Pseudonymous Signature on eIDAS Token Implementation Based Privacy Threats

Mirosław Kutyłowski, Lucjan Hanzlik, Kamil Kluczniak

Wrocław University of Technology, Poland

ACISP 2016,
Melbourne

Authentication

Authentication:

process that enables the electronic identification of a natural or legal person, or the origin and integrity of data in electronic form to be confirmed
(eIDAS Regulation of EU)

Authentication

Authentication:

process that enables the electronic identification of a natural or legal person, or the origin and integrity of data in electronic form to be confirmed
(eIDAS Regulation of EU)

ephemeral: **the verifier gets convinced at the moment of protocol execution**
the proof might be worthless for the third parties and not to be used for later checks
classical ZKP authentication protocols

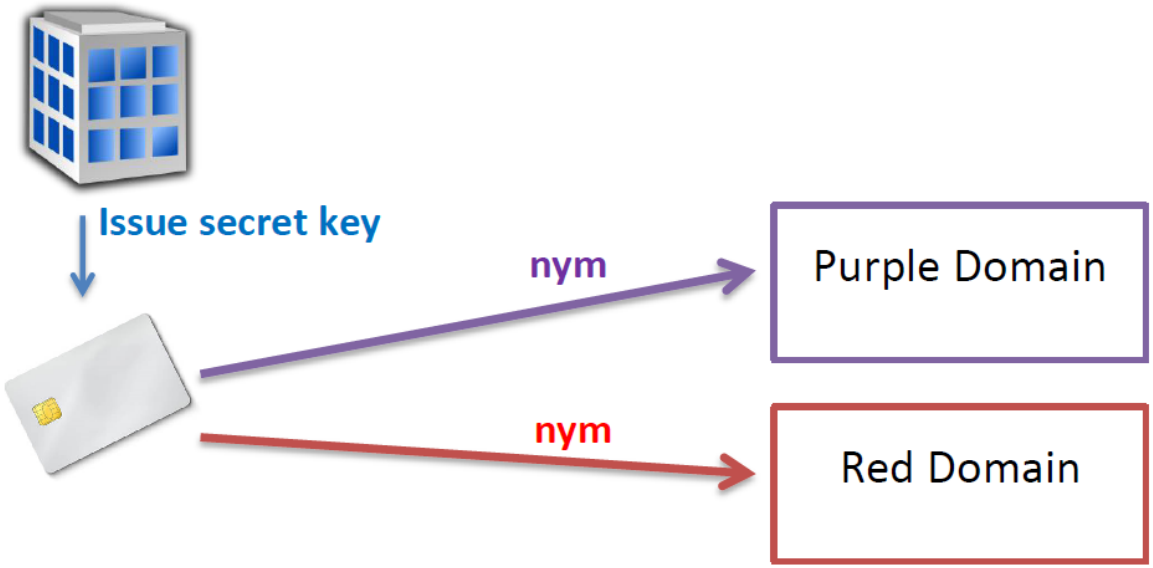
long-lasting: **the proof can be presented to third parties at a later time**
electronic signatures

Electronic Identity Documents - Anonymity

Pseudonym:

an additional ID that does not reveal the real identity

Prevents Sybil attacks - appearing under different against the same service.



Pseudonyms

Domain/Sector

service area where the user must appear under the same pseudonym.

like a user account

Pseudonyms

Domain/Sector

service area where the user must appear under the same pseudonym.

like a user account

Unlinkability

While in a sector the user must always appear under the same pseudonym, **the pseudonyms in different sectors must be unlinkable.**

Pseudonyms

Domain/Sector

service area where the user must appear under the same pseudonym.

like a user account

Unlinkability

While in a sector the user must always appear under the same pseudonym, **the pseudonyms in different sectors must be unlinkable.**

Seclusiveness

Only the issuer may admit new users. It should be infeasible to create **false identities**.

German eID - “eIDAS Token”

new version of BSI Technical Recommendation (2015)

German Federal Office for Information Security (BSI): Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token 2.20. Technical Guideline TR-03110-2 (2015)

Pseudonymous Signature for the German eID:

- a single signing key per user (regardless of the number of domains)
- a separate pair of pseudonyms per domain (public keys)
- the pseudonyms are derived on-the-fly from the secret key

German eID - “eIDAS Token”

new version of BSI Technical Recommendation (2015)

German Federal Office for Information Security (BSI): Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token 2.20. Technical Guideline TR-03110-2 (2015)

Pseudonymous Signature for the German eID:

- a single signing key per user (regardless of the number of domains)
- a separate pair of pseudonyms per domain (public keys)
- the pseudonyms are derived on-the-fly from the secret key

Disadvantages

breaking into just 2 eID documents reveals the system keys and enables to forge eIDs

German eID - “eIDAS Token”

Pseudonymous Signature - some details

System keys

The issuer generates a pair of system keys SK_{ICC} and SK_{M} both in $\mathbb{Z}_p$.

German eID - “eIDAS Token”

Pseudonymous Signature - some details

System keys

The issuer generates a pair of system keys SK_{ICC} and SK_{M} both in $\mathbb{Z}_p$.

User private keys

for user U, the issuer generates a pair of keys $\mathrm{SK}_{U,1}$ and $\mathrm{SK}_{U,2}$ s.t.:

$$\mathrm{SK}_{\mathrm{ICC}} = \mathrm{SK}_{U,1} + \mathrm{SK}_{M} \cdot \mathrm{SK}_{U,2} \bmod p$$

German eID - “eIDAS Token”

Pseudonymous Signature - some details

System keys

The issuer generates a pair of system keys SK_{ICC} and SK_{M} both in $\mathbb{Z}_p$.

User private keys

for user U, the issuer generates a pair of keys $\mathrm{SK}_{U,1}$ and $\mathrm{SK}_{U,2}$ s.t.:

$$\mathrm{SK}_{\mathrm{ICC}} = \mathrm{SK}_{U,1} + \mathrm{SK}_{M} \cdot \mathrm{SK}_{U,2} \bmod p$$

Pseudonyms

pair of pseudonyms of the user in a sector with the public key $\mathrm{PK}_{\mathrm{sector}} \in \mathbb{G}$

$$nym_{U,1} = (\mathrm{PK}_{\mathrm{sector}})^{\mathrm{SK}_{U,1}}$$
$$nym_{U,2} = (\mathrm{PK}_{\mathrm{sector}})^{\mathrm{SK}_{U,2}}$$

German eID - “eIDAS Token”

Pseudonymous Signature - some details

System keys

The issuer generates a pair of system keys SK_{ICC} and SK_{M} both in $\mathbb{Z}_p$.

User private keys

for user U, the issuer generates a pair of keys $\mathrm{SK}_{U,1}$ and $\mathrm{SK}_{U,2}$ s.t.:

$$\mathrm{SK}_{\mathrm{ICC}} = \mathrm{SK}_{U,1} + \mathrm{SK}_{M} \cdot \mathrm{SK}_{U,2} \bmod p$$

Pseudonyms

pair of pseudonyms of the user in a sector with the public key $\mathrm{PK}_{\mathrm{sector}} \in \mathbb{G}$

$$nym_{U,1} = (\mathrm{PK}_{\mathrm{sector}})^{\mathrm{SK}_{U,1}}$$
$$nym_{U,2} = (\mathrm{PK}_{\mathrm{sector}})^{\mathrm{SK}_{U,2}}$$

A signature is a “Signature of Knowledge” of the secret keys which are in the proper form.

Delegating tracing capabilities to “Big Brother”

Can the Issuer delegate the ability to link the pseudonyms to a Tracer without allowing the Tracer to forge signatures?

Delegating tracing capabilities to “Big Brother”

Can the Issuer delegate the ability to link the pseudonyms to a Tracer without allowing the Tracer to forge signatures?

Setup

- For each user U the issuer creates x_U and s_U s.t.

$$x_U = SK_{U,1} + SK_{U,2} \cdot s_U$$

$$\mathrm{SK}_{ICC} = SK_{U,1} + SK_{U,2} \cdot SK_M$$

Delegating tracing capabilities to “Big Brother”

Can the Issuer delegate the ability to link the pseudonyms to a Tracer without allowing the Tracer to forge signatures?

Setup

- For each user U the issuer creates x_U and s_U s.t.
$$x_U = SK_{U,1} + SK_{U,2} \cdot s_U$$
$$\mathrm{SK}_{ICC} = SK_{U,1} + SK_{U,2} \cdot SK_M$$
- The issuer sets a domain dependent trapdoor as $T_{domain,U} = PK_{domain}^{x_U}$ and s_U.

Delegating tracing capabilities to “Big Brother”

Can the Issuer delegate the ability to link the pseudonyms to a Tracer without allowing the Tracer to forge signatures?

Setup

- For each user U the issuer creates x_U and s_U s.t.
$$x_U = SK_{U,1} + SK_{U,2} \cdot s_U$$
$$\mathrm{SK}_{ICC} = SK_{U,1} + SK_{U,2} \cdot SK_M$$
- The issuer sets a domain dependent trapdoor as $T_{domain,U} = PK_{domain}^{x_U}$ and s_U.

Tracer

The tracer can check
$$T_{domain,U} \stackrel{?}{=} nym_{1,U} \cdot nym_{2,U}{}^{s_U} = PK_{domain}^{\mathrm{SK}_{U,1}} \cdot PK_{domain}^{\mathrm{SK}_{U,2} \cdot s_U}$$

How to prevent the issuer from knowing the secret keys?

Manufacturing

- After manufacturing, an eID stores two pairs of keys: $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$.

How to prevent the issuer from knowing the secret keys?

Manufacturing

- After manufacturing, an eID stores two pairs of keys: $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$.
- Both pairs satisfy $\mathrm{SK}_{ICC} = x_{1,i} + x_{2,i} \cdot \mathrm{SK}_M$ for $i = 1, 2$

How to prevent the issuer from knowing the secret keys?

Manufacturing

- After manufacturing, an eID stores two pairs of keys: $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$.
- Both pairs satisfy $\mathrm{SK}_{ICC} = x_{1,i} + x_{2,i} \cdot \mathrm{SK}_M$ for $i = 1, 2$

Personalize

1. The eID sends

   $$IN_{1,1} = g^{x_{1,1}}, IN_{2,1} = g^{x_{2,1}}, IN_{1,2} = g^{x_{1,2}}, IN_{2,2} = g^{x_{2,2}}$$

   to the document owner.

How to prevent the issuer from knowing the secret keys?

Manufacturing

- After manufacturing, an eID stores two pairs of keys: $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$.
- Both pairs satisfy $\mathrm{SK}_{ICC} = x_{1,i} + x_{2,i} \cdot \mathrm{SK}_M$ for $i = 1, 2$

Personalize

1. The eID sends
$$IN_{1,1} = g^{x_{1,1}}, IN_{2,1} = g^{x_{2,1}}, IN_{1,2} = g^{x_{1,2}}, IN_{2,2} = g^{x_{2,2}}$$
to the document owner.
2. The document owner chooses a pair a, b s.t. $a + b = 1 \mod p$ and sends it to the eID.

How to prevent the issuer from knowing the secret keys?

Manufacturing

- After manufacturing, an eID stores two pairs of keys: $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$.
- Both pairs satisfy $\mathrm{SK}_{ICC} = x_{1,i} + x_{2,i} \cdot \mathrm{SK}_M$ for $i = 1, 2$

Personalize

1. The eID sends
$$IN_{1,1} = g^{x_{1,1}}, IN_{2,1} = g^{x_{2,1}}, IN_{1,2} = g^{x_{1,2}}, IN_{2,2} = g^{x_{2,2}}$$
to the document owner.
2. The document owner chooses a pair a, b s.t. $a + b = 1 \mod p$ and sends it to the eID.
3. The eID should now hold the pair $\mathrm{SK}_{U,1} = a \cdot x_{1,1} + b \cdot x_{1,2}$ and $\mathrm{SK}_{U,2} = a \cdot x_{2,1} + b \cdot x_{2,2}$

Are such keys in the proper form?

$$\mathrm{SK}_{U,1} = a \cdot x_{1,1} + b \cdot x_{1,2} \quad \text{and} \quad \mathrm{SK}_{U,2} = a \cdot x_{2,1} + b \cdot x_{2,2}$$

Note that

$$\begin{aligned}
\underline{\mathrm{SK}_{U,1} + \mathrm{SK}_{U,2} \cdot SK_M} = \\
a \cdot x_{1,1} + b \cdot x_{1,2} + (a \cdot x_{2,1} + b \cdot x_{2,2}) \cdot SK_M = \\
a \cdot (x_{1,1} + x_{2,1} \cdot \mathrm{SK}_M) + b \cdot (x_{1,2} + x_{2,2} \cdot SK_M) = \\
a \cdot \mathrm{SK}_{ICC} + b \cdot \mathrm{SK}_{ICC} = \\
\mathrm{SK}_{ICC} \cdot (a \cdot b) \mod p = \\
\underline{SK_{ICC}}
\end{aligned}$$

How to prevent the issuer from knowing the secret keys?

Keys on the smart card:

$$\mathrm{SK}_{U,1} = a \cdot x_{1,1} + b \cdot x_{1,2} \quad \text{and} \quad \mathrm{SK}_{U,2} = a \cdot x_{2,1} + b \cdot x_{2,2}$$

How to prevent the issuer from knowing the secret keys?

Keys on the smart card:

$$\mathrm{SK}_{U,1} = a \cdot x_{1,1} + b \cdot x_{1,2} \quad \text{and} \quad \mathrm{SK}_{U,2} = a \cdot x_{2,1} + b \cdot x_{2,2}$$

The Owner

1. Computes and stores
   $I_1 \leftarrow IN_{1,1}^{a} \cdot IN_{1,2}^{b} = g^{SK_{U,1}}$ and $I_2 \leftarrow IN_{2,1}^{a} \cdot IN_{2,2}^{b} = g^{SK_{U,2}}$

How to prevent the issuer from knowing the secret keys?

Keys on the smart card:

$$\mathrm{SK}_{U,1} = a \cdot x_{1,1} + b \cdot x_{1,2} \quad \text{and} \quad \mathrm{SK}_{U,2} = a \cdot x_{2,1} + b \cdot x_{2,2}$$

The Owner

1. Computes and stores
   $I_1 \leftarrow IN_{1,1}^{a} \cdot IN_{1,2}^{b} = g^{SK_{U,1}}$ and $I_2 \leftarrow IN_{2,1}^{a} \cdot IN_{2,2}^{b} = g^{SK_{U,2}}$
2. So he may verify the smart card giving it as input a domain public key $PK_{domain} = g^{h}$, with known h.

How to prevent the issuer from knowing the secret keys?

Keys on the smart card:

$$\mathrm{SK}_{U,1} = a \cdot x_{1,1} + b \cdot x_{1,2} \quad \text{and} \quad \mathrm{SK}_{U,2} = a \cdot x_{2,1} + b \cdot x_{2,2}$$

The Owner

1. Computes and stores
   $I_1 \leftarrow IN_{1,1}^{a} \cdot IN_{1,2}^{b} = g^{SK_{U,1}}$ and $I_2 \leftarrow IN_{2,1}^{a} \cdot IN_{2,2}^{b} = g^{SK_{U,2}}$
2. So he may verify the smart card giving it as input a domain public key $PK_{domain} = g^{h}$, with known h.
3. The owner will obtain two pseudonyms nym_1 and nym_2 from the eID, and checks
   $$nym_1 \stackrel{?}{=} I_1^{h} \quad \text{and} \quad nym_2 \stackrel{?}{=} I_2^{h}$$

To sum up

1. The document owner may personalize his eID, while the final secret keys are still “certified”.

To sum up

1. The document owner may personalize his eID, while the final secret keys are still “certified”.
2. Even the issuer does not know the users secret keys and his pseudonyms.

To sum up

1. The document owner may personalize his eID, while the final secret keys are still “certified”.
2. Even the issuer does not know the users secret keys and his pseudonyms.
3. We still do not have seclusiveness - now if someone breaks one card, he may compute the system keys.

To sum up

1. The document owner may personalize his eID, while the final secret keys are still “certified”.
2. Even the issuer does not know the users secret keys and his pseudonyms.
3. We still do not have seclusiveness - now if someone breaks one card, he may compute the system keys.

What now?

- With use of bilinear maps its not a big deal.
- But we need more reliable standards...

Thank You