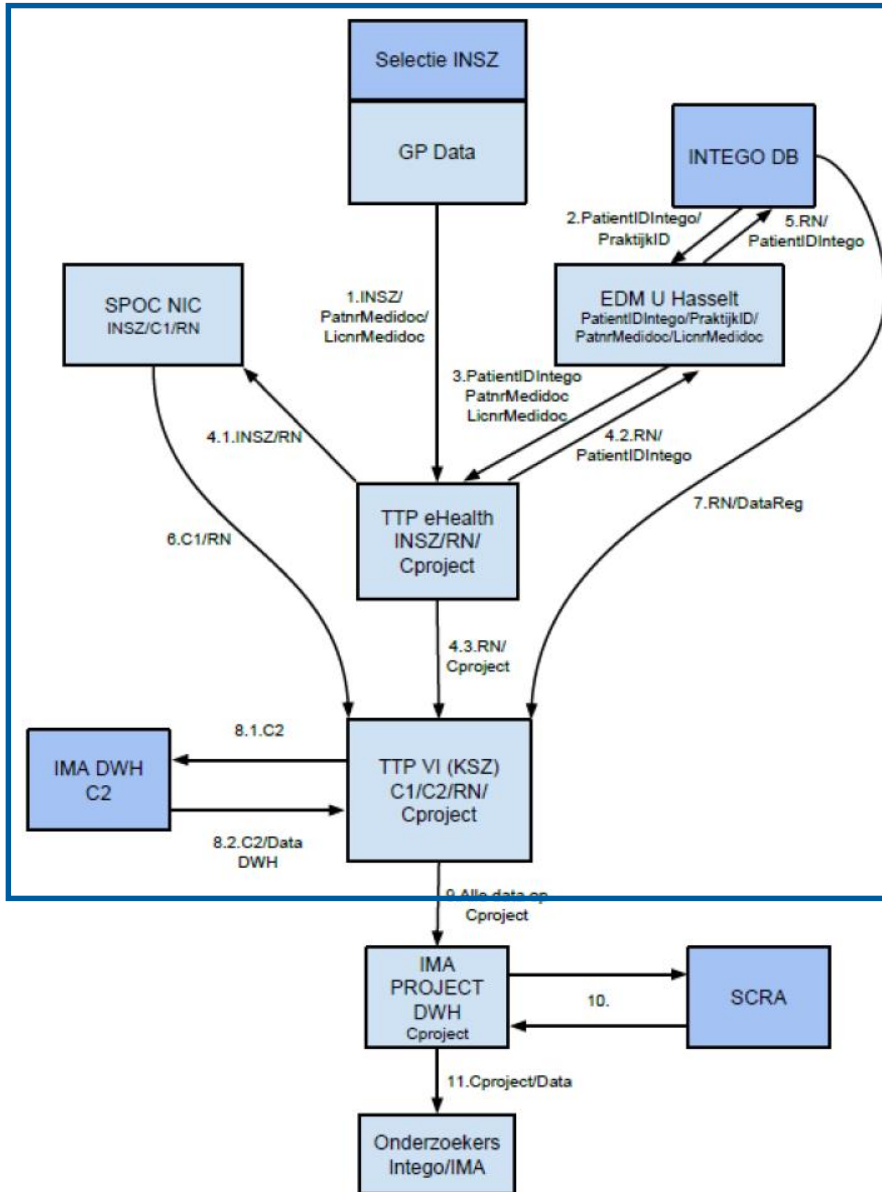
- ▶ Specifieke medische gegevens &
- ▶ Gegevens over verzekering als zelfstandige

## Medewerking van verschillende bronnen vereist



Voor onderzoekdoeleinden kruisen en pseudonimiseren van persoonsgegevens afkomstig van verschillende bronnen

# Oblivious join



Beraadslaging Nr. 17/071 van 19/9/17 (links)

Beraadslaging Nr. 19/062 van 2/4/19 (rechts)

- ▶ Complexe flow
- ▶ Op maat
- ▶ Traag
- ▶ Data lekkage

## Context

Kruisen en pseudonimiseren van persoonsgegevens voor wetenschappelijk onderzoek

## Scenario

### Welke burgers?

- ▶ Alle zelfstandigen in bijberoep
- ▶ die meer dan € 50 000 per jaar verdienen in hoofdberoep

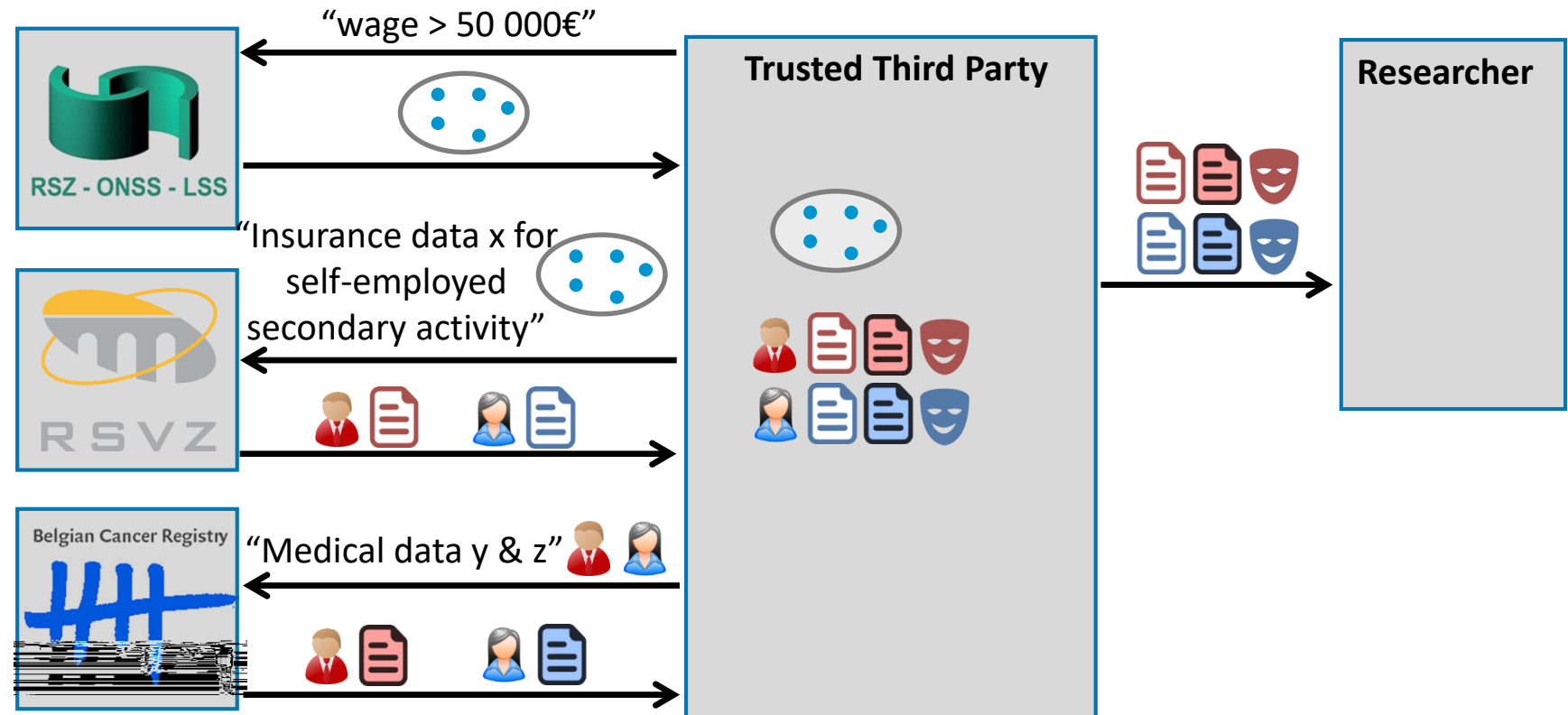
### Welke data?

- ▶ Specifieke medische gegevens
- ▶ gegevens over verzekering als zelfstandige

## Issues

- ▶ Bronnen en/of TTP komen te veel persoonsgegevens te weten
- ▶ Extra intermediären verhogen complexiteit

## Een mogelijke traditionele benadering



# Oblivious join

## Context

Kruisen en pseudonimiseren van persoonsgegevens voor wetenschappelijk onderzoek

## Objectieven

- ▶ Gestandaardiseerde aanpak
- ▶ Efficiënt
- ▶ Geen data-lekken

## Filosofieën

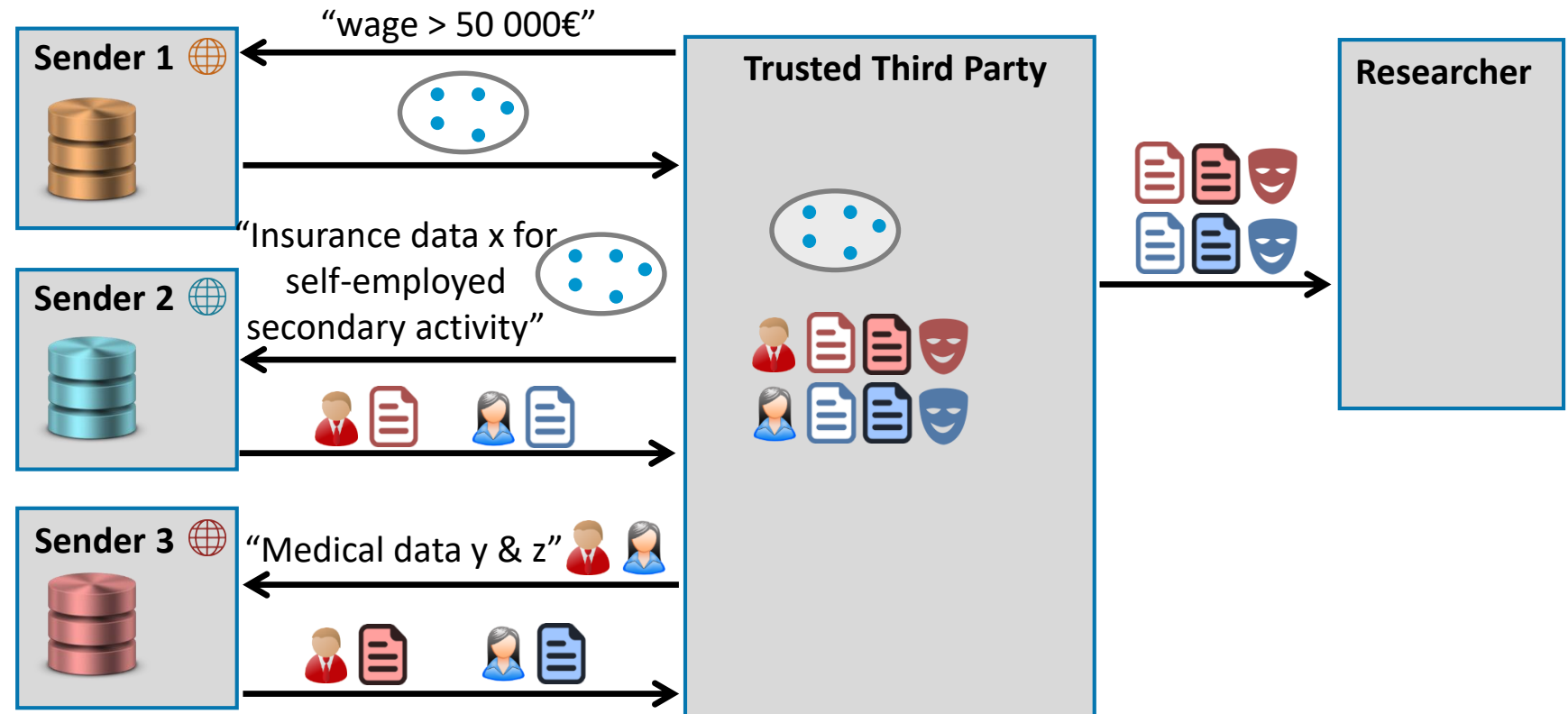
### Gecentraliseerd

Alle data in centraal  
warehouse  
Data archipel

### Gedecentraliseerd

Data blijft in bron  
Oblivious join

## Een mogelijke traditionele benadering



# Oblivious join

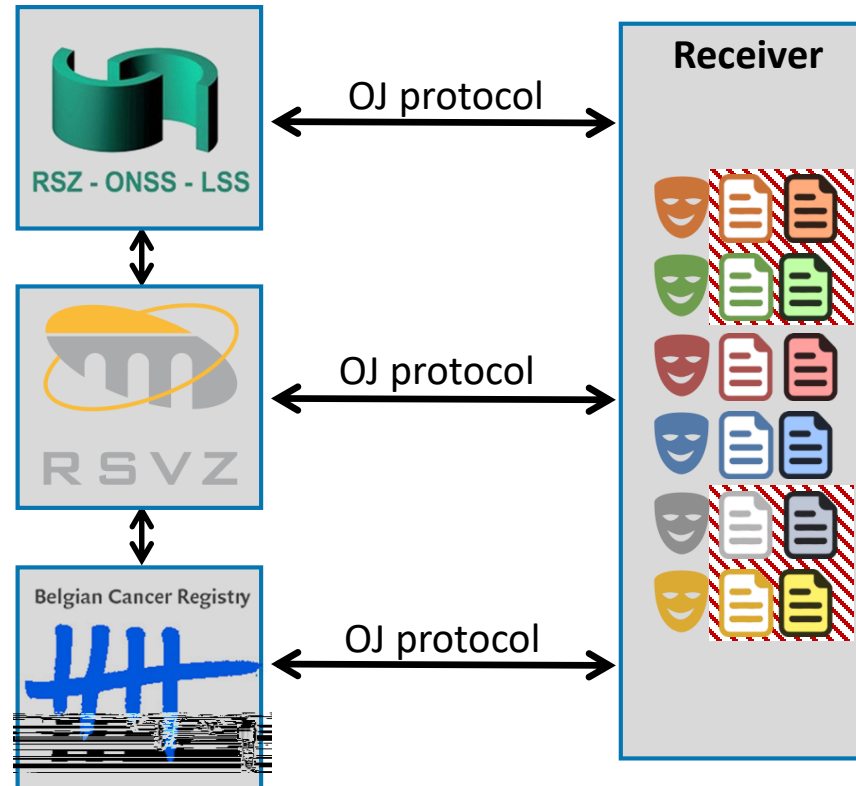
## Idee

Elke zender verstuurt alle potentieel relevante data geëncrypteerd & gepseudonimiseerd  
ontvanger kan enkel decrypteren indien voor dezelfde burger iets ontvangen van elke bron

## Assumpties

- Query (beraadslaging) is publiek
- Gedeelde identifier

Er lekken noch statistische noch persoonsgegevens naar de zenders



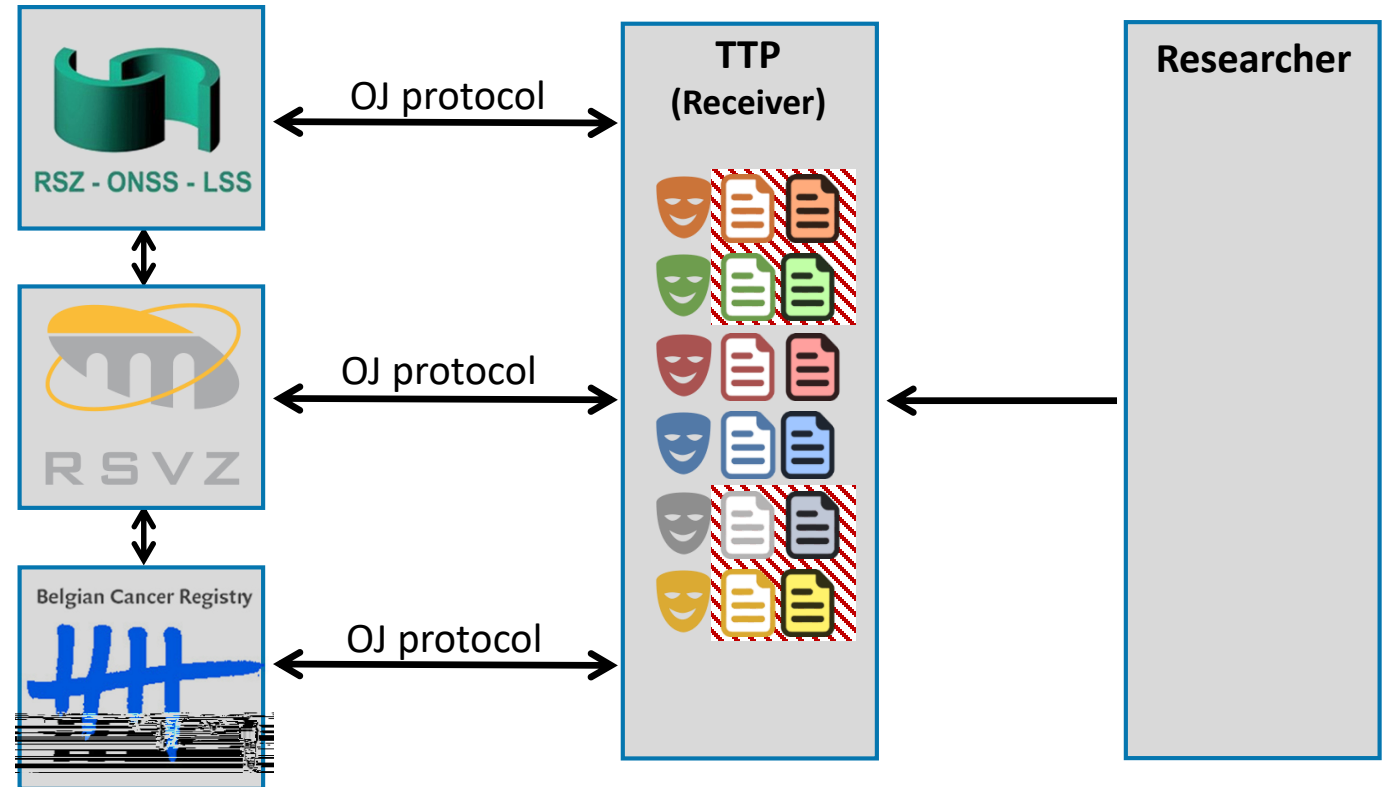
Receiver komt enkel de minimum noodzakelijke gepseudonimiseerde **persoonsgegevens** te weten.

Minimale hoeveelheid **statistische data** lekt naar receiver  
(in vb.: #burgers met loon > € 50 000 en #zelfstandigen in bijberoep)

# Oblivious join

## TTP (Trusted third party)

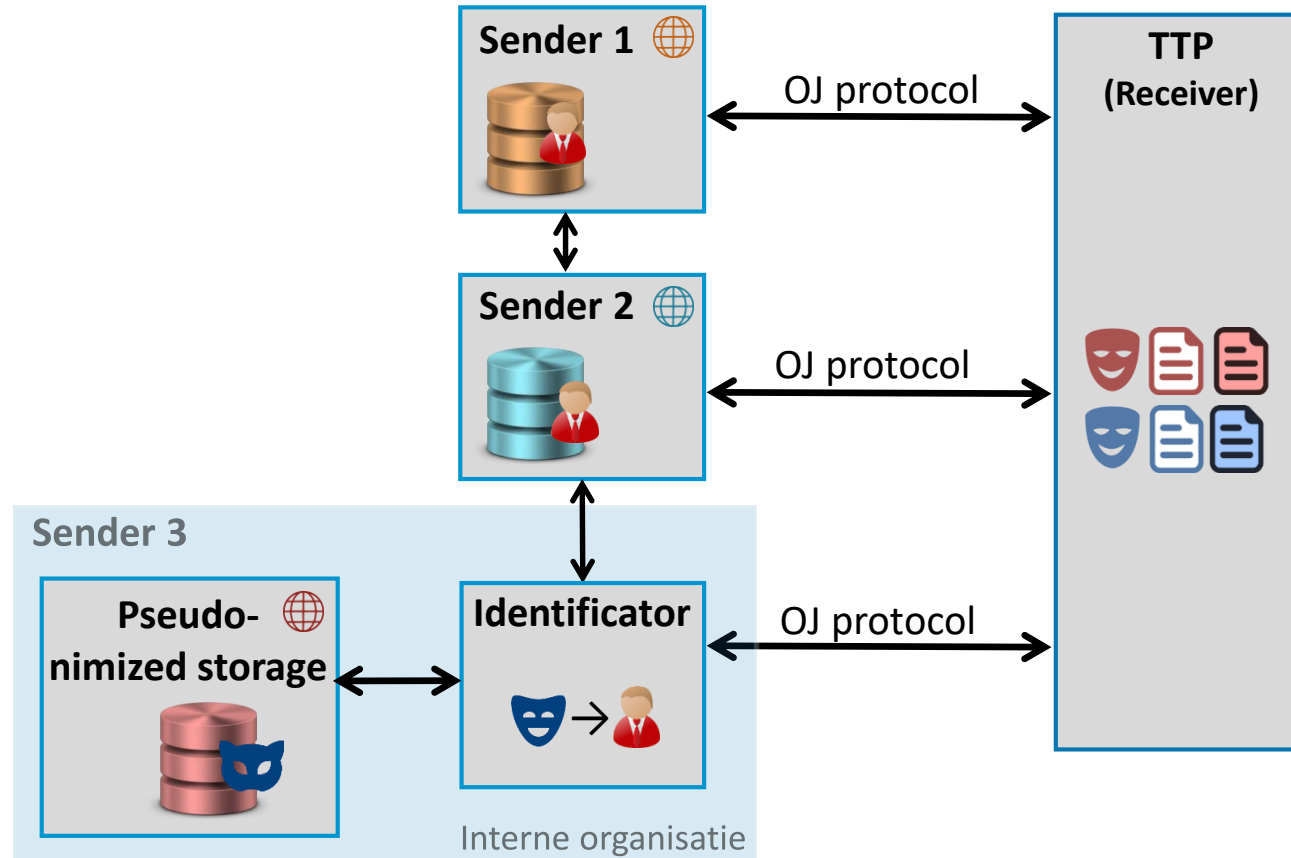
- ▶ Komt enkel minimaal noodzakelijke data te weten
- ▶ Verwijdert asap irrelevante cijferteksten
- ▶ Doet eventueel bijkomende operaties (vb. controles)
- ▶ Doet toegangscontrole op onderzoeker
- ▶ 'Trust' zeer beperkt



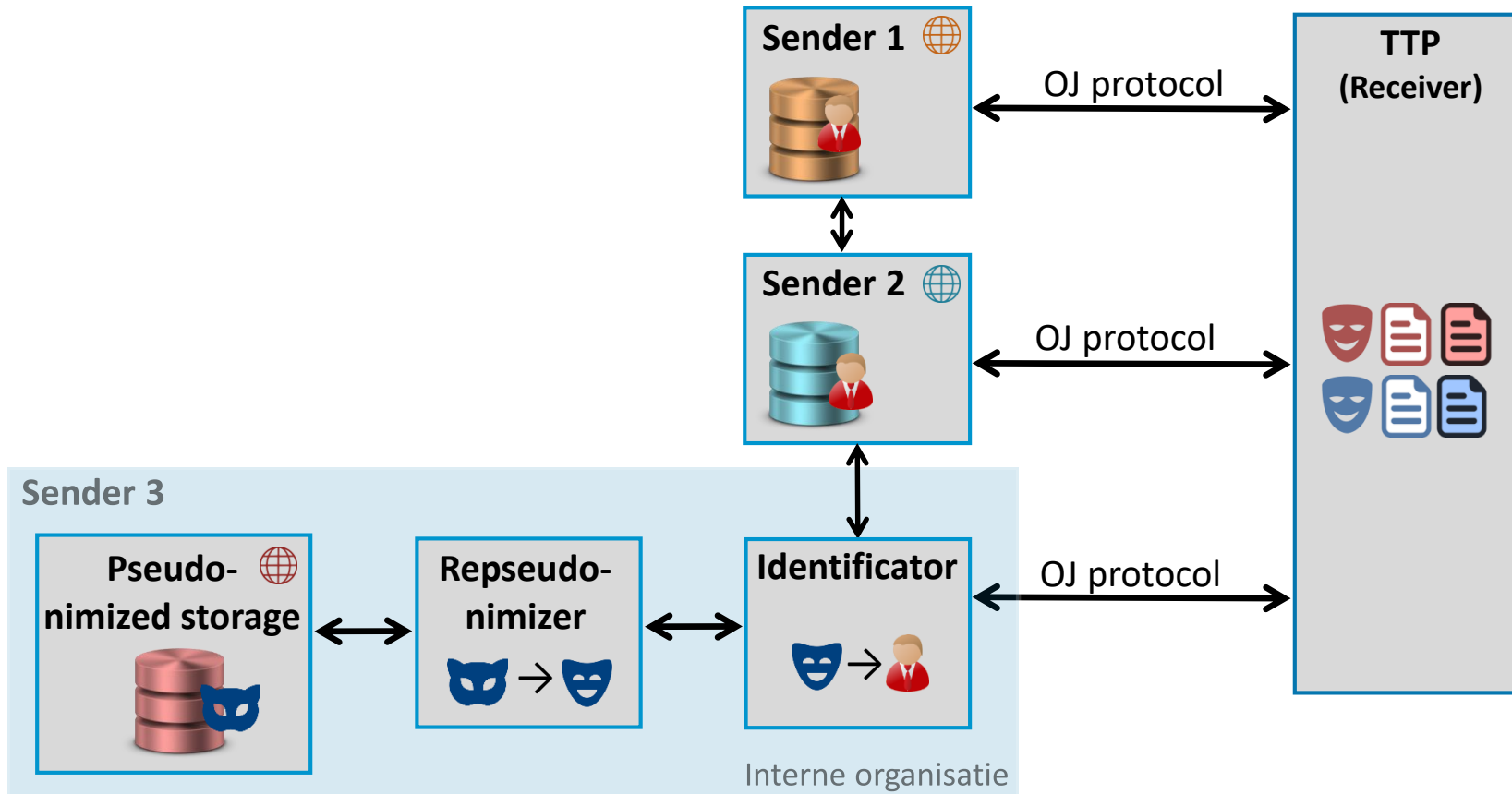
# Oblivious join

**Pseudonymized storage service**  
kent data maar kan ze niet linken  
aan een natuurlijk persoon

**Identificator**  
kent bijhorende identifiers,  
maar niet de data

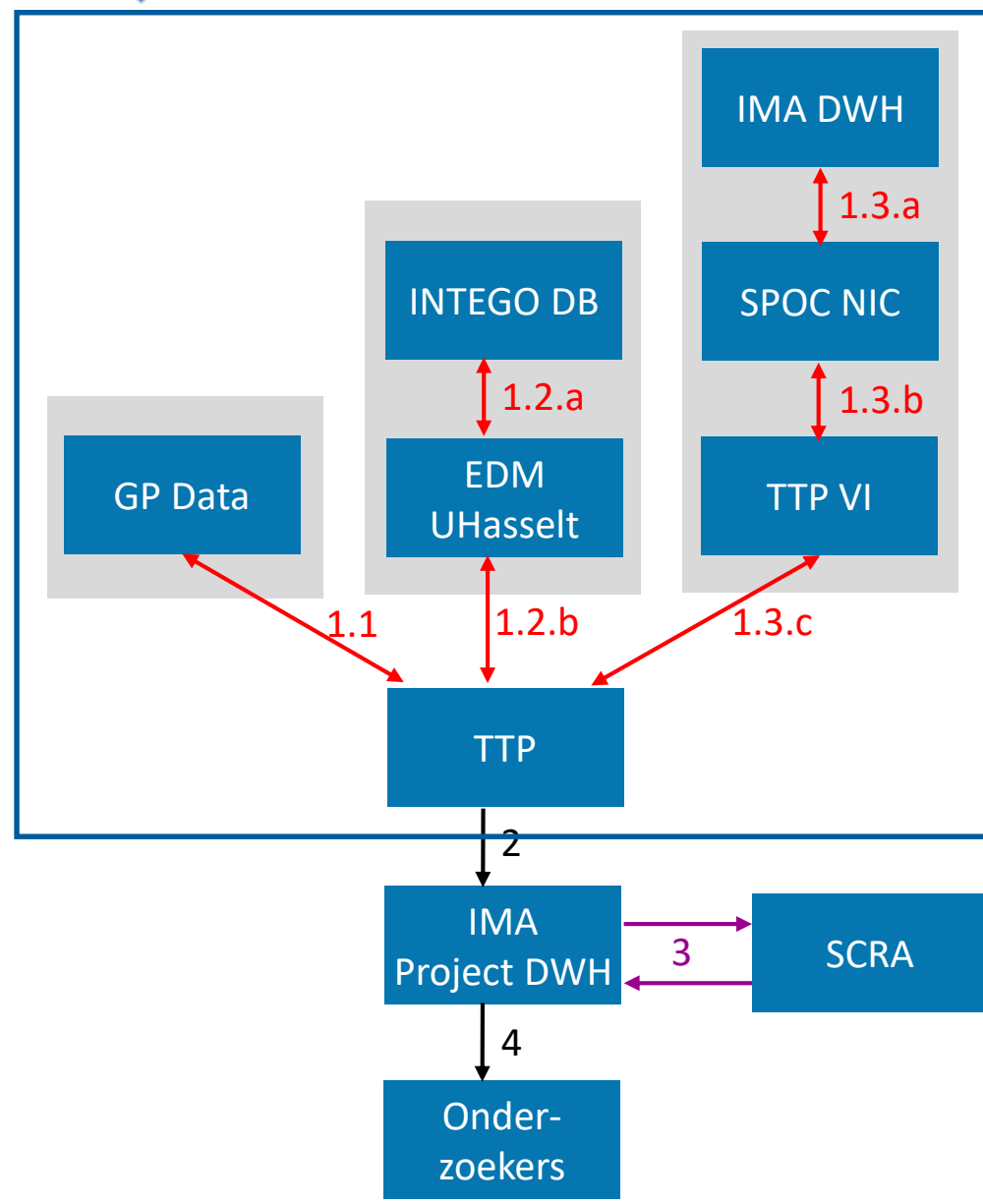
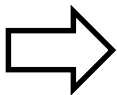
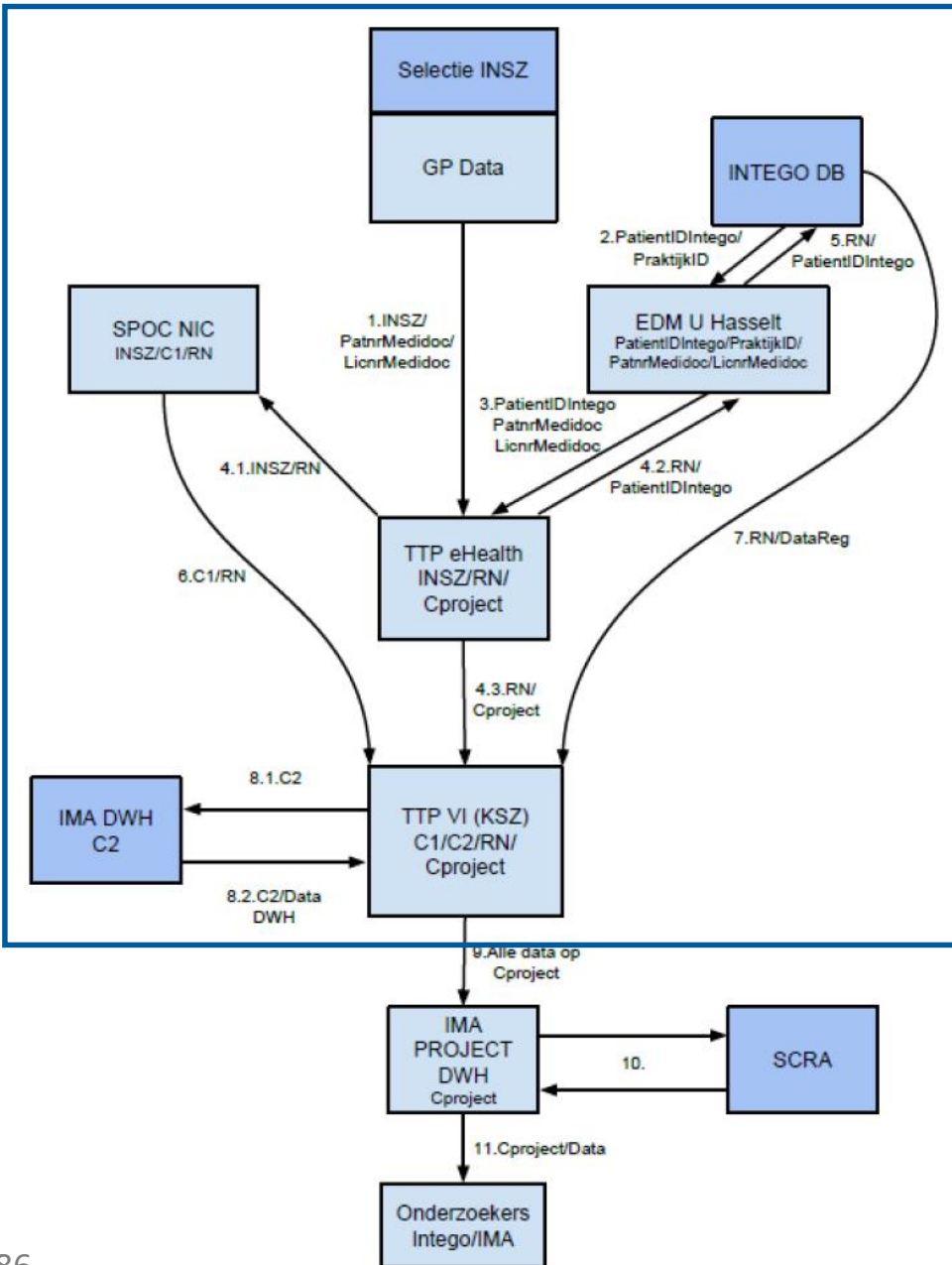


# Oblivious join



# Oblivious join

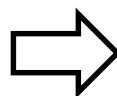
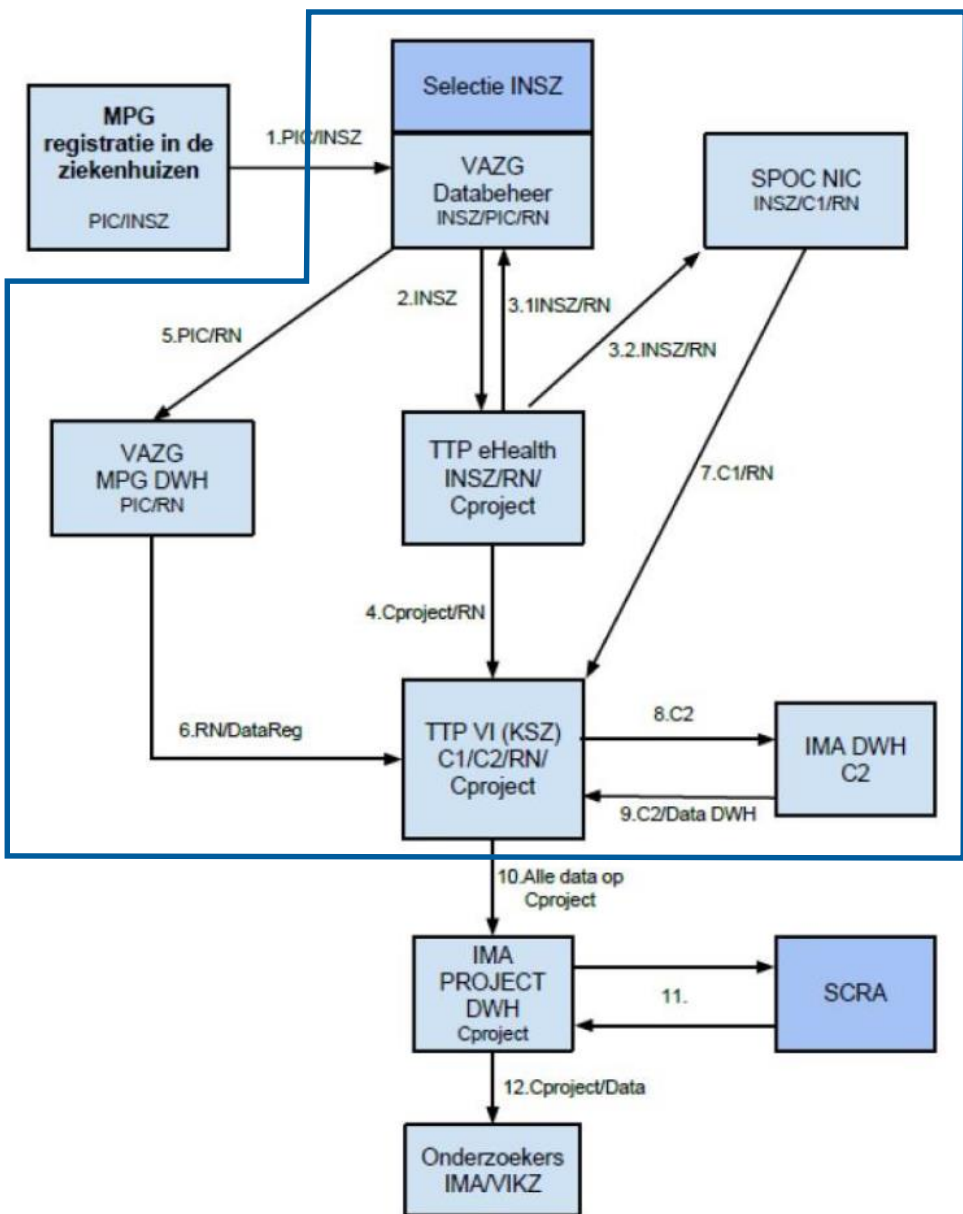
Beraadslaging Nr. 17/071 van 19 september 2017



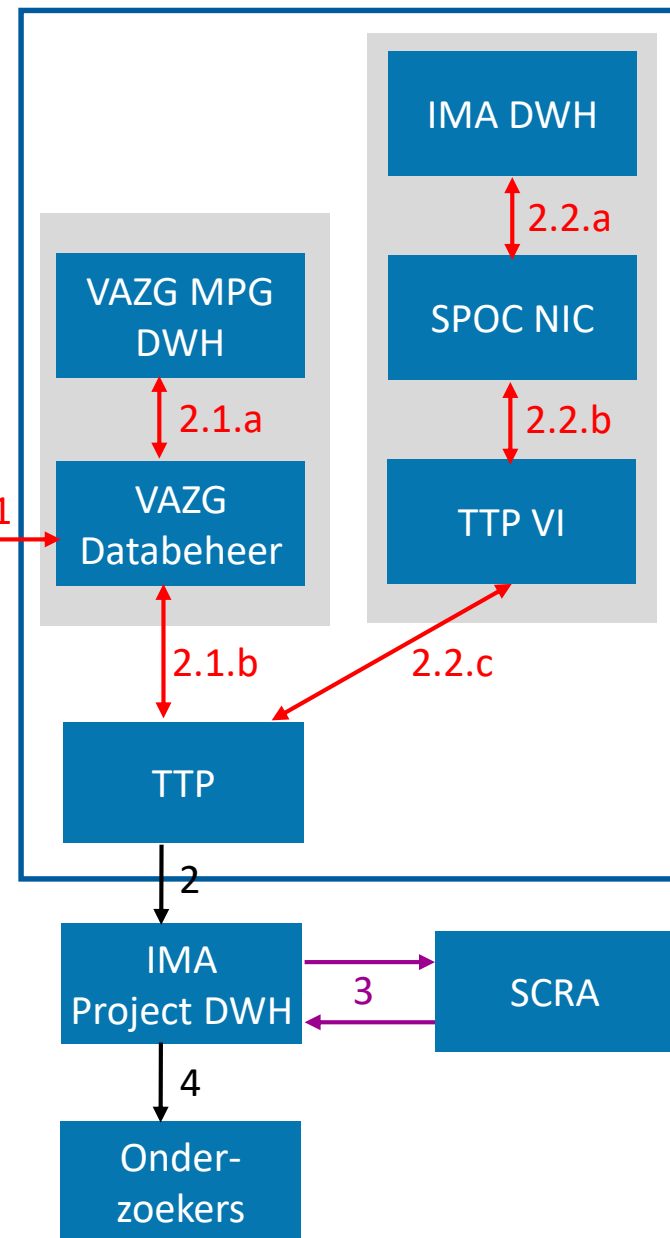
- Oblivious join geautomatiseerd
- Analyse
- Data transfer

Standaard flow in parallel  
Geen data lekken

# Oblivious join



Ziekenhuizen

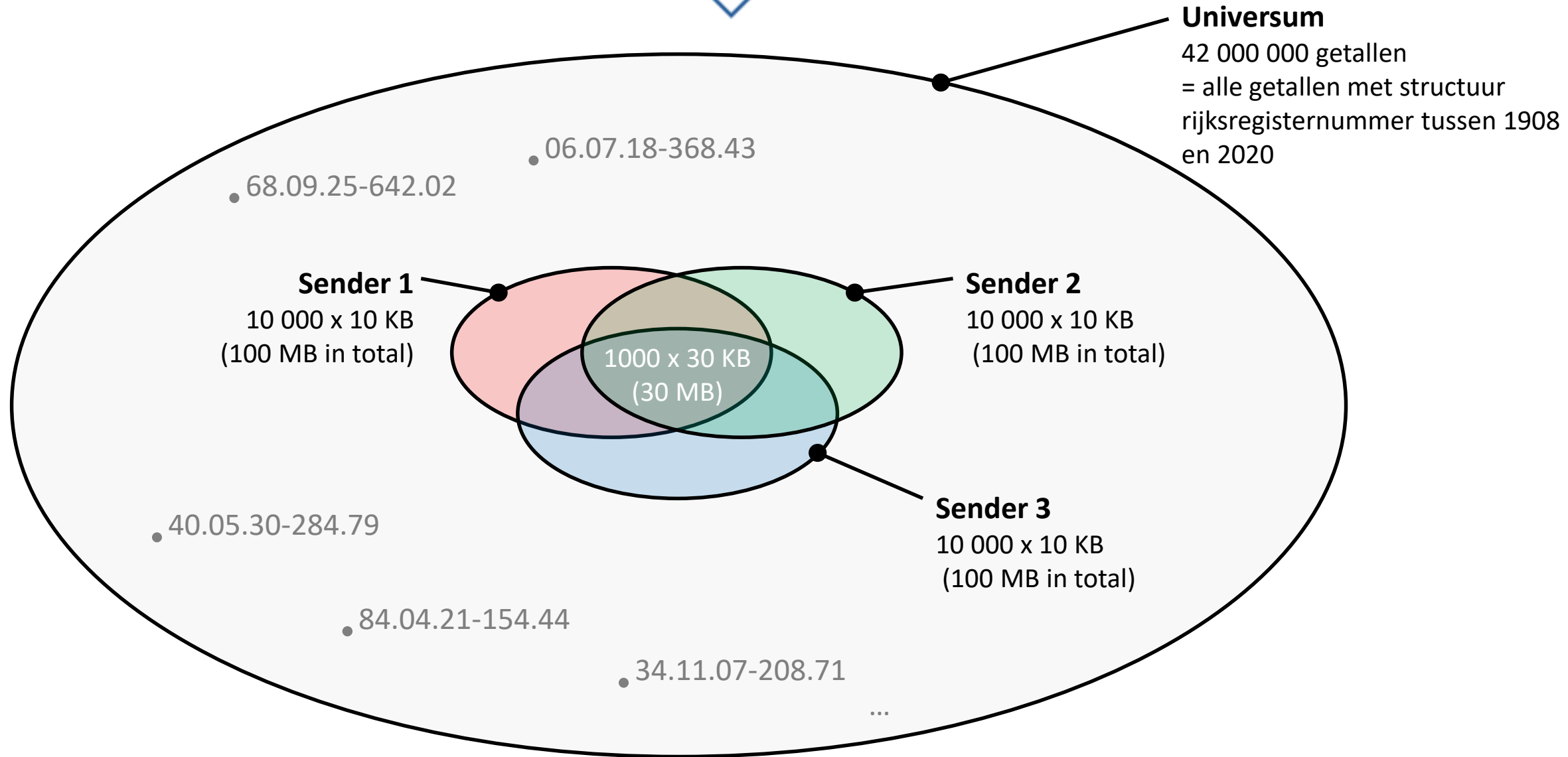


Beraadslaging Nr. 19/062 van 2 april 2019

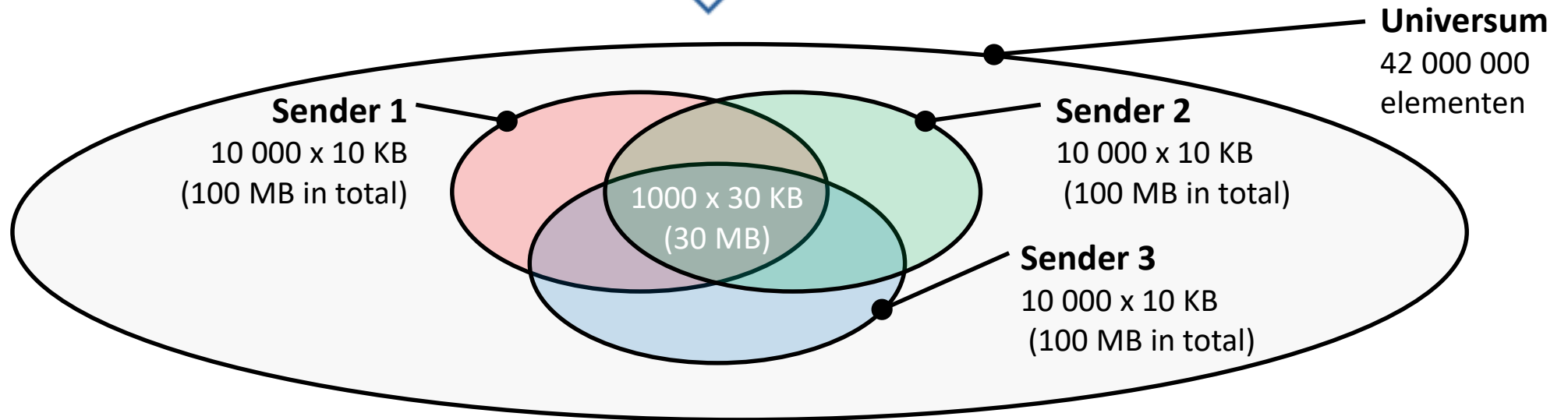
- Oblivious join geautomatiseerd
- Analyse
- Data transfer

Standaard flow in parallel  
Geen data lekken

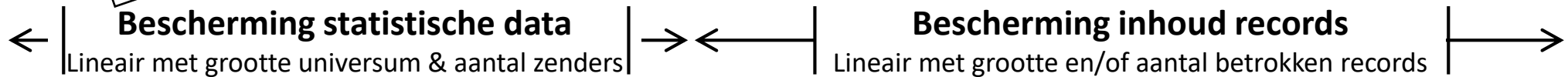
# Oblivious join - Performantie



# Oblivious join - Performantie



	Sender			Receiver	
	Offline precalculation	Pseudonym agreement	Offline encryption	Import	Offline decryption
128 bits	40 min	46 min	2,4 sec.	13 sec.	1,3 sec.
256 bits	<3u	2u	2,5 sec.	73 sec.	0,7 sec.



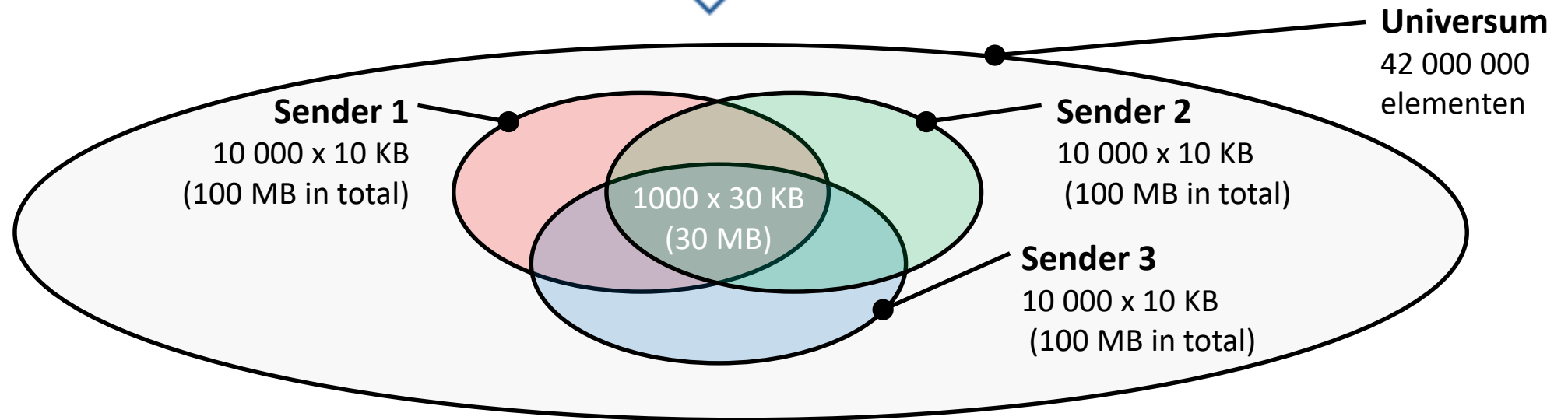
## Setup

Data in-memory / Ubuntu 18.04, Intel Core i5-8250 CPU @ 1,60Ghz, 6GB

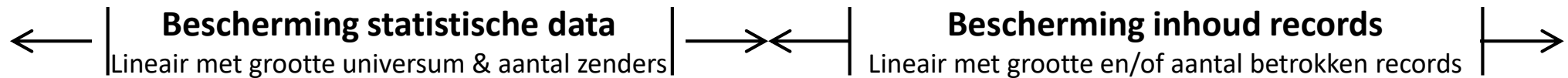
4 cores, 8 threads, one thread used.

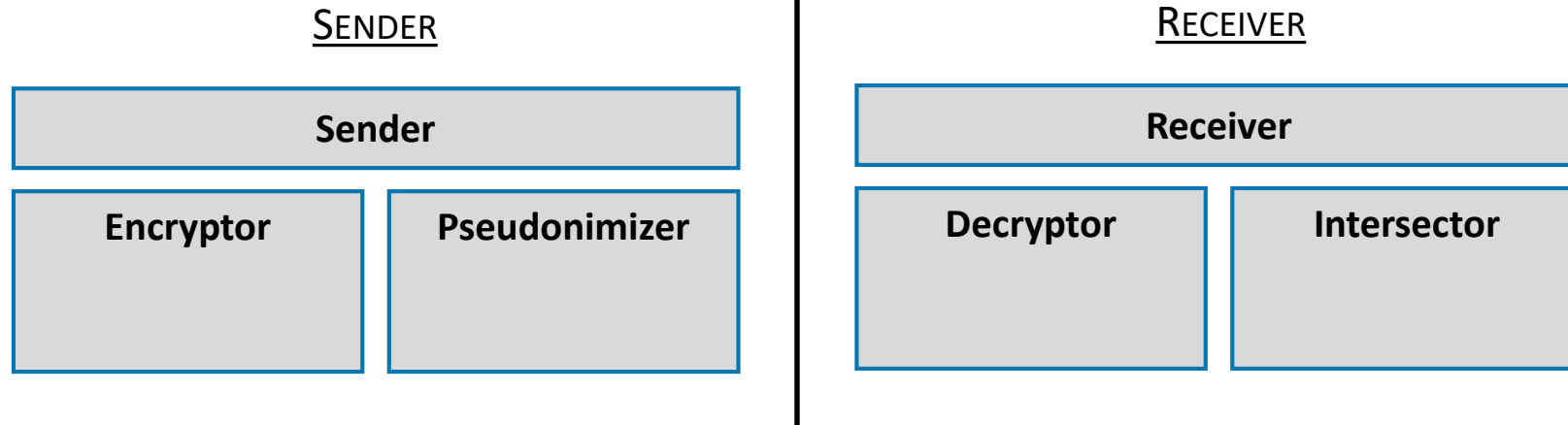
Measurements only of crypto calculations, not of storage IO or communication

# Oblivious join - Performantie



	Sender			Receiver
	Data from senders	Data to senders	Data to receiver	Data from senders
128 bits	1,9 GB	1,9 GB	100 MB	3 * 100 MB = 300 MB
256 bits	3,9 GB	3,9 GB	100 MB	3 * 100 MB = 300 MB





## Status code

- ▶ Library voor test- en demodoeleinden
- ▶ Todo: Opsplitsen zenders
- ▶ Todo: Efficiëntieverhoging
- ▶ Bij voldoende interesse verder afwerken en naar projectmodus
- ▶ Open source?

## Beschrijving

- ▶ In wording: technische beschrijving
- ▶ Later: academische review
- ▶ Later: toegankelijke tekst

## Overleg

- ▶ Juridische diensten
- ▶ Veiligheidsdiensten
- ▶ KSZ, eHealth, Rijksregister



**Smals Research werkt aan een generieke oplossing**

Het kruisen van gegevens op een gestandaardiseerde manier

zonder dat daarbij ongewenst gegevens lekken naar zenders of ontvanger

# Oblivious join

Concept door Smals Research

Theorie:



Eigen code:



Getest:



Use case:



# Geavanceerde cryptografie

## Speciale cijfertekst

- 1 Threshold encryption
- 2 Format-preserving encryption
- 3 Proxy reencryption

## Authenticatie

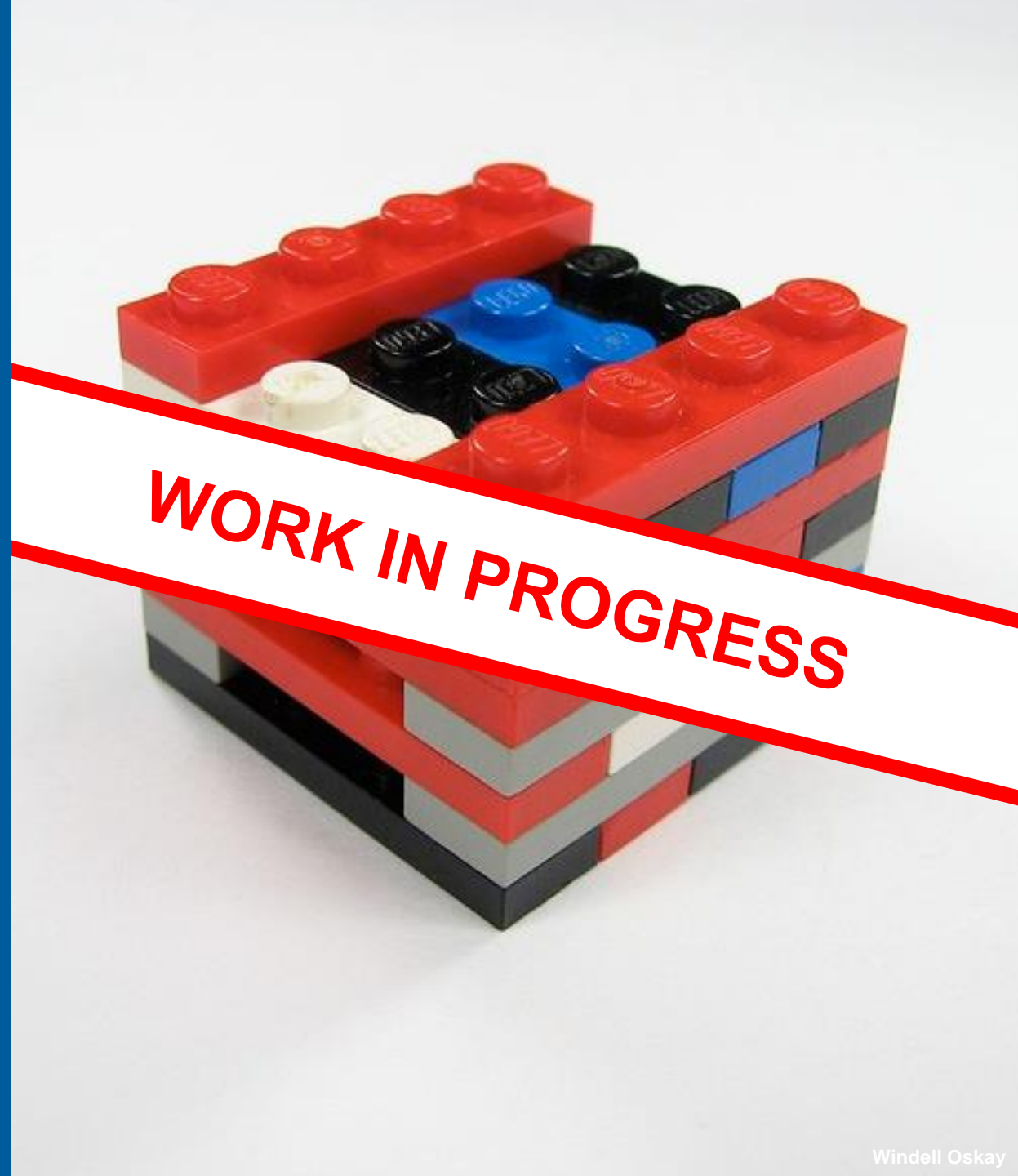
- 4 Secure remote password protocol
- 5 Attribute-based credentials

---

## Privacyvriendelijke opvraging

- 6 Secure multiparty computation
- 7 Oblivious transfer
- 8 Private set intersection
- 9 Oblivious join

Er is veel meer!



# Geavanceerde cryptografie

	Maturiteit	Gebruik
<b>Speciale cijfertekst</b> <ul style="list-style-type: none"><li>1 Threshold encryption</li><li>2 Format-preserving encryption</li><li>3 Proxy reencryption</li></ul>	<ul style="list-style-type: none"><li>●</li><li>●</li><li>●</li></ul>	<ul style="list-style-type: none"><li>Sterke confidentialiteit &amp; hoge beschikbaarheid gegevens</li><li>Pseudonimiseren persoonsgegevens met behoud structuur id</li><li>Delegeren decryptiemogelijkheid</li></ul>
<b>Authenticatie</b> <ul style="list-style-type: none"><li>4 Secure remote password protocol</li><li>5 Attribute-based credentials</li></ul>	<ul style="list-style-type: none"><li>●</li><li>●</li></ul>	<ul style="list-style-type: none"><li>Maximale veiligheid met behulp van paswoord</li><li>Selectief prijsgeven persoonsgegevens door burger</li></ul>
<b>Privacyvriendelijke opvraging</b> <ul style="list-style-type: none"><li>6 Secure multiparty computation</li><li>7 Oblivious transfer</li><li>8 Private set intersection</li><li>9 Oblivious join</li></ul>	<ul style="list-style-type: none"><li>●</li><li>●</li><li>●</li><li>●</li></ul>	<ul style="list-style-type: none"><li>Gedecentraliseerd uitvoeren code met confidentiële invoer</li><li>Opvragen gegevens over specifieke burger</li><li>Identificeren burgers met dossier bij mij én bij andere entiteit</li><li>Kruisen &amp; pseudonimiseren persoonsgegevens</li></ul>

● Gestandaardiseerd.

● Niet gestandaardiseerd. Lijkt wel inzetbaar.

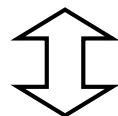
● Potentieel nuttig. Wellicht nog te complex.

● Beloftevol. Wordt verder uitgewerkt.

*“[Passende technische en organisatorische] maatregelen kunnen onder meer bestaan in het minimaliseren van de verwerking van persoonsgegevens, het zo spoedig mogelijk pseudonimiseren van persoonsgegevens, transparantie met betrekking tot de functies en de verwerking van persoonsgegevens, het in staat stellen van de betrokkene om controle uit te oefenen op de informatieverwerking en uit het in staat stellen van de verwerkingsverantwoordelijke om beveiligingskenmerken te creëren en te verbeteren. ”*

GDPR, hoofdstuk 1 §78

**Juridische stimulans om verder te gaan dan gebruik van ‘crypto werkpaarden’**

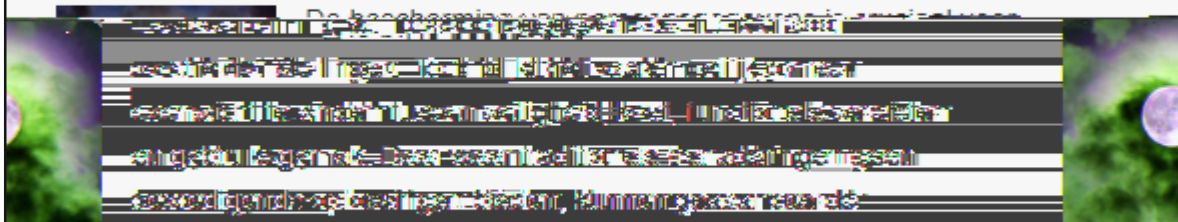


**Oplossingen moeten wel in overeenstemming zijn met regulering**



## Bescherming van persoonsgegevens met geavanceerde cryptografie

Posted on 17/09/2019 by Kristof Verslype



[Continue reading](#) →

security | Tagged [cryptografie](#), [Privacy](#), [Privacy by design](#), [Security](#) | [Leave a](#)

Posted in [S](#)  
[reply](#)

## Linking Together Personal Data in the Era of Big Data & GDPR

Posted on 18/04/2018 by Kristof Verslype



In May 2018, the much-discussed GDPR will be enacted. Besides identified data and anonymous data, the European regulation introduces a new category of data, called pseudonymous data. This articles presents an approach, based on cryptographic pseudonyms, that can help governments to ... [Continue reading](#)

→

Posted in [Big Data](#), [E-gov](#), [Security](#) | Tagged [Analytics](#), [Big Data](#), [cryptography](#), [gdpr](#), [Privacy](#), [Privacy by design](#), [pseudonym](#), [Security](#) | [Leave a reply](#)

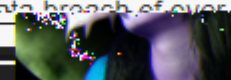
## Cryptografische pseudoniemen snellen de GDPR te hulp

Posted on 21/05/2019 by Kristof Verslype



Er worden steeds meer persoonsgegevens verwerkt, die dan ook op een afdoende manier beschermd moeten worden. Vaak volstaan de genomen veiligheidsmaatregelen niet en lezen we in de pers over opnieuw gecodeerde bronch of over het niet

[Continue reading](#) →

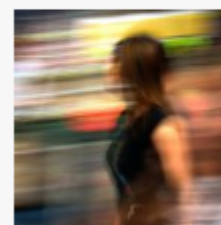


[gdpr](#), [Privacy](#), [Privacy by design](#),  
| [Leave a reply](#)

Posted in [Security](#) | Tagged [cryptography](#),  
[pseudonym](#), [pseudonymisation](#), [Security](#)

## “Vergeetachtige verzending” voor vertrouwelijk onderzoek naar personen

Posted on 18/06/2019 by Kristof Verslype



Geregeld is onderzoek nodig naar verdachte personen. Dit neemt niet weg dat de privacy van deze en andere personen gerespecteerd moet worden. Ook de confidentialiteit van het onderzoek moet gegarandeerd blijven. Dit artikel reikt een waardevolle technologie aan om aan ... [Continue reading](#) →

Posted in [Security](#) | Tagged [cryptography](#), [oblivious transfer](#), [Privacy](#), [Privacy by design](#), [Security](#) | [Leave a reply](#)



Niet bedoeling dat u alle details onthoudt

Intuïtie voor wat eventueel met crypto  
opgelost kan worden

**Functionele  
vereisten**



**Security & privacy  
vereisten**

Misschien minder noodzaak  
aan vertrouwde partij dan gedacht

**Ziet u een potentiële toepassing?**



Kristof Verslype  
Cryptographer, PhD  
Smals Research



 [kristof.verslype@smals.be](mailto:kristof.verslype@smals.be)

 [www.smals.be](http://www.smals.be)  
[www.smalsresearch.be](http://www.smalsresearch.be)  
[www.cryptov.net](http://www.cryptov.net) (personal)



## Books

- Simon Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. 2011
- Jonathan Katz, Yehuda Lindell. *Introduction to Modern Cryptography*. 2014.
- Kai Rannenberg, Jan Camenisch. *Attribute-based Credentials for Trust: Identity in the Information Society*. 2016.

## Websites

- BlueKrypt. Cryptographic Key Length Recommendation. [www.keylength.com](http://www.keylength.com)
- Vitalink – Gegevens delen, van Vitaal belang. [www.vitalink.be](http://www.vitalink.be)
- The Stanford SRP Homepage. <http://srp.stanford.edu/>

## Documents

- Julien Cathalo (Smals Research). *Threshold Encryption*. <https://www.smalsresearch.be/publications/document/?docid=65>
- NIST. *SP 800-38G: Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*. 2016. <https://csrc.nist.gov/publications/detail/sp/800-38g/final>
- ECRYPT-CSA. *D5.4 Algorithms, Key Size and Protocols Report*. 2018.

## Academic papers

- Mihir Bellare, Tomas Ristenpart, Phillip Rogaway, Till Stegers. *Format-Preserving Encryption*. 2009. <https://eprint.iacr.org/2009/251.pdf>
- Benny Pinkas, Thomas Schneider, Michael Zohner. *Scalable Private Set Intersection Based on OT Extension*. 2018. <https://dl.acm.org/citation.cfm?id=3154794>
- David W. Archer, Dan Bogdanov, Liina Kamm, Yehuda Lindell, Kurt Nielsen, Jakob Illeborg Pagter, Nigel P. Smart, Rebecca N. Wright. *From Keys to Databases – Real-World Applications of Secure Multi-Party Computation*. 2018. <https://eprint.iacr.org/2018/450.pdf>
- M Byali, A Patra, D Ravi, P Sarkar. *Fast and Universally-Composable Oblivious Transfer and Commitment Scheme with Adaptive Security*. IACR Cryptology ePrint Archive, 2017
- C. Peikert, V. Vaikuntanathan, B. Waters. *A Framework for Efficient and Composable Oblivious Transfer*. CRYPTO 2008: Advances in Cryptology – CRYPTO 2008 pp 554-571

100 Benoît Libert, Damien Vergnaud. *Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption*. 2008