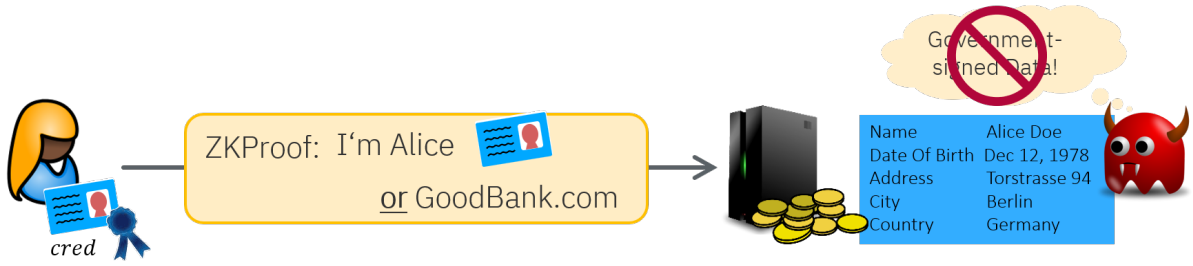
Cybersecurity – Identity Management
Prof. Dr. Anja Lehmann
Master Project | Winter Term 26/27

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

It Wasn't Me, It Was You: Deniable Authentication for the EUDI

The eIDAS 2.0 regulation requires every EU member state to provide a digital identity solution—the EU Digital Identity (EUDI) Wallet—by the end of 2026 [1]. The EU decided to build the wallet from classic ECDSA-signed credentials, where a digital identity is simply an issuer's signature on the set of user attributes. In Germany this choice drew criticism from consumer- and data-protection organizations [4,5], who argued that signed credentials lack an important privacy property*: plausible deniability*. When a user presents their digital identity, anyone can verify this presentation—and hence verifiers might be tempted to store and trade the authenticated data. This issue received attention since it is perceived as a downgrade from the prior German eID, where authentication is an online process between chip and verifier and cannot be forwarded.

This criticism, however, conflates a property of the signature scheme with a property of the presentation protocol. Recent work [2] shows that deniability is easy to add on top of ECDSA, using standard zero-knowledge techniques. The idea (sketched in more detail at the end of this document) is that the verifier holds its own key pair and that the user never hands over the issuer's signature in full. Instead, she reveals only a part of it and proves, in zero knowledge, that she *either* knows the rest of a valid issuer signature *or* the verifier's secret key. Since the verifier knows its own secret key, it could have produced this proof itself—so the presentation convinces the intended verifier but is worthless as evidence to anyone else, exactly the deniability the critics asked for.



This is a promising starting point, but it is only a first step: The construction relies on each credential being used only *once*—which, while consistent with the planned EUDI deployment, might be more restrictive than necessary—and it has not yet been integrated into real EUDI data formats. This master project takes up these two open challenges.

Open Challenge 1: Deniability Beyond the Single-Show Setting

The deniability guarantee crucially relies on every ECDSA signature being used exactly once. The reason is that a presentation reveals one part of the issuer's signature in the clear, and this part is fixed by the signature. Thus, two presentations of the same credential expose the same value. Deniability rests on the argument that the verifier could have fabricated the transcript itself, sampling this value at random. But if two verifiers hold transcripts containing the *same* value, it is no longer plausible that each produced it independently by chance. The shared value is only explained by a genuine

presentation of one underlying credential, so a third party seeing both transcripts obtains exactly the transferable evidence that deniability was meant to prevent.

The proposed deniable ECDSA construction in [2] sidesteps this by relying on single-use credentials, which users obtain through batch issuance. This is the deployment model of the EUDI wallet anyway, where single-use is intended to ensure presentation unlinkability. Unlinkability is essential in minimal-disclosure settings, such as proving that one is over 18 without revealing further attributes. Deniability, however, is mainly needed in the maximal-disclosure setting, i.e., where a user reveals all her attributes. Here unlinkability is out of reach, since the user fully identifies herself—so it would be valuable to achieve deniability without inheriting this expensive deployment setup.

The first task is to develop a stronger security model and secure constructions in which can be safely reused. This means formalising a *multi-show* notion of deniability that captures correlations across transcripts and investigating presentation protocols that satisfy it.

Open Challenge 2: Integration into the EUDI Workflow and Data Formats

The second avenue is practical. The deniable presentation of [2] must be made to fit the real-world EUDI machinery, which builds on the SD-JWT standard [3]: credentials use salted-hash commitments for selective disclosure, presentations are tied to a session through a key-binding component carrying the verifier's nonce, and the relying party authenticates with a certified key pair. The last point is convenient, as the deniable presentation requires the verifier to hold a key pair anyway, which the certified relying-party key can serve. The task is to integrate the deniable presentation into this workflow—mapping it into the SD-JWT presentation structure and reconciling it with features such as session freshness and replay protection—and to build a proof-of-concept implementation that benchmarks the added communication and computation overhead.

Your Tasks in Short

The exact scope will depend on the size and interests of the group.

- Formalise a multi-show security model for deniable signed credentials
- Design and formally analyse constructions that satisfy the requirements of the model, in particular, that allow the reuse of (ECDSA) signatures without losing deniability
- Integrate deniable presentations into the EUDI data formats and protocol flows
- Demonstrate real-world feasibility through a proof-of-concept implementation and a performance evaluation

Requirements

We expect a solid understanding and interest in cryptography and provably secure constructions. You should have successfully attended & enjoyed the “Introduction to Cryptography”, and ideally also the “Advanced Cryptography” course. In any case, you should bring the willingness and curiosity to delve into the (mathematical) details of plausible deniability, and to explore the complex and sometimes conflicting requirements of the European Digital Identity Wallet.

Sneak Peak: How To Make ECDSA Signatures Deniable

Let us assume a user holds a credential over their attributes $\boldsymbol{attr}$, which is simply an issuer’s ECDSA signature on $\boldsymbol{attr}$. Recently, [2] showed that it suffices to re-interpret ECDSA signatures through the lens of discrete-logarithm relations, then a standard zero-knowledge proof technique can be employed to achieve deniability.

Recap: ECDSA signatures. Recall that the issuer holds a key pair (sk, pk) and that for an ECDSA signature on $attr$, the issuer chooses a random scalar $k \xleftarrow{r} \mathbb{Z}_p^*$, computes $R \leftarrow G^k$ and sets r to be the x-coordinate of the elliptic curve point R. Then it computes $s \leftarrow k^{-1}(H(attr) + r \cdot sk)$ and sets the signature to the pair (r, s). The signature is verified by computing $R' \leftarrow \left(G^{H(attr)} + pk^r\right)^{s^{-1}}$ and checking if the x-coordinate of R' is r.

The key observation is that the ECDSA verification equation itself induces a discrete-logarithm relation: writing $\tilde{G} \coloneqq G^{H(attr)} \cdot pk^r$, a valid signature (r, s) satisfies the Schnorr relation $R = \left(\tilde{G}\right)^{s^{-1}}$. The holder can therefore prove knowledge of the signature with a lightweight and standard zero-knowledge-proof of knowledge of a discrete logarithm. Now we can apply the generic approach to deniability: the verifier is assumed to hold a (discrete logarithm) key pair (vsk, vpk) as well, and the user proves that she knows *either the discrete logarithm between* R *and* $\tilde{G}$*, or between* vpk *and* vsk*.* Crucially, this is a zero-knowledge proof: it convinces the recipient that one of the two statements holds, but does not reveal *which* one. This asymmetry is what makes it work: The verifier is convinced because it knows it did not fabricate the proof using its own secret key — so the only remaining explanation is that the holder knew a valid issuer signature. A third party has no such certainty: since the verifier *could* have produced the very same proof with its own secret key, and the proof hides which branch was used, the transcript is indistinguishable from one the verifier made up alone. The resulting "designated-verifier signature" therefore convinces the intended relying party but is non-transferable to anyone else. The construction costs just a few group elements on top of a standard ECDSA signature.

No Multi-Show Deniability. Unfortunately, this approach does no longer ensure plausible deniability once we allow for the same credential to be presented multiple times: the user must reveal R in a presentation, and if now multiple verifiers own transcripts that include identical R's, it is no longer plausible that each of them forged a user presentation and "by chance" sampled the same R value.

Contact

Prof. Anja Lehmann: anja.lehmann@hpi.de
Karla Friedrichs: karla.friedrichs@hpi.de
Cavit Özbay: cavit.oezbay@hpi.de

References

[1] *European Digital Identity Wallet. Architecture and Reference Framework*. https://eu-digital-identity-wallet.github.io/eudi-doc-architecture-and-reference-framework/2.0.0/

[2] Bertram, M., and Lehmann, A. *Deniability for Signed Credentials: Revisiting the Authenticated Channel vs. Signed Data Debate in the EUDI Wallet.* *https://hpi.de/fileadmin/user_upload/fachgebiete/lehmann/Publications/deniableECDSA.pdf*

[3] Fett, D., Yasuda, K., and Campbell, B. (2025). *Selective Disclosure for JSON Web Tokens*. RFC 9901. https://datatracker.ietf.org/doc/rfc9901/

[4] Verbraucherzentrale Bundesverband (vzbv): Report on the architecture of the EUDI-wallet (2024), https://www.vzbv.de/sites/default/files/2025-02/vzbv-Gutachten%20zur%20Architektur%20der%20EUDI-Wallet.pdf

[5] epicenter.works: Data Protection Analysis: eIDAS Architecture Reference Framework 1.4, https://epicenter.works/fileadmin/medienspiegel/user_upload/epicenter.works_-_ARF1.4.pdf