On Delegation of Verifiable Presentations

Andrea Flamini (University of Trento and Politecnico di Torino), Andrea Gangemi (Politecnico di Torino), **Enrico Guglielmino** (Politecnico di Torino), Vincenzo Orabona (Eustema S.p.A)

andrea.flamini@unitn.it *enrico.guglielmino@polito.it*
andrea.gangemi@polito.it *v.orabona@eustema.it*

3rd International Workshop on Trends in Digital Identity (TDI 2025)

Overview

1. First Section
   - Verifiable Credentials
   - ARF-Compliant Verifiable Credentials
   - Delegation of VPs

2. Second Section
   - Delegation of an ARF-Compliant VP
   - Security notions
   - Instantiation in EBSI and EUDI frameworks

Verifiable Credentials

Verifiable credentials (VC) are the digital analogue of physical credentials. Their security relies on the use of cryptographic tools.

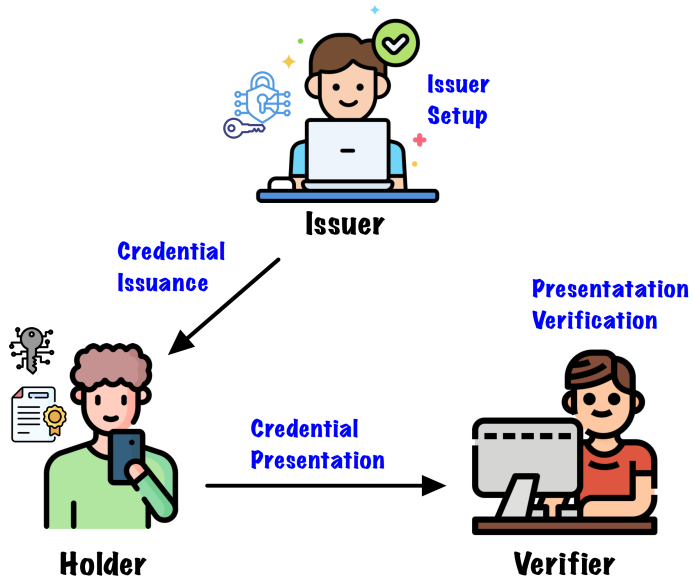


Figure: Verifiable Credentials: actors and operations

With entry into force of the eIDAS regulation european citizens will be provided a digital wallet (EUDI Wallet) storing VCs.
Their structure is described in the EUDI Architecture and Reference Framework (ARF).

ARF-Compliant Verifiable Credentials: issuance of VC

The *ARF-Compliant* verifiable credentials **issued by** $\mathsf{pk}_{\mathsf{Iss}}$ are structured as follows:

$$\mathsf{cred} = \Big(\big(\sigma, \{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}}\big) , \{a_i\}_{i\in[l]}, \{\mathsf{salt}_i\}_{i\in[l]} , \mathsf{sk}_{\mathsf{cred}} \Big)$$

where

- $\sigma \xleftarrow{\$} \mathsf{Sign}((\{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}}), \mathsf{sk}_{\mathsf{Iss}})$
- $\mathsf{com}_i \leftarrow H(a_i || \mathsf{salt}_i) \forall i \in [l]$

What is Selective Disclosure?

Selective disclosure allows revealing only specific attributes from a verifiable credential, ensuring privacy by not exposing unnecessary data.

ARF-Compliant Verifiable Credentials: presentation of VC

How can the holder generate a verifiable presentation (VP) for $\{a_i\}_{i \in \mathsf{Rev}}, \mathsf{Rev} \subseteq [l]$ from its VC $\mathsf{cred} = ((\sigma, \{\mathsf{com}_i\}_{i \in [l]}, \mathsf{pk}_{\mathsf{cred}}) , \{a_i\}_{i \in [l]}, \{\mathsf{salt}_i\}_{i \in [l]} , \mathsf{sk}_{\mathsf{cred}})$?

Open the commitments $\{\mathsf{com}_i\}_{i \in \mathsf{Rev}}$ revealing $\{\mathsf{salt}_i\}_{i \in \mathsf{Rev}}, \{a_i\}_{i \in \mathsf{Rev}}$.

$$\mathsf{pres} = (\underbrace{((\sigma, \{\mathsf{com}_i\}_{i \in [l]}, \mathsf{pk}_{\mathsf{cred}}) , \{\mathsf{salt}_i\}_{i \in \mathsf{Rev}}, \{a_i\}_{i \in \mathsf{Rev}} , \mathsf{nonce})}_{\mathsf{pres}'}, \sigma')$$

Where $\sigma' \xleftarrow{\$} \mathsf{Sign}(\mathsf{pres}', \mathsf{sk}_{\mathsf{cred}})$;

ARF-Compliant Verifiable Credentials: verification of VP

How to verify

$$\text{pres} = (\underbrace{(\ (\sigma, \{\text{com}_i\}_{i\in[l]}, \text{pk}_{\text{cred}})\ , \{\text{salt}_i\}_{i\in\text{Rev}}, \{a_i\}_{i\in\text{Rev}}, \text{nonce})}_{\text{pres}'}, \sigma')$$

The verifier performs the following checks:

- verify the signature of the issuer: $1 \leftarrow \text{Vf}(\sigma, (\{\text{com}_i\}_{i\in[l]}, \text{pk}_{\text{cred}}), \text{pk}_{\text{Iss}})$;
- verify the opening of the commitments: $\text{com}_i = H(a_i||\text{salt}_i), \forall i \in \text{Rev}$;
- verify the signature of the holder: $1 \leftarrow \text{Vf}(\sigma', \text{pres}', \text{pk}_{\text{cred}})$.

If the previous checks are satisfied, the verifier accepts and outputs 1, otherwise it outputs 0.

Delegation of VPs

As specified in the EUDI Wallet Implementation Roadmap, an important extension of VC schemes that would improve their usability is the ability to support delegation of VPs.

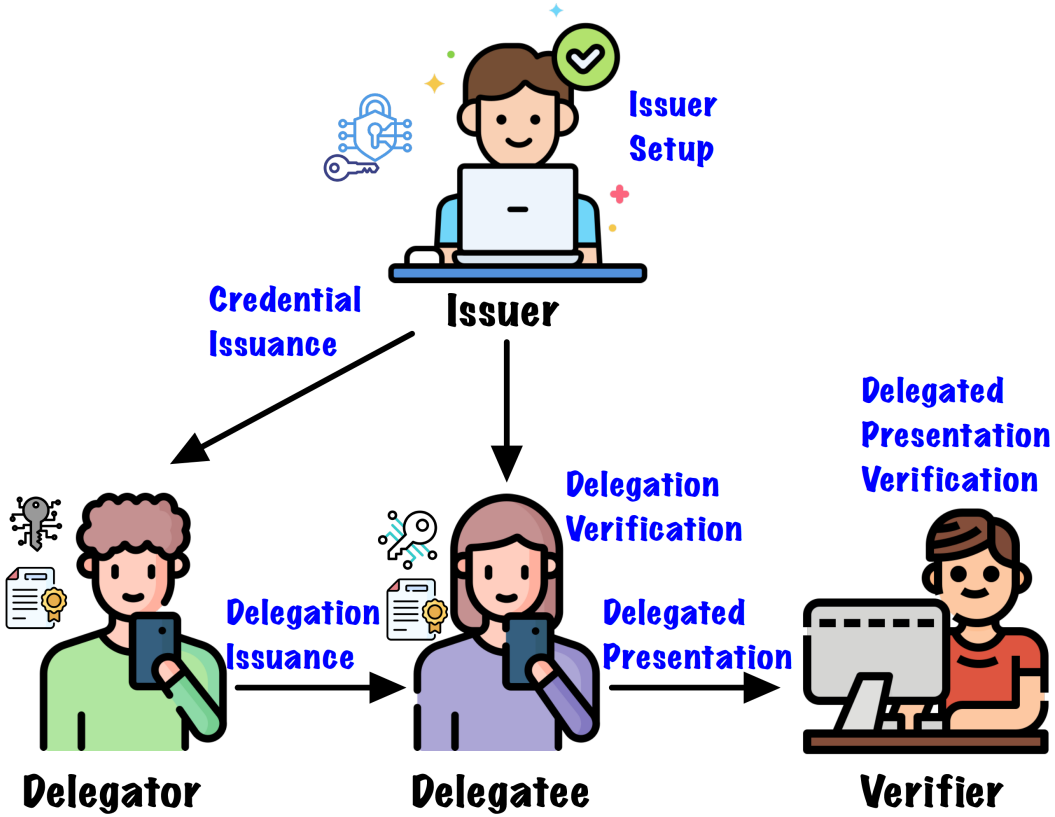


Figure: VP delegation scheme.

Use cases: pharmacy, online services, financial operations...

Delegation of VPs

What is the general structure of a delegation?

$$\mathsf{del} = (\Delta_{\mathsf{ID}}, \mathsf{scope}, \mathsf{DP}, \pi_{\mathsf{DP}})$$

where:

- Δ_{ID} is the *delegate identity*;
- scope is the *delegation scope*;
- DP is the *delegator payload*;
- π_{DP} is a proof that the delegator has a VC satisfying DP which is bound to Δ_{ID} and scope.

Delegation of VPs

Interaction framework among Delegator, Delegatee, and Verifier

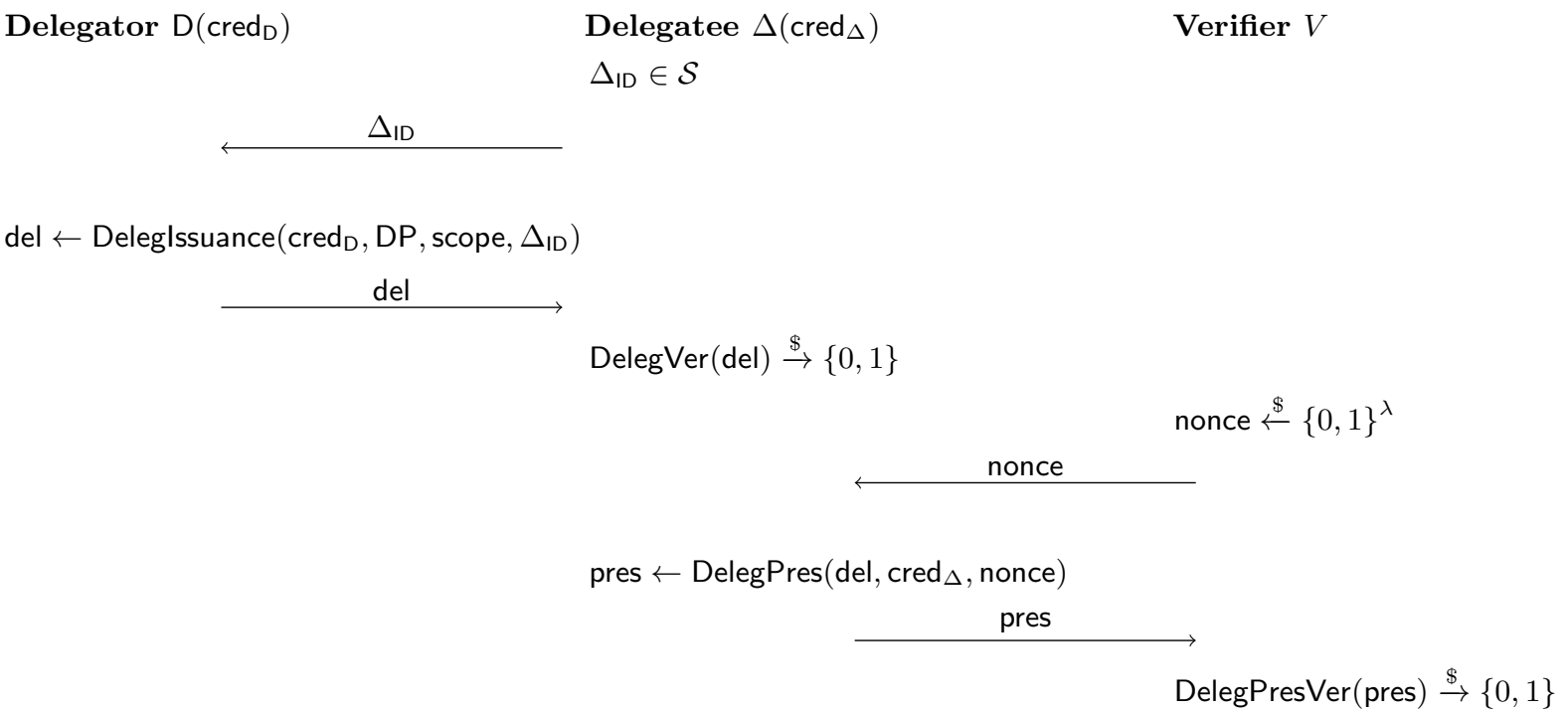


Figure: Interactions between the Delegator D, the Delegatee Δ, and the Verifier V.

Delegation issuance

$$\mathsf{DelegIssuance}(\mathsf{cred}_D, \mathsf{DP}, \mathsf{scope}, \Delta_{\mathsf{ID}}) \xrightarrow{\$} \underbrace{(\Delta_{\mathsf{ID}}, \mathsf{scope}, \mathsf{DP}, \pi_{\mathsf{DP}})}_{\mathsf{del}}$$

Given $(\Delta_{\mathsf{ID}}, \mathsf{scope}, \mathsf{DP})$, the delegator computes π_{DP} as follows:

1. $\mathsf{pres}' \leftarrow \Big(\underbrace{(\sigma, \{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}_D}), \{\mathsf{salt}_i\}_{i\in\mathsf{DP}}, \mathsf{DP}}_{\text{unsigned presentation of DP}}, \underbrace{\mathsf{scope}, \Delta_{\mathsf{ID}}}_{\text{added information}} \Big)$;
2. Signs pres': $\sigma' \xleftarrow{\$} \mathsf{Sign}(\mathsf{pres}', \mathsf{sk}_{\mathsf{cred}_D})$
3. sets $\pi_{\mathsf{DP}} \leftarrow (\mathsf{pres}', \sigma')$;

Return
$\mathsf{del} \leftarrow (\Delta_{\mathsf{ID}}, \mathsf{scope}, \mathsf{DP}, \underbrace{(\sigma', \underbrace{(\sigma, \{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}_D}), \{\mathsf{salt}_i\}_{i\in\mathsf{DP}}, \mathsf{DP}, \mathsf{scope}, \Delta_{\mathsf{ID}}}_{\mathsf{pres}'})}_{\pi_{\mathsf{DP}}})$.

Delegation of an ARF-Compliant VP

Recall: $\mathsf{del} \leftarrow (\Delta_{\mathsf{ID}}, \mathsf{scope}, \mathsf{DP}, (\sigma', \underbrace{\underbrace{(\sigma, \{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}_\mathsf{D}}), \{\mathsf{salt}_i\}_{i\in\mathsf{DP}}, \mathsf{DP}}_{\mathsf{pres}'}, \mathsf{scope}, \Delta_{\mathsf{ID}}}_{\pi_{\mathsf{DP}}}))$.

Delegation verification

$$\mathsf{DelegVer}(\mathsf{del}) \xrightarrow{\$} \{0,1\}$$

To verify the delegation, the delegatee performs the following checks:

- Verify the signature of the issuer: $1 \xleftarrow{?} \mathsf{Vf}(\sigma, (\{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}_\mathsf{D}}), \mathsf{pk}_{\mathsf{Iss}})$;
- Check that $\mathsf{com}_i = H(a_i||\mathsf{salt}_i), \forall i \in \mathsf{DP}$;
- verify the signature σ' of pres' using the public key $\mathsf{pk}_{\mathsf{cred}_\mathsf{D}}$: $1 \xleftarrow{?} \mathsf{Vf}(\sigma', \mathsf{pres}', \mathsf{pk}_{\mathsf{cred}_\mathsf{D}})$.

Delegated presentation

$$\mathsf{DelegPres}(\mathsf{del}, \mathsf{cred}_\Delta, \mathsf{nonce}) \xrightarrow{\$} \underbrace{(\mathsf{del}, \pi_{\mathsf{del}})}_{\mathsf{pres}}$$

The delegatee computes π_{del} as follows:

1. computes $\mathsf{pres}'' \leftarrow \big(\underbrace{(\sigma, \{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}_\Delta}), \{\mathsf{salt}_i\}_{i\in\Delta_{\mathsf{ID}}}, \mathsf{del}}_{\text{unsigned presentation of } \Delta_{\mathsf{ID}}}, \mathsf{nonce}\big)$;
2. signs pres'' computing $\sigma'' \xleftarrow{\$} \mathsf{Sign}(\mathsf{pres}'', \mathsf{sk}_{\mathsf{cred}_\Delta})$
3. sets $\pi_{\mathsf{del}} \leftarrow (\mathsf{pres}'', \sigma'')$

Returns $\mathsf{pres} \leftarrow (\sigma'', \big(\underbrace{(\sigma, \{\mathsf{com}_i\}_{i\in[l]}, \mathsf{pk}_{\mathsf{cred}_\Delta}), \{\mathsf{salt}_i\}_{i\in\Delta_{\mathsf{ID}}}, \mathsf{del}}_{\text{unsigned presentation of } \Delta_{\mathsf{ID}}}, \mathsf{nonce}\big))$.

Delegation of an ARF-Compliant VP

Recall: $\text{pres} \leftarrow (\sigma'', (\underbrace{(\sigma, \{\text{com}_i\}_{i\in[l]}, \text{pk}_{\text{cred}_\Delta}), \{\text{salt}_i\}_{i\in\Delta_{\text{ID}}}, \text{del}}_{\text{unsigned presentation of } \Delta_{\text{ID}}}, \text{nonce}))$.

Delegated presentation verification

$$\text{DelegPresVer}(\text{pres}) \xrightarrow{\$} \{0,1\}$$

The verifier checks that:

- the delegation is valid, i.e. $\text{DelegVer}(\text{del}) \rightarrow 1$;
- π_{del} is a valid presentation of the attributes in Δ_{ID} specified in del, i.e.:
  1. the signature σ'' of pres'' is valid using $\text{pk}_{\text{cred}_\Delta}$: $1 \leftarrow \text{Vf}(\sigma'', \text{pres}'', \text{pk}_{\text{cred}_\Delta})$;
  2. $\text{com}_i = H(a_i || \text{salt}_i)\, \forall i \in \Delta_{\text{ID}}$;
  3. the signature of the issuer is valid: $1 \leftarrow \text{Vf}(\sigma, (\{\text{com}_i\}_{i\in[l]}, \text{pk}_{\text{cred}_\Delta}), \text{pk}_{\text{Iss}})$.
- the value scope included in del is satisfied.

Correctness

Given a VP delegation scheme

$$\mathcal{VPDS} = (\mathsf{DelegIssuance}, \mathsf{DelegVer}, \mathsf{DelegPres}, \mathsf{DelegPresVer}),$$

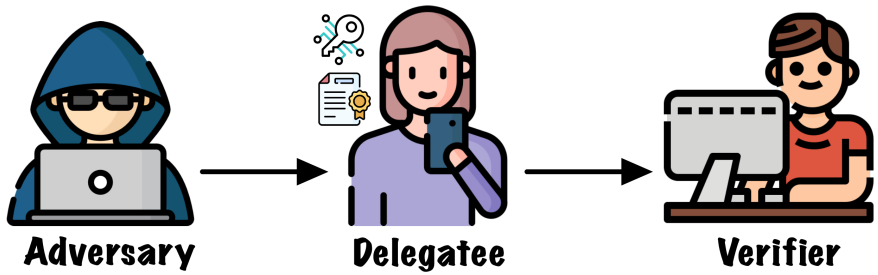
we say that the scheme is correct if $\mathsf{DelegPresVer}(\mathsf{pres}) \rightarrow 1$ whenever:

- $\mathsf{del} \xleftarrow{\$} \mathsf{DelegIssuance}(\mathsf{cred}_D, \mathsf{DP}, \mathsf{scope}, \Delta_{ID})$ where cred_D satisfies the statements contained in DP
- $\mathsf{pres} \xleftarrow{\$} \mathsf{DelegPres}(\mathsf{del}, \mathsf{cred}_\Delta, \mathsf{nonce})$, where cred_Δ satisfies the statements contained in Δ_{ID}.

Unforgeability

We consider two notions of unforgeability:

- the unforgeability of the delegation algorithm DelegIssuance



- the unforgeability of the delegation presentation algorithm DelegPres

Instantiation in EBSI and EUDI frameworks

The protocol we have described can be integrated into existing ecosystems such as EBSI or in the EUDI Wallet context without defining new data structures, only new verification procedures.

- The delegation del can be a *VC issued by the delegator* that has as attributes the components scope, Δ_{ID}, DP and π_{DP}.

Figure: Representation of delegation as a VC.

Instantiation in EBSI and EUDI frameworks

- The only modification to the verification protocol is that the verifier must check that π_{DP} is indeed a valid presentation of the statement DP and that the presentation π_{del} created by the delegate using $cred_{del}$ is a valid presentation of Δ_{ID}.
- In EBSI the only entities entitled to issue credentials are legal persons whose DID is registered in the Trusted Issuer Registry (TIR).

  If the delegator is only a physical person, a third party, registered in the TIR must create the delegation VC.

Thank you for your attention!