A third generation eID scheme based on polymorphic pseudonymization

Eric Verheul

Agenda

- Motivation and context eID scheme
- Design choices Dutch eID scheme
- Four privacy choices made
- The eID scheme 2.0

Movivation eID Scheme

Dutch Government ambition "Digital 2017"

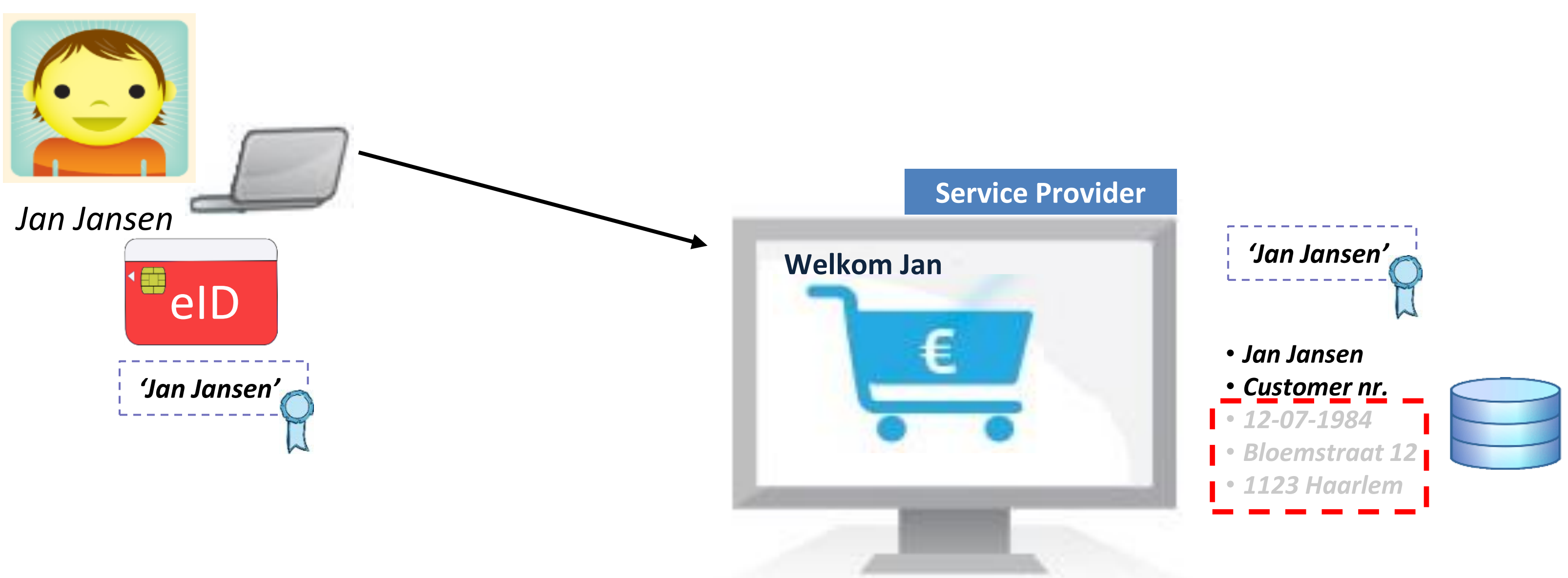
- In 2017, Dutch citizens and businesses should be able to digitally interact with Dutch government.
- Dutch government wants to provide strong forms of authentication to its citizens.
- Belgium, Denmark, Germany, Estonia, Luxembourg, Portugal, Spain, Sweden preceded the Netherlands.

eID scheme context: first generation eIDs

- First generation eID schemes were developed at the beginning of the century, e.g. in Belgium (2003).
- Here, the government issues an eID card to its citizens, i.e. a smart card with a digital certificate installed. The certificate contains the full name of the citizen.
- The eID card can be used for both public (e-government) and private service providers.

eID (card) attention points

- Reliability: how to prevent mistaken identities (“identiteitsverwisselingen”) and identity theft?
- Privacy: is it acceptable that a private party (‘webshop’) always gets access to the name of the user?
- User-friendliness: is *one* eID card acceptable, i.e. in all circumstances? Will this always conveniently work on a PC, tablet, smartphone, etc.?



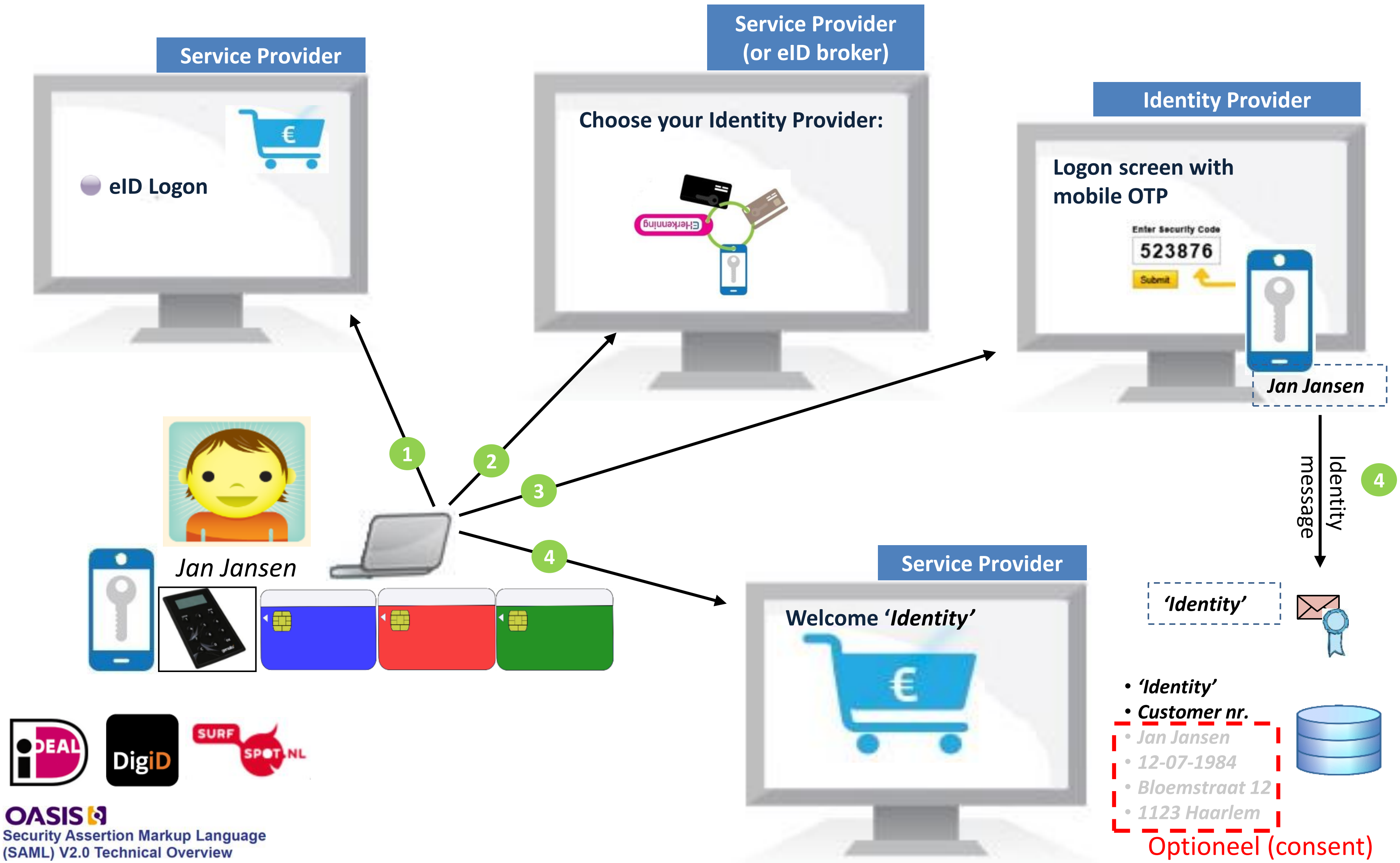
eID scheme context: second generation eIDs

The German eID card (2010):

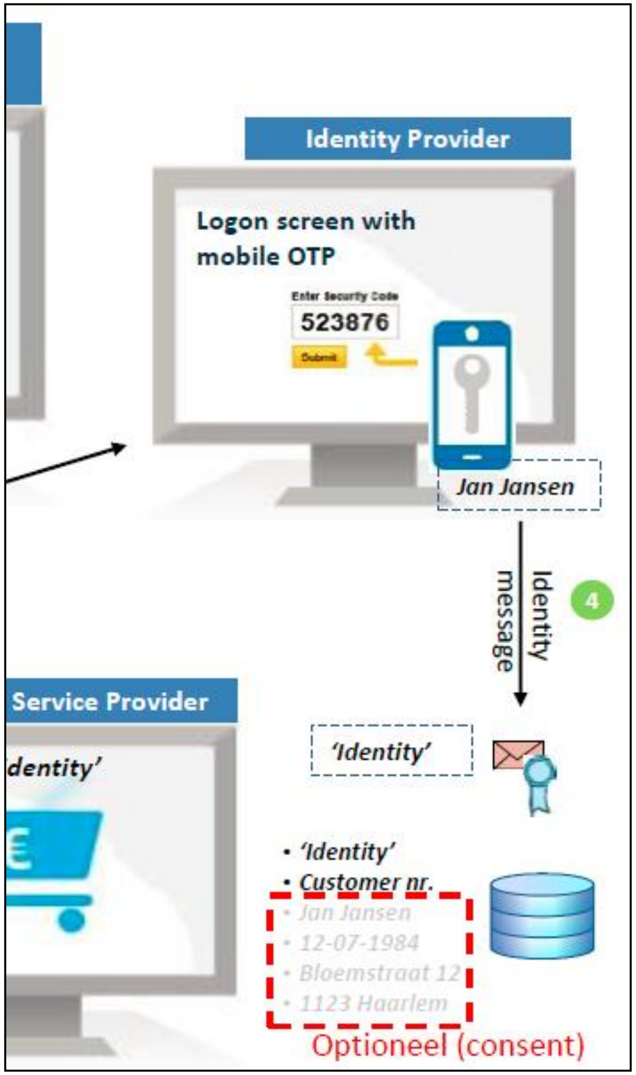
- is a second generation eID scheme based on electronic passport technology to securely read fingerprints from an e-passport (EAC),
- can also be used for public and private services,
- provides pseudonyms to (private) service providers; each has its own pseudonym domain,
- these pseudonyms can be supplemented with attributes under user consent,
- is of one type (form factor) that will not work conveniently in all contexts.

∴ The German eID scheme is privacy friendly but not particularly user-friendly.

From requirements to Dutch eID choices: federated authentication

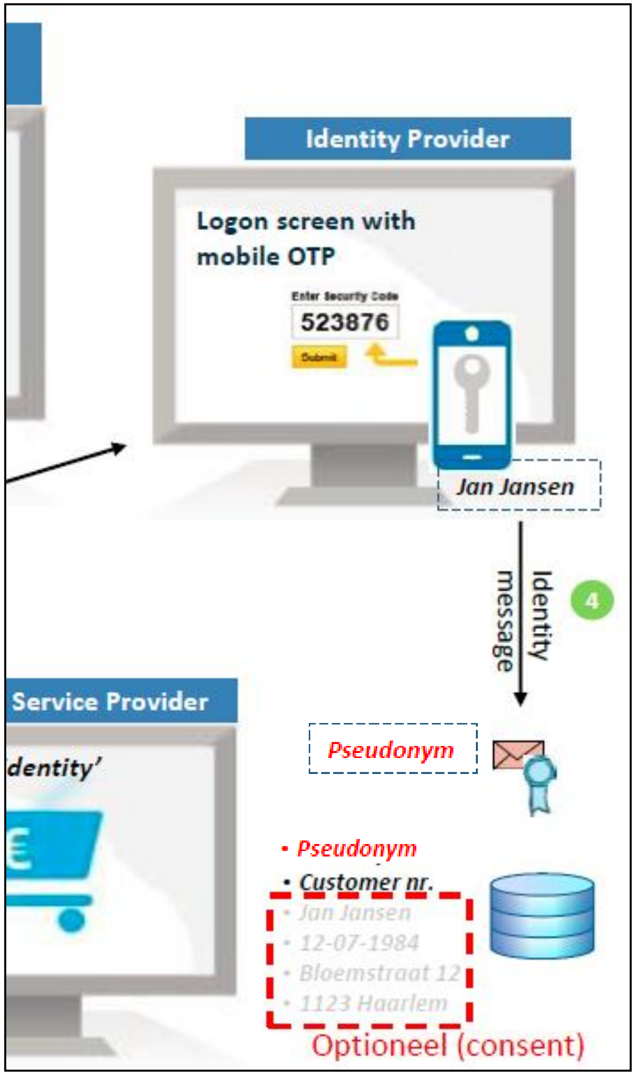


From requirements to choices: other choices



- (Strong) authentication quality based on STORK.
- Extra layer on top of SAML. Messages between parties are signed and encrypted in line with SAML. This leaves room for various privacy implementations.
- eID brokers do not see (unencrypted) personal data.
- eID identities are independent of the identity provider.
- eID identities are unique. So, different users have different Identities.
- Identities are based on the BSN without introducing linkable numbers. See next slide.

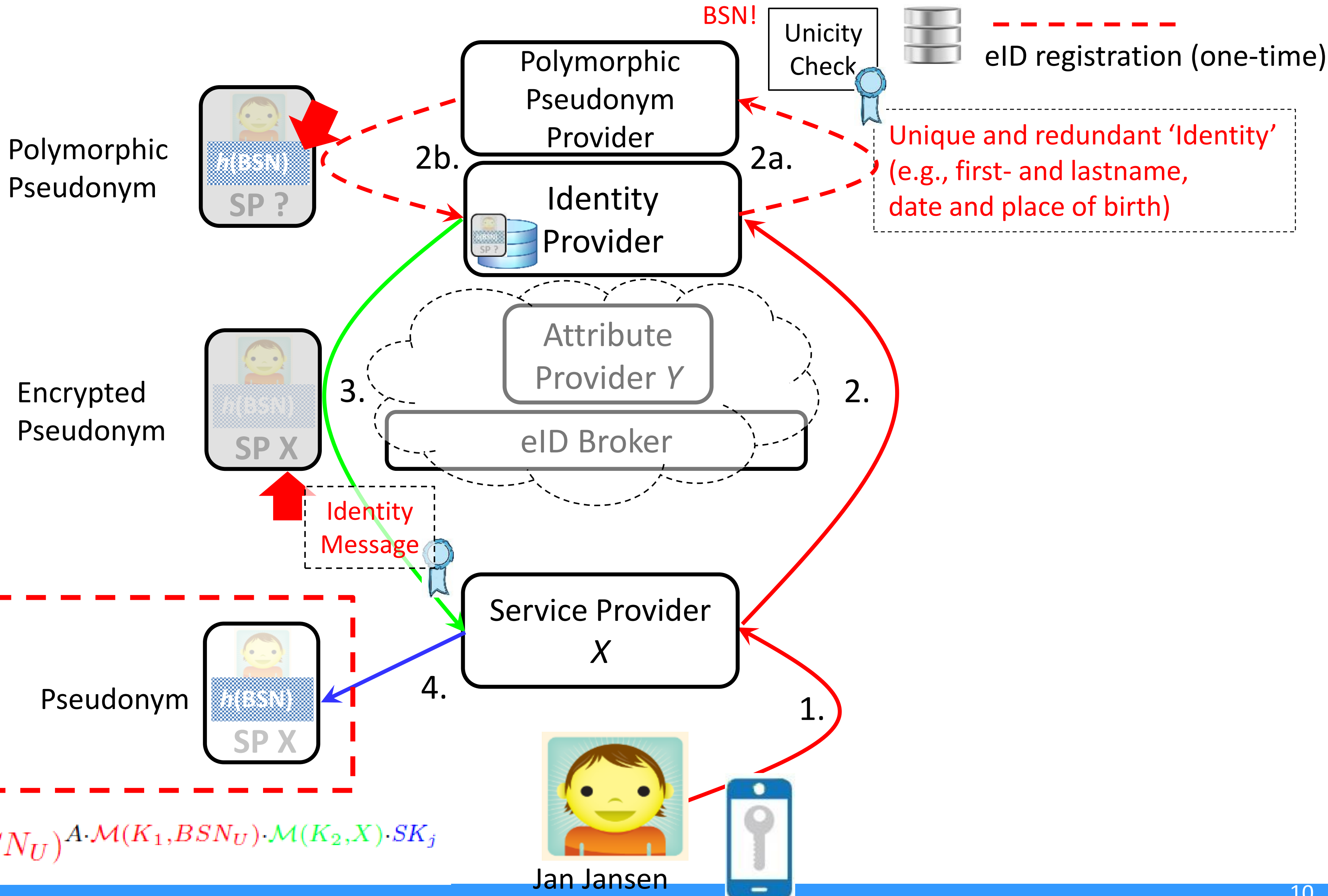
Privacy choices



Specific privacy choices eID v.2

1. Identities are based on BSN pseudonyms instead of personal data.
2. Identity Providers should not have access to *person numbers*, e.g. the pseudonyms!, to prevent linking issues.
3. There should be support for 'anonymity', i.e. that Identity Providers do not know which Service Providers their clients visit.
4. eID pseudonymity should be removable for law enforcement.

Setup of the eID scheme 2.0



$\mathcal{I}(BSN_U)^{A \cdot \mathcal{M}(K_1, BSN_U) \cdot \mathcal{M}(K_2, X) \cdot SK_j}$

Polymorphic Pseudonym metaphor

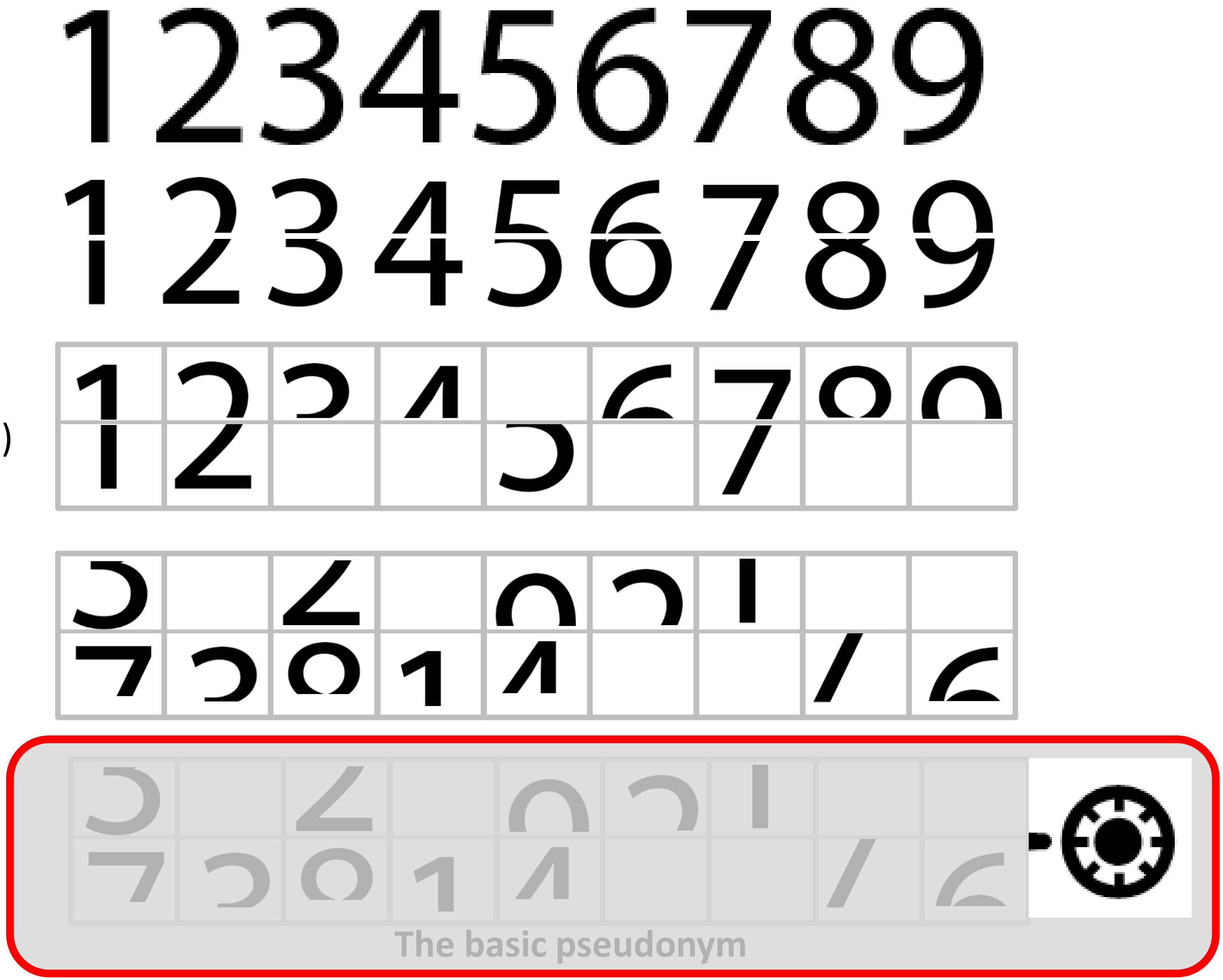
BSN of
Jan Jansen

BSN cut in pieces

Some BSN pieces
Thrown away (hash)

Mixing of
Left pieces

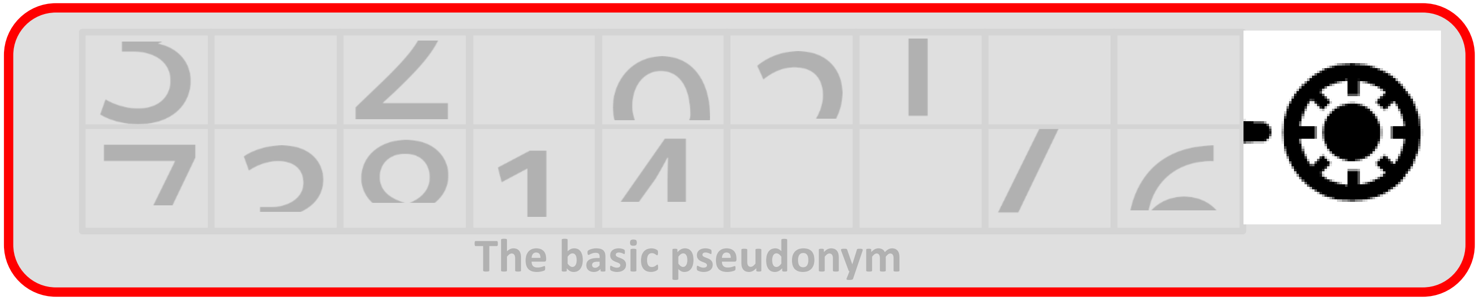
Placing pieces in
a ‘shake vault’ of
an Identity Provider



The polymorphic pseudonym at the IdP

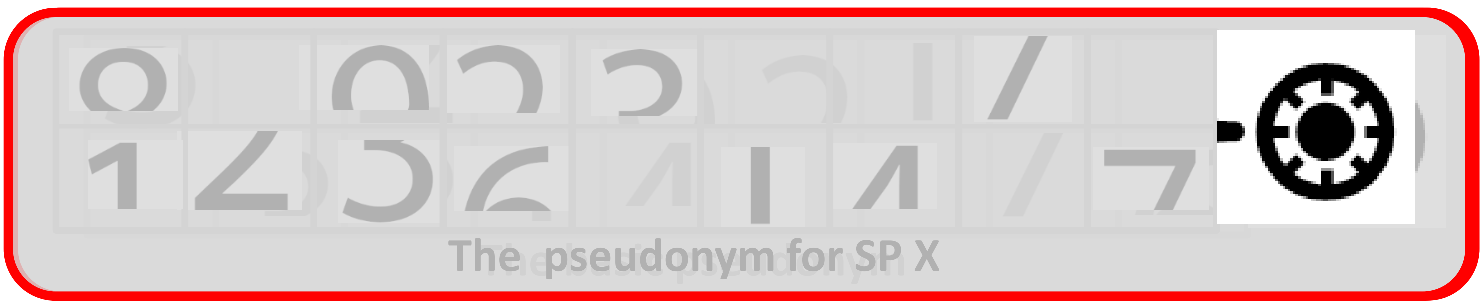
Polymorphic Pseudonym → Encrypted Pseudonym

Placing pieces in a ‘shake vault’ of an Identity Provider



The polymorphic pseudonym at the IdP

Identity Provider applies ‘shake instructions’ specific for SP X



The **encrypted** pseudonym for SP X

The ElGamal public key system

- Let $G = \langle g \rangle$ be a multiplicative group of prime order q generated by a generator element g.
- By GF(q) we denote the Galois field of the integers modulo q.
- We assume that the Discrete Log, Diffie-Hellman and Decision Diffie-Hellman problems are hard in G. For instance, G is the elliptic curve group based on brainpoolP320r1.
- For a random $k \in \mathrm{GF}(q)$ the ElGamal encryption of plaintext $S \in G$ under public key $y = g^x$ is $\mathcal{EG}(S, y, k) = (g^k, S \cdot y^k, y)$. Sometimes we simply write $\mathcal{EG}(S, y)$, i.e. do not mention k.

The ElGamal public key system

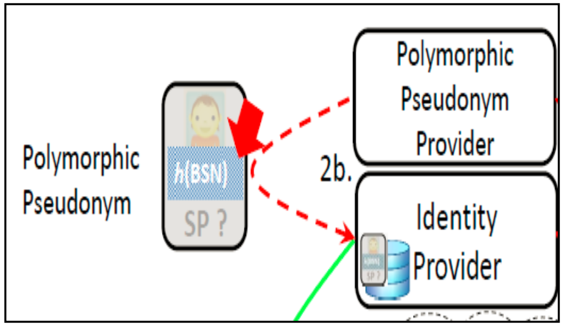
- Let $G = \langle g \rangle$ be a multiplicative group of prime order q generated by a generator element g.
- By $\mathrm{GF}(q)$ we denote the Galois field of the integers modulo q.
- We assume that the Discrete Log, Diffie-Hellman and Decision Diffie-Hellman problems are hard in G. For instance, G is the elliptic curve group based on brainpoolP320r1.
- For a random $k \in \mathrm{GF}(q)$ the ElGamal encryption of plaintext $S \in G$ under public key $y = g^x$ is $\mathcal{EG}(S, y, k) = (g^k, S \cdot y^k, y)$. Sometimes we simply write $\mathcal{EG}(S, y)$, i.e. do not mention k.

Proposition *Let $\mathcal{EG}(S, y, k) = (A, B; C)$ be an ElGamal encryption of plaintext S under public key $y = g^x$ and let z be an element of $\mathrm{GF}(q)^*$. Then the following equalities hold:*

1. $(A^z, B^z, C) = \mathcal{EG}(S^z, y, k{\cdot}z)$, $\longrightarrow$ Change plaintext inside
2. $(A^z, B, C^{(z^{-1})}) = \mathcal{EG}(S, y^{(z^{-1})}, k{\cdot}z)$, $\longrightarrow$ Change private key
3. $(A \cdot g^z, B \cdot C^z, C) = \mathcal{EG}(S, y, k + z)$. $\longrightarrow$ Re-randomize

Proposition *Deciding that $\mathcal{EG}(S_1, y, k)$ and $\mathcal{EG}(S_2, y, k')$ hold the same plaintext, i.e. that $S_1 = S_2$ is equivalent to the Decision Diffie-Hellman problem and is thus hard.* $\longrightarrow$ Re-randomized ElGamal encryptions are indistinguishable.

Forming of PPs



The *PP provider* (at one-time registration of IdP client)

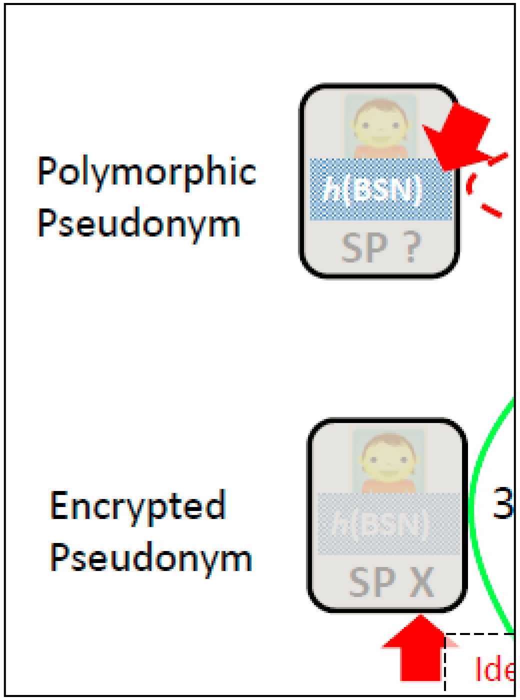
- For each identity provider, say the *i*-th, the pseudonym provider is provided its public key f_i by the KMA. The identity provider does not possess the private key!
- The pseudonym provider calculates the following ElGamal encryption during the one time generation of a user polymorphic pseudonym based on the BSN:

$$BSN_U$$

$$\mathcal{E}\mathcal{G}(\ \mathcal{I}(BSN_U)^{A \cdot \mathcal{M}(K_1, BSN_U)}, f_i)$$

Randomized (non-linkable person numbers)

Transforming PPs to EPs



The Identity provider (at each authentication), transforms the polymorphic pseudonym to a form suitable for the service provider X inside the ElGamal encryption from the outside in three steps:

$$\mathcal{EG}(\ \mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)}, f_i)$$

Registered PP

$$\mathcal{EG}(\ \mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)}, f_i)$$

Change inside pseudonym

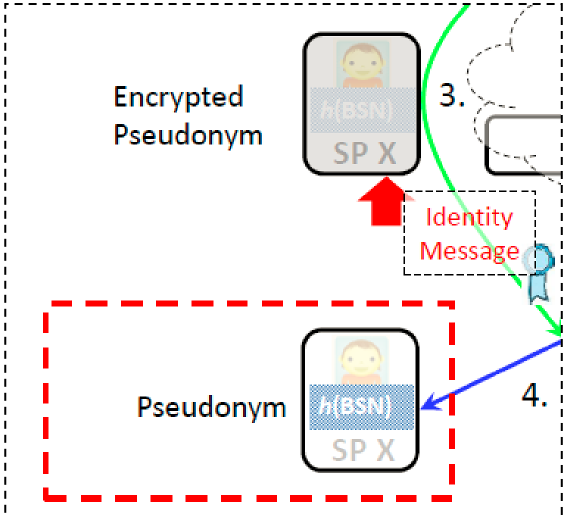
$$\mathcal{EG}(\ \mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)}, f_i^{\frac{1}{\mathcal{M}(K_3,X)}})$$

Change Decryption key

$$\boxed{\widetilde{\mathcal{EG}}(\ \mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)}, f_i^{\frac{1}{\mathcal{M}(K_3,X)}})}$$

Re-randomize (non-linkable person numbers)

Transforming EPs to Ps



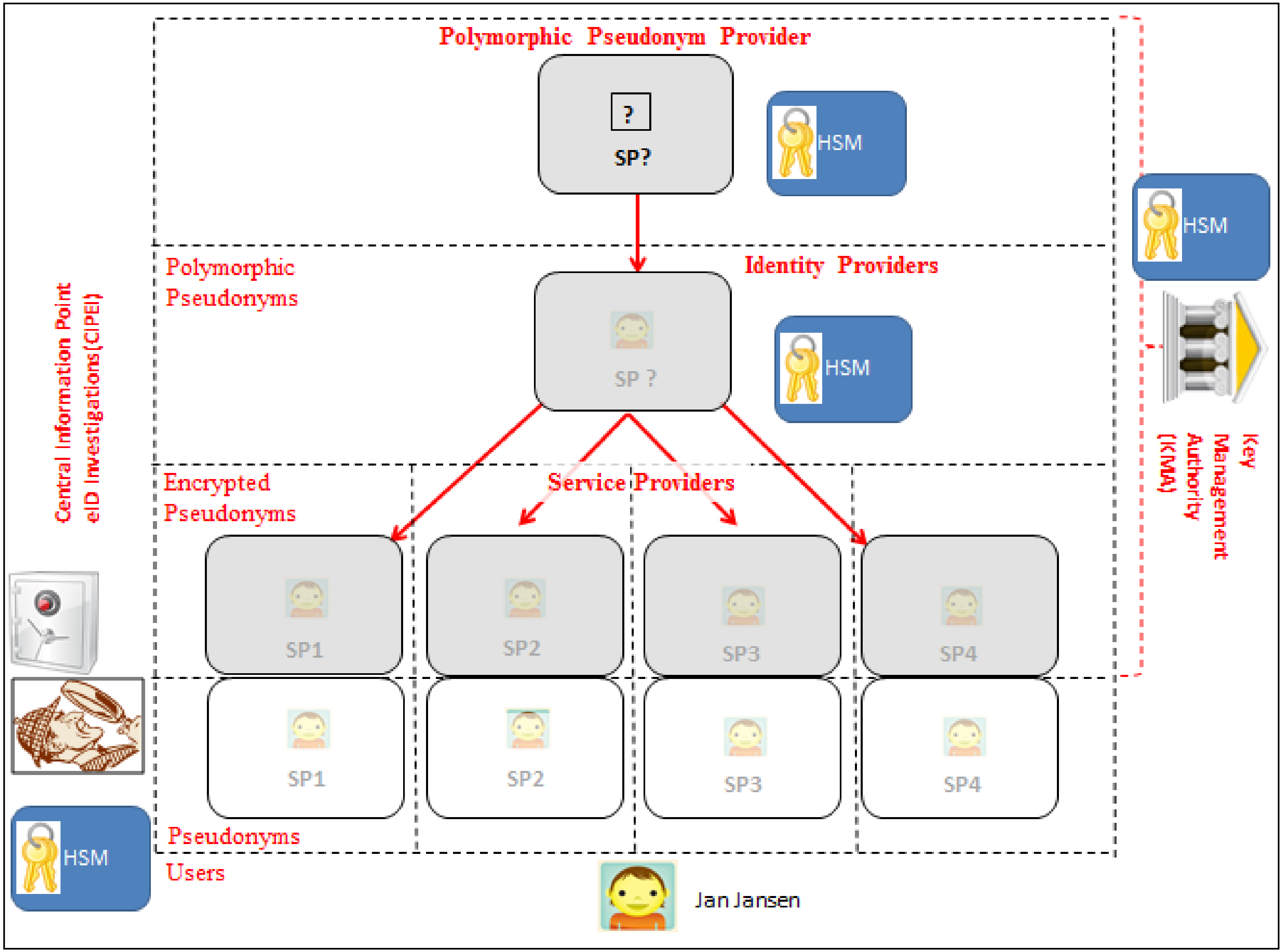
Finally, the *service provider* decrypts the encrypted pseudonym and raises the expression to its key *SK_j* after receipt, ‘closing’ the pseudonym.

$$\widetilde{\mathcal{EG}}\left(\mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)}, f_i^{\frac{1}{\mathcal{M}(K_3,X)}} \right)$$ ***Received EP***

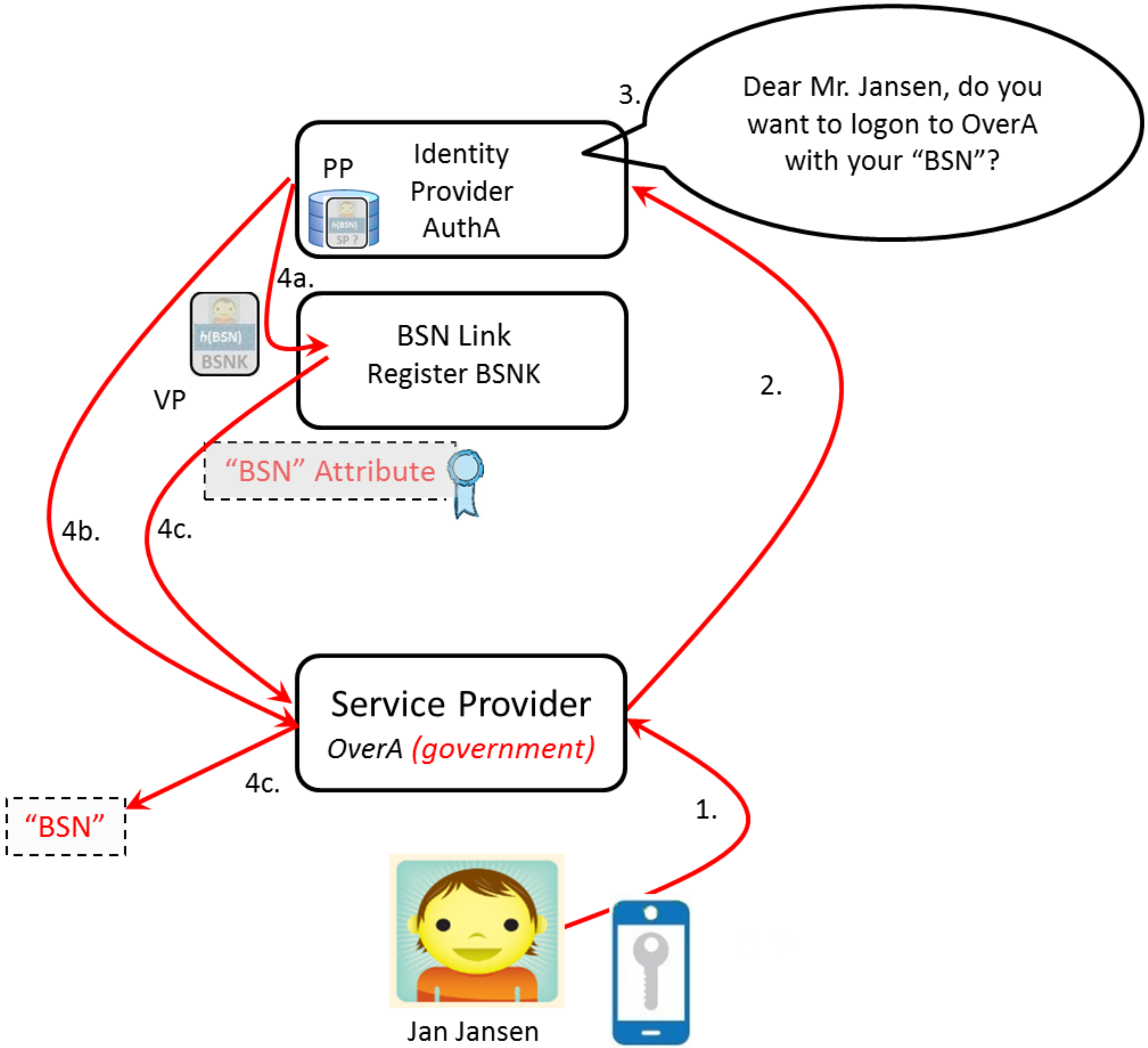
$$\mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)}$$ ElGamal decrypt

$$\boxed{\mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)\cdot SK_j}}$$ Pseudonym Closing

eID scheme 2.0 infrastructure



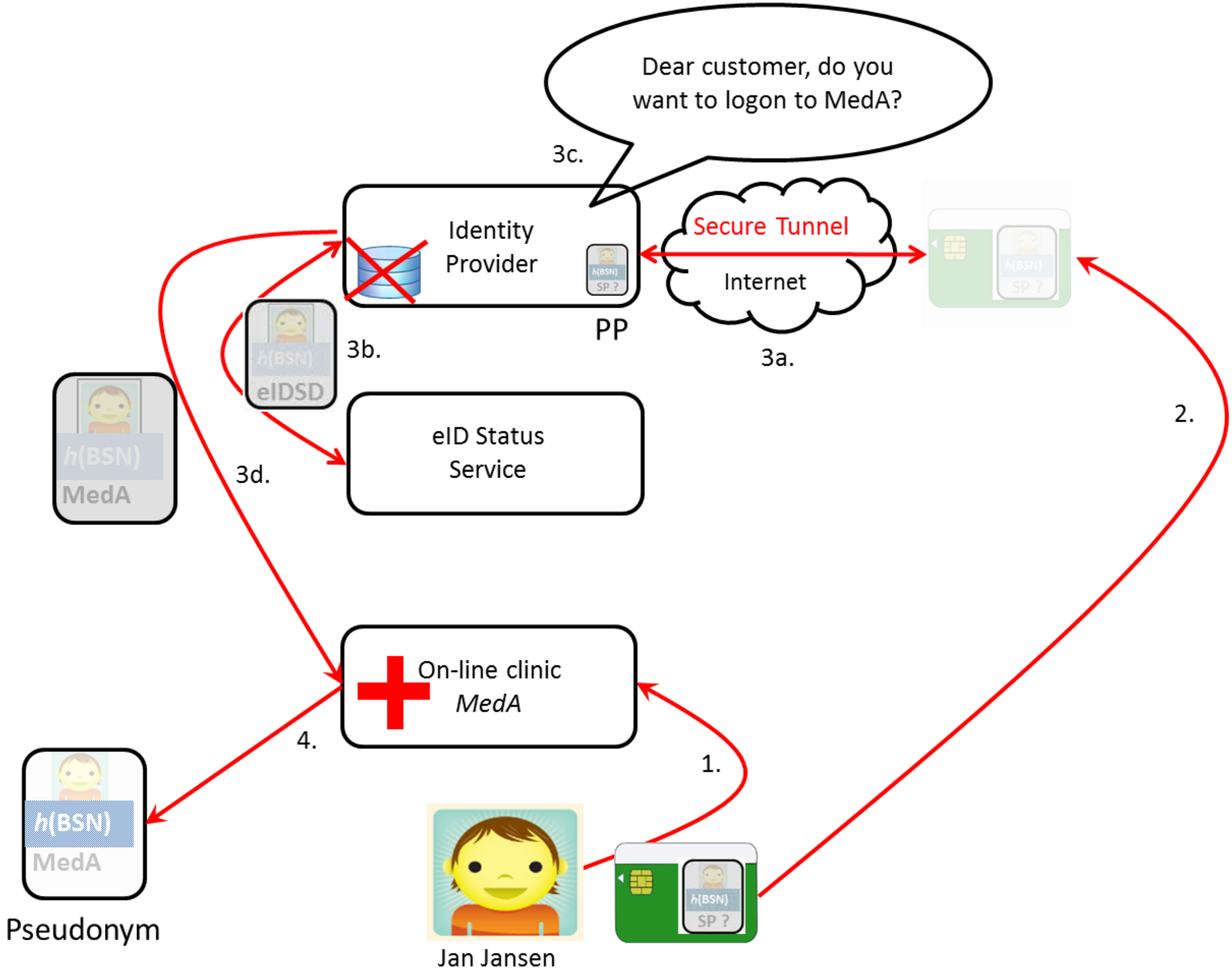
Basic Use case (registered user)



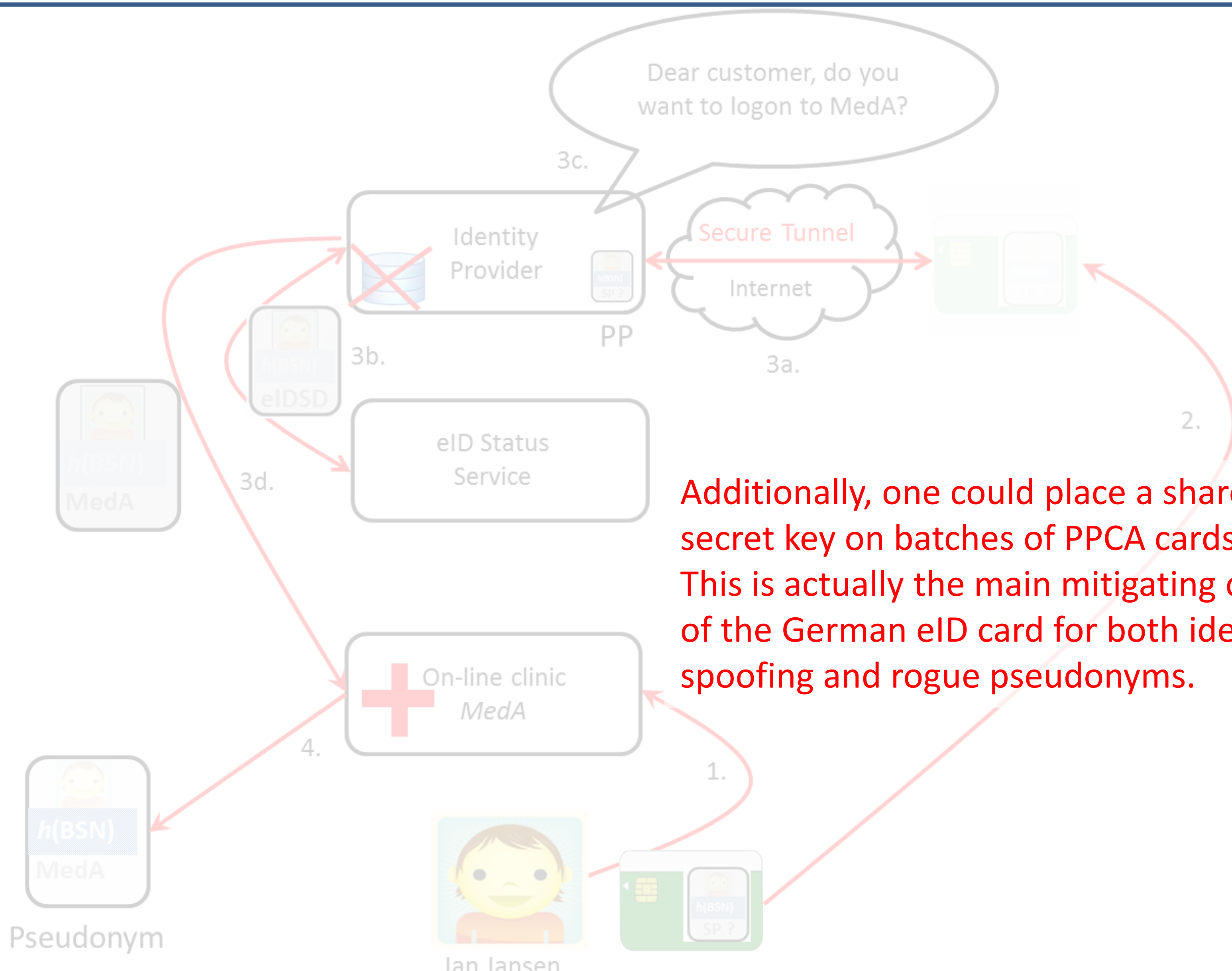
Anonymous eID access: PPCA

- PPCA supports that the Identity Provider can authenticate its customers, but does not know which service providers they visit.
- Basic idea: place the user polymorphic pseudonym on a smartcard issued by the Identity Provider instead of in the Identity Provider client database.
  - If the card would provide the same polymorphic pseudonym to the Identity Provider each time it would make it linkable. *To address this: re-randomize the ElGamal encryption on the card*. *ElGamal encryptions are indistinguishable!*
  - If the card cannot check it is read by the Identity Provider, then anybody can read its polymorphic pseudonyms and replay them claiming to be the user (*identity theft*). *To address this: let the Identity Provider authenticate itself to the card. This is standard technique (EAC) used in e-passports to read fingerprints*.
  - If the Identity Provider cannot check that the card is genuine one can send a random polymorphic pseudonym leading to rogue eID pseudonyms. If the Idenity Provider cannot check that the card has not been stolen/lost it can be misused. *To address both issues: introduce an eID status service. The Identity Provider queries the status of the PPCA card based on the encrypted pseudonym of the eID status service, i.e.: a) the card 'exists' and b) is not revoked.*

Anonymous eID access: PPCA



Anonymous eID access: PPCA



Additionally, one could place a shared secret key on batches of PPCA cards. This is actually the main mitigating control of the German eID card for both identity spoofing and rogue pseudonyms.

PPCA: further details

- A polymorphic pseudonym (P_1, P_2, P_3) takes the form of three points on an elliptic curve generated by a base point G. The point P_3 is the public key of the identity provider.
- A randomization is not complex; the card generates a random number *r* and forms

$$(P_1 + r*G,\ P_2 + r*P_3,\ P_3).$$

  This takes two point multiplications and two point additions.
- Some card platforms (e.g., Javacard) do not support point additions from Java code (hidden in co-processor).
- For this one can also take two randomized polymorphic pseudonyms

$$(P_1, P_2, P_3);\ (Q_1, Q_2, P_3)$$

  and return ($r*P_1$, $r*P_2$, P_3); ($s*Q_1$, $s*Q_2$, P_3) with r random and s = 1-r. This only involves point multiplications.
- The identity provider adds them, ,i.e. ($r*P_1 + s*Q_1$,$r*P_2$+ $s*Q_2$, P_3), which will give a random polymorphic pseudonym.

Legal access

Two kinds of request can be handled by the CIPEI and the KMA under *dual control*:

- "de-pseudonymization": A pseudonym and a reference to the related service provider domain is given by the law enforcement agency; the identity (BSN) of the person behind the pseudonym is requested.
- "Pseudonymization on request": The BSN of a user is given by the law enforcement agency together with a reference to a service provider domain; the pseudonym of the person in the service provider domain is requested.

- A use-case for de-pseudonymization can be a legal complaint of a service provider against a user, e.g. relating to fraud or grooming. The law enforcement agency is then able to retrieve the identity of the user.
- A use-case for pseudonymization on request can be to assess if a suspect has also been active with similar service providers.
- The CIPEI and KMA governance should give sufficient trust to the public.

$$O_1 = \mathcal{I}(BSN_U)^{\mathcal{M}(K_1,BSN_U)\cdot\psi\cdot\chi}$$

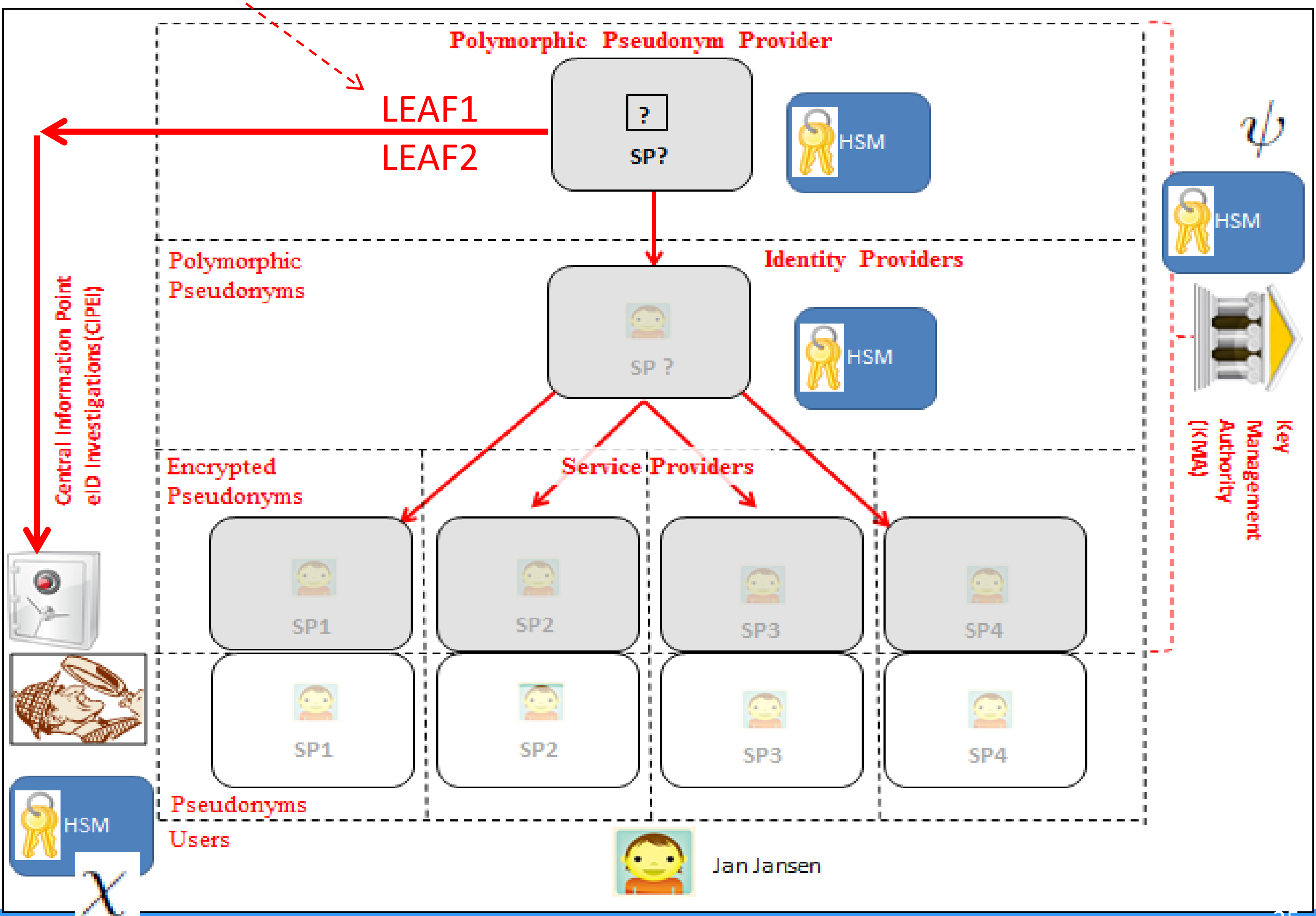
$$O_2 = \mathcal{H}(O_1)$$

$$\left.\begin{array}{l} L_{1a} = \mathcal{H}(O_2) \\ L_{1b} = \mathcal{E}(O_2, BSN_U) \end{array}\right\}$$

AES

Legal access

$$\mathcal{I}(BSN_U)^{A\cdot\mathcal{M}(K_1,BSN_U)\cdot\mathcal{M}(K_2,X)\cdot SK_j}$$



Questions?

- Details on http://www.eid-stelsel.nl/over-eid-stelsel/programma-eid/werkgroepen/.