ETSI TR 119 476 V1.2.1 (2024-07)

Electronic Signatures and Trust Infrastructures (ESI);
Analysis of selective disclosure and zero-knowledge proofs applied to Electronic Attestation of Attributes

Reference
RTR/ESI-0019476v121

Keywords
identity, trust services

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Electronic Signatures and Trust Infrastructures (ESI).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The eIDAS2 regulation and the Architecture and Reference Framework (ARF) define regulatory requirements on selective disclosure and unlinkability for the EUDI Wallet. The present document provides a general yet comprehensive analysis of signature schemes, formats and protocols that cater for selective disclosure, unlinkability and predicates. More specifically, the present document includes an analysis of how certain selective disclosure techniques can be applied on eIDAS2 and the EUDI Wallet.

The term selective disclosure means that a user should be capable of presenting a subset of attributes from at least one, but potentially multiple, (Qualified) Electronic Attestations of Attributes ((Q)EAAs). For example, a user should be able to only present their birth date.

The term unlinkability means that different parties should not be able to connect the user's selectively disclosed attributes beyond what is disclosed. There are different categories and degrees of unlinkability, and the present report focuses both on verifier unlinkability and full unlinkability. Verifier unlinkable means that one or more verifiers cannot collude to determine if the selectively disclosed attributes describe the same identity subject, whilst fully unlinkable means that no party can collude to determine if the selectively disclosed attributes describe the same identity subject.

Predicate proofs are verifiable Boolean assertions (true or false) about attributes in a (Q)EAA without disclosing the attribute value itself. For example, a user could derive a proof that they are above the age of 20 from their birthdate and show only the proof as opposed to the birthdate itself. Predicate proofs are often employed in Zero-Knowledge Proof (ZKP) systems aimed at limiting information disclosure.

The selective disclosure signature schemes described in the present report are divided in the following categories:

- **Atomic (Q)EAA schemes**. An atomic electronic attribute attestation is a (Q)EAA with a single attribute claim, which can be issued by a (Q)TSP upon request or as part of a batch to an EUDI Wallet. The atomic (Q)EAAs can be selected by the user and be included in a verifiable presentation that is presented to a verifier.

- **Multi-message signature schemes**. The category of multi-message signature schemes has the capability of proving knowledge of a signature while selectively disclosing any subset of the signed messages. The following schemes in this category are described: BBS/BBS+, Camenisch-Lysyanskaya (CL) signatures, Mercurial signatures, and Pointcheval-Sanders Multi-Signatures (PS-MS). ISO/IEC have standardized parts of BBS and PS-MS in ISO/IEC 20008 [i.143], and have taken the initiative to standardize BBS+ and PS-MS in ISO/IEC PWI 24843 [i.144] and ISO/IEC CD 27565 [i.150]. Furthermore, there are cryptographic research projects, such as MoniPoly, where undisclosed attributes have no impact on the proof size.

- **Salted attribute hashes**. The general concept of this category is to combine each attribute with a salt, hash the combined values, and insert the resulting salted attribute hashes in a list that is signed. The user presents a selection of attributes to the verifier, which can validate them against the list of salted attribute hashes. The following schemes, based on salted attribute hashes, are described: HashWires, Authentic Chained Data Containers (ACDC), and Gordian Envelopes.

- **Proofs for arithmetic circuits (programmable ZKPs)**. This category of ZKP protocols enable the user to prove to the verifier that a certain statement is true, without revealing any additional information beyond the truth of the statement itself. The discussion of proofs for arithmetic circuits is focused on zk-SNARKs.

The present document also includes descriptions of (Q)EAA formats that can be used with selective disclosure. The (Q)EAA formats are divided in the following categories:

- **Atomic (Q)EAA formats**. These (Q)EAA formats are based on the category of atomic (Q)EAA formats. The following (Q)EAA formats in this category are described: PKIX X.509 attribute certificate with atomic attribute and W3C Verifiable Credential with atomic attribute.

- **Multi-message signature (Q)EAA formats**. This category of (Q)EAA formats is based on the multi-message signature schemes. Mainly W3C and Hyperledger have specified such formats to be used for privacy preserving features. The following (Q)EAA formats in this category are described: W3C VC Data Model with ZKP, W3C VC Data Integrity with BBS Cryptosuite, W3C Data Integrity ECDSA Cryptosuites v1.0, and Hyperledger AnonCreds (format).

- **(Q)EAAs with salted attribute hashes**. This category of (Q)EAA formats is based on the concept of salted attribute hashes. These (Q)EAA formats specify in detail how the attributes are combined with the random salts and hashed, inserted in a list, which is signed. The following (Q)EAA formats of this category are described: IETF SD-JWT and ISO/IEC 18013-5 [i.140] Mobile Security Object (MSO).

- **JSON container formats**. This category of generic JSON container formats allows for combining and presenting a mix of selective disclosure signature schemes. The following JSON container formats are described: IETF JSON WebProof (JWP) and W3C JSON Web Proofs For Binary Merkle Trees.

Furthermore, the present document describes systems and protocols with selective disclosure capabilities. The systems and protocols are divided in the following categories:

- **Atomic attribute (Q)EAA presentation protocols**. This category of protocols is designed to present the atomic attribute (Q)EAA formats. The atomic attribute (Q)EAAs may be issued on demand to the user, upon request by a verifier. The following protocols in this category are described: PKIX X.509 attribute certificates with single attributes and VC-FIDO for atomic (Q)EAAs.

- **Multi-message signature protocols and solutions**. This category of protocols is based on the multi-message signature schemes, such as BBS+ and CL-signatures, and are used to present selected attributes of the (Q)EAAs. The following protocols and solutions in this category are described: Hyperledger AnonCreds (protocols) and Direct Anonymous Attestation (DAA) used with Trusted Platform Modules (TPMs); the TPMs have been deployed in personal computers at a large scale.

- **Salted attribute hashes protocols**. These solutions and protocols are designed to present selectively disclosed attributes based on salted attribute hashes. The OpenAttestation solution of Singapore's Smart Nation is described in the present report. Furthermore, ISO mDL MSOs can be shared over the proximity protocols described in ISO/IEC 18013-5 [i.140] or over the Internet by using ISO/IEC 23220-4 [i.146]. The SD-JWTs can be presented with different protocols, such as OID4VP (OpenID for Verifiable Presentations), ISO 18013-7 [i.141] or ISO/IEC 23220-4 [i.146].

- **Solutions based on proofs for arithmetic circuits (programmable ZKPs)**. The solutions that are based on proofs for arithmetic circuits intend to use ZKP schemes such as zk-SNARK to facilitate data-minimizing verifiable presentations based on existing digital identity infrastructures. In particular, they can provide selective disclosure, unlinkability, and predicates. The projects Cinderella (zk-SNARKs used with X.509 certificates) and zk-creds (zk-SNARKs used with ICAO passports) are described in the present document.

- **Anonymous attribute based credentials systems**. These solutions are implementations of existing multi-message signature schemes such as CL-signatures or BBS+, with the purpose to present anonymous credentials ((Q)EAAs) to a verifier. The following solutions in this category are described: Idemix (Identity Mixer), U-Prove, ISO/IEC 18370 [i.142] (blind digital signatures), and Keyed-Verification Anonymous Credentials (KVAC).

- **ISO mobile driving license (ISO mDL)**. The ISO mDL standard specifies various flows for selective disclosure of attributes. In the present document, the following ISO mDL flows are described: ISO/IEC 18013-5 [i.140] (device retrieval flow), ISO/IEC 18013-5 [i.140] (server retrieval flows), ISO/IEC 18013-7 [i.141] (unattended flow) and ISO/IEC 23220-4 [i.146] (operational protocols).

The ARF proposes two protection mechanisms for the PID, which support selective disclosure but not unlinkability (unless batch issued):

- ISO/IEC 18013-5 [i.140] (ISO mDL). The ISO mDL mdoc contains all attributes of a user, whilst the ISO mDL MSO contains the corresponding hashed salted attributes.

- A JWT encoding of the W3C Verifiable Credentials Data Model v1.1 in conjunction with IETF SD-JWT. The JWT contains the user attributes, whilst the SD-JWT contains the corresponding hashed salted attributes.

The present document includes an extensive analysis of ISO mDL MSO and SD-JWT and how the formats comply with the eIDAS2 requirements on selective disclosure and unlinkability.

The ISO mDL MSO and the SD-JWT formats, and related presentation protocols, cater for selective disclosure based on the concept of salted attribute hashes. Furthermore, the MSO and SD-JWT formats support SOG-IS approved cryptographic algorithms and can also be used with quantum-safe cryptography for future use. The conclusion is thus that MSO and SD-JWT meet the eIDAS2 regulatory and technical requirements on selective disclosure.

As stated, ISO mDL MSO and SD-JWT are not fully unlinkable, although they can provide verifier unlinkability with certain operational measures. In order to achieve verifier unlinkability, batches of ISO mDL MSOs or SD-JWTs need to be issued to each EUDI Wallet. The random salts in the ISO mDL MSO and SD-JWT should be unique, meaning that refreshed MSOs and SD-JWTs are presented to a relying party. Furthermore, the user public keys used for holder binding, if present, need to be unique too.

There are many similarities between the ISO mDL issuers and the eIDAS2 compliant PID Providers (PIDPs) or QTSPs. The PIDPs/QTSPs can issue PIDs/(Q)EAAs to EUDI Wallets as follows to cater for selective disclosure:

- The PIDP/QTSP issues ISO mDL mdoc and/or JWT as PID/(Q)EAAs to the EUDI Wallet.

- The PIDP/QTSP issues ISO mDL MSOs and/or SD-JWTs batchwise to the EUDI Wallet. The ISO mDL MSOs are associated with the ISO mDL mdoc, and the SD-JWTs with the JWT. Random salts are used for the hashed salted attributes in each MSO or SD-JWT. This will cater for verifier unlinkability when the MSOs or SD-JWTs are presented to and validated by a relying party.

- The EUDI Wallet selectively discloses certain attribute(s) of an ISO mDL mdoc or JWT. One ISO mDL MSO or SD-JWT is selected from the batch in the EUDI Wallet, and is associated with the disclosed attribute(s).

- The relying party can use the eIDAS2 trust list (which is equivalent to an ISO mDL VICAL) to retrieve the QTSP/PIDP trust anchor (which is equivalent to the IACA trust anchor). The relying party validates the MSOs or SD-JWTs signatures by using the QTSP/PIDP trust anchor. The relying party also verifies that the presented selected attribute hash is present in the MSO or SD-JWT.

These recommendations could be considered for the upcoming ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81] that will standardize the issuance policies and profiles of (Q)EAAs.

Multi-message signature schemes such as BBS+, Camenisch-Lysyanskaya (CL) signatures, Mercurial signatures, and Pointcheval-Sanders Multi-Signatures (PS-MS) cater for full unlinkability, although they are not yet fully standardized. Hence, ISO/IEC PWI 24843 intends to standardize BBS+ and PS-MS with blinded signatures, which may allow for a future standard that could be used in compliance with the EUDI Wallet requirements on selective disclosure and unlinkability in eIDAS2.

There are also systems based on programmable ZKPs in the form of zk-SNARKs, such as Cinderella and zk-creds, that can achieve both selective disclosure and unlinkability with existing digital identity infrastructures such as X.509 certificates or ICAO passports. Such systems can generate pseudo-certificates that share selected attributes from the (Q)EAAs and attest holder binding and non-revocation without exposing linkable cryptographic identifiers. In contrast to multi-signature schemes, anonymous credentials based on programmable ZKPs can be made compatible with deployed secure hardware and are easily extendable. However, these projects are still in the research phase. Still, they may be considered for the EUDI Wallet and eIDAS2 relying parties.

Furthermore, there are recommendations on how to store such (Q)EAA formats in the EUDI Wallet, and how to present selectively disclosed attributes to eIDAS2 relying parties. These recommendations can be considered for the upcoming ETSI TS 119 462 [i.79] on EUDI Wallet interfaces.

The present document also analyses the privacy aspects of revocation schemes and validity status checks. In order to achieve privacy preserving features for revocation and validity status checks it is recommended to use OCSP in Must-Staple mode, implement Revocation Lists or validity Status Lists with additional privacy techniques such as Private Information Retrieval or Private Set Intersection, and use cryptographic accumulators where possible given the associated complexity. If programmable ZKP schemes (such as zk-SNARKs) are combined with existing credentials (such as X.509), the status validity checks are performed at the EUDI Wallet, and only the relevant information (revocation state) without any linkable cryptographic identifiers is disclosed with the verifier.

The present document also includes an analysis of post-quantum computing attacks on cryptographic schemes with selective disclosure capabilities. More specifically, the hashed salted attributes formats, such as ISO mDL MSO and SD-JWT, can be signed with post-quantum safe cryptographic algorithms. Also the atomic (Q)EAA formats can be secured with post-quantum safe signatures. The multi-message signature schemes, such as BBS+ and CL-signatures, have the following characteristics in a post-quantum world: an attacker can use a quantum computer to reveal the signer's private key from the public key and thereafter forge proofs and signatures, but an attacker will not be able to break data confidentiality, meaning that undisclosed messages are safe in a post-quantum world, as are undisclosed signature values. As regards to the programmable ZKP schemes, it depends on the design of the arithmetic circuit proof if it is post-quantum safe or not, meaning that there are zk-SNARKs that are post-quantum safe whilst others are not.

Finally, there is an annex with research projects about innovative ZKP schemes. One such approach is to design cryptographic ZKP schemes based on quantum physics. Quantum Key Distribution (QKD), quantum physics applied to the graph 3-colouring ZKP scheme, and ZKP using the quantum Internet (based on Schnorr's algorithm) are described in the annex. The ZKP schemes based on quantum physics are still in the research phase, but may be considered for the future. There are also cryptographic research initiatives on post-quantum safe (lattice-based) anonymous credentials, which cater for privacy-preserving signature schemes. The most recent research in this field is related to efficient anonymous credentials that are post-quantum safe, yet with small signature sizes.

Introduction

A historical perspective

To facilitate an understanding of the concepts in the present document, the present clause begins with a brief account of the history of selective disclosure and Zero-Knowledge Proofs (ZKPs), the problems they were introduced to address, their applications, and their potential uses in electronic attestations of attributes. The present document also discusses related concepts where required.

Cryptographic schemes for selective disclosure, unlinkability, blinded signatures, Zero-Knowledge Proofs (ZKPs), predicates and range proofs have been researched and developed since the 1980s. The first ZKP scheme was published in a paper 1985 [i.97] by the researchers Shafi Goldwasser, Silvio Micali, and Charles Rakoff. The abstract of this paper defines ZKP as: "*Zero-Knowledge Proofs are defined as those proofs that convey no additional knowledge other than the correctness of the proposition to the question*".

The present document on selective disclosure can be linked to the broader work on signatures that allow for updates to the signed document. In their 1994 paper "Incremental Cryptography: The Case of Hashing and Signing" [i.14], Bellare, Goldreich, and Goldwasser investigate cryptographic transformations where the updates to the results are proportional to the amount of modification done. Using digital signatures as a case, the authors propose the idea of updating the signature upon modification of the underlying message in a way that is proportional to the amount of change in the message (as opposed to simply signing the new message). The authors called for future work to explore various operations, such as delete and update, that could be supported by incremental signatures.

It is important to note that ZKP is not a selective disclosure scheme in and of itself, but rather a property of a proof system. Goldwasser, Micali and Rakoff (1985) defined ZKP [i.97] as "*those proofs that convey no additional knowledge other than the correctness of the proposition to the question*". Thus, ZKP is not limited to selective disclosures or signatures proofs in the context of electronic attestations of attributes. On the contrary, Brassard et al. demonstrated in their paper "Minimum disclosure proofs of knowledge" [i.32] that everything that has a proof also has a ZKP version of that proof.

Put differently, every selective disclosure related proof has a ZKP version of that proof. But it is incorrect to state that every selective disclosure scheme is done using ZKP, or that every ZKP is used for selective disclosure. ZKPs matter because usually, in digital identification, holders share substantially more information than the verifier asks for, e.g. superfluous identity attributes, unique cryptographic information (signatures, public keys, revocation IDs). Using a ZKP, the holder only proves what the verifier wants to know (precisely the required identity attributes, i.e. selective disclosure; that the attributes are signed by the issuer without revealing the linkable digital signature (unlinkability), that an attribute has a required property without sharing it (predicates such as range proofs). As such, ZKPs can be considered as facilitating the perfect implementation of the data minimization principle.

Electronic attestations of attributes represent a context in which several features, such as selective disclosure or proofs about knowledge of states like a valid signature value, have been implemented with the ZKP property. Among the earliest work here was done by Feige, Fiat, and Shamir (1987) who demonstrated how ZKP can be used in identification schemes by a user demonstrating knowledge as opposed to prove the validity of assertions. Since then, ZKP has been widely deployed in many of the privacy focused selective disclosure capable electronic attestation of attribute solutions.

Another pioneer in the field of ZKP was the American cryptographer David Chaum who published the scientific paper Blind Signatures for Untraceable Payments [i.53] in 1982, which described anonymized digital money (DigiCash) for the first time. The concept of Blind Signatures was designed to ensure complete privacy of users who wanted to conduct online transactions.

In 2002, Steinfeld, Bull, and Zheng published their paper "Content Extraction Signatures" (CES) [i.190]. In it, the authors present a way to perform the delete operation without knowledge of the signer's private key. The authors argue that this would allow a user "to disclose only certain parts of a document" as opposed to "forcing the document holder to disclose all of its contents to a third party for the signature to be verifiable". The authors then go on to present the idea of context extraction, i.e. "the extraction of certain selected portions of a signed document" in cases where a user "does not wish to pass on the whole document to a third (verifying) party". Their method is based on signing digests of data subsets. Relatedly, Johnson et al. (2002) presented their work on redactable signatures, which are conceptually very similar to CES. In fact, the proposed schemes in the papers overlap, together detailing four different schemes for CES. Two of these rely on commitment vectors, and two on the homomorphic properties and batching of RSA respectively.

Brands (2002) directly applies these concepts to electronic attestations of attributes. In his 2002 paper "A Technical Overview of Digital Credentials" [i.30] Brands discusses the "selective disclosure properties of data fields" in digital credentials. In that paper, Brands presents the idea to "hash attributes [...] using a collision-intractable hash function; to disclose these attributes, Alice discloses the preimages of the corresponding [attributes]". Interestingly, Brands proposed design also relies on a proof of knowledge of the digital signature, which is among the first references to the use of ZKP for enhancing privacy when presenting electronic attestations of attributes. Brands' paper is also among the earliest work on the use of predicates in electronic attestations of attributes. In essence, Brands' work was based on commitment vectors and the algebraic manipulations (e.g. addition and multiplication) of these commitments, allowing proofs containing AND, OR, and NOT connectives between attributes and for a single attribute.

The above mentioned work laid the groundwork for the concept of selective disclosure and unlinkability. Ongoing work presented workarounds to discovered vulnerabilities in some of the proposed schemes, and introduced more advanced features that further improved privacy e.g. by enabling multi-show unlinkable selective disclosures (defined in clause 3.1 and for additional details see "Anonymous Credentials" [i.41] by Camenisch and Lysyanskaya in 2003). Notable early examples of implementations of this work focused on enhanced privacy include AnonCreds and Idemix (both based on Camenisch-Lysyanskaya signatures as detailed herein under clause 4), as well as U-Prove (based on Brands' work). A more recent example of a multi-message signature scheme capable of selective disclosure is the BBS+ signature scheme (detailed in clause 4.3 and is based on group signatures and the work of Boneh, Boyen, and Shacham, 2004). However, as noted in Camenisch et al. (2013) [i.41], real-world deployments of cryptographic primitives, schemes and protocols in electronic attestations of attributes have been slow due to them being hard to understand and "very difficult to use" as they often require advanced cryptography and the combination of several protocols to achieve the desired privacy goals. In a survey, Asghar (2011) [i.9] lists some of these often employed mechanisms, including blind signatures (Chaum, 1983), ZKPs (Goldwasser, Micali, and Rakoff, 1985), group signatures, commitment schemes (formalized in Brassard, Chaum, and Crépeau, 1988 [i.32]), and multi-message signing; which often need to be employed in tandem to reach privacy goals important for selective disclosure including multi-show unlinkability, blinding, and the ability to present a subset of the signed attestation.

In contrast to the focus on increasing privacy, others sought more performant schemes with lower but still acceptable levels of privacy. A notable example here is the early work of Bull, Stanski, and Squire (2003) [i.35], who presented a way to "enable selective disclosure of verifiable content" using a randomized salt to blind the attribute disclosures, using an identifier for each disclosable attribute, and the principle of signing the hash digests of attributes. To disclose the desired attributes, a user would simply present a subset of the attestation to the verifier, together with the attributes and salts to disclose. Variations of this salted hash digest based approach is used both in the ISO/IEC 18013-5:2001 [i.140] standard and in the IEFT SD-JWT specifications. Note that these techniques do not achieve the same levels of privacy as their more advanced counterparts (e.g. U-Prove, AnonCreds, Idemix, and BBS+) because they lack unlinkability and support for selected predicates, but they are easier to use and more performant.

The academic research of cryptographic schemes for selective disclosure, unlinkability, and predicates have continued from the mid 2010s until present day: Bulletproofs [i.36] and Pointcheval-Sanders Multi-Signatures [i.176] provide range proofs over committed values, whilst zk-SNARKs (clause 4.5.2) are advanced protocols for fully programmable ZKPs. More information about those cryptographic schemes is described in clause 4 of the present document.

The Internet standardization organizations Hyperledger, IETF and W3C® have followed the academic cryptographic research by creating Internet standards for selective disclosure, unlinkability, and predicates. Hyperledger has specified AnonCreds [i.104]. IETF has specified the BBS Signature Scheme [i.116], JSON WebProofs [i.120], PKIX attribute certificates [i.125], and SD-JWT [i.123]. W3C has specified BBS Cryptosuite and the Verifiable Credentials Data Model describes ZKPs [i.209]. Furthermore, ISO/IEC 18013-5 [i.140] specifies selective disclosure for the mobile driving license by introducing the Mobile Security Object (MSO) for the device retrieval use case. Clauses 5 and 6 in the present document describe the mentioned standards in more detail.

Overview and use cases

An overview of various use cases is provided in Figure 1 to illustrate the concepts of selective disclosure, unlinkability, and predicates.

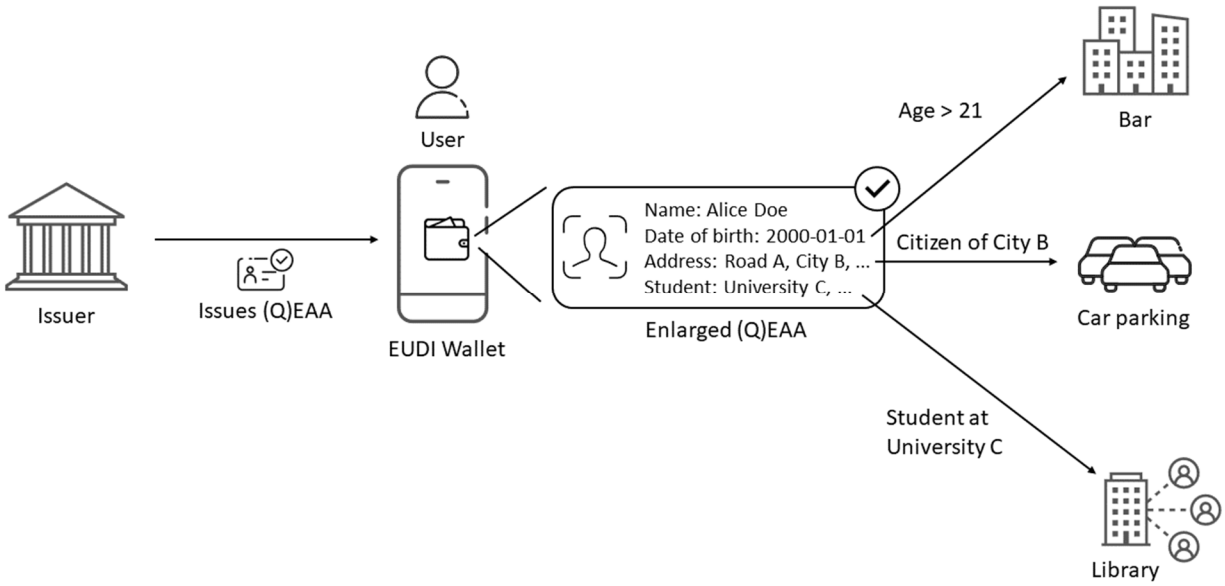

Figure 1: Overview of selective disclosure

First, an issuer creates and issues a (Qualified) Electronic Attestation of Attribute (EAA) (denoted as (Q)EAA) to a user, whereupon the (Q)EAA is stored in the user's EUDI Wallet.

EXAMPLE 1: The (Q)EAA contains the attributes name (first name and last name), date of birth, address (street, city, zip code, etc.), and student information (university, exams, course, etc.).

NOTE 1: The issuer may also issue a Person Identification Data (PID) with the same attributes, but a (Q)EAA is used for readability in this particular example.

The (Q)EAA that is stored in the user's EUDI Wallet is also associated with cryptographic keys that are necessary for the cryptographic scheme's selective disclosure capabilities. In order to access the private keys, the user needs to authenticate with PIN-code or biometrics. Clauses 6.3 and 6.5.3 in the Architecture and Reference Framework (ARF) [i.59] provide more information on the EUDI Wallet security architecture and the supported cryptographic keys management systems.

Now, the user can use its EUDI Wallet to present selected attributes of the (Q)EAA to various relying parties. A user may present multiple attributes to each verifier and is not limited to present only a single attribute claim. The user may also be able to create a presentation that includes claims from at least two (Q)EAAs even if these are issued by different issuers (herein referred to as combined presentation).

When going to a bar, for example, the user may only present a proof that she is over the age of 21 years.

NOTE 2: This is an example of a selective disclosure in combination with a predicate proof. The EUDI Wallet contains the user's actual date of birth (2000-01-01), but the EUDI Wallet only presents a proof that $21 \leq age$.

NOTE 3: This example can also be achieved using selective disclosure of a single attribute. The EUDI Wallet could contain an attestation with the key value pair `"age_over_21": "True"`. This is much simpler from a technical perspective but less flexible.

When parking the car in City B, the user may present a proof that she is a citizen of City B in order to get a discount when paying for the parking ticket. Unlinkability here helps prevent behavioural profiling and the user presents only a proof of knowledge of the undisclosed issuer's signature (the signature is linkable data).

NOTE 4: This can be achieved using a ZKP. The EUDI Wallet only presents a ZKP of knowledge of a valid signature without disclosing said signature. Analogously, a proof of holder binding without revealing the holder's linkable public key may be needed, which can also be given with a ZKP.

When borrowing a book at the university library, the user may only present that she is taking Course D at University C to prove that she is eligible to borrow the course literature.

NOTE 5: This is an example of selective disclosure of a single attribute. The EUDI Wallet contains detailed student information (university, degrees, courses, etc.), but the EUDI Wallet only presents the single claim that user studies at University C.

The concept of verifier unlinkability relates to the amount of additional information that colluding verifiers can discover about the user. High unlinkability means that the colluding verifiers learn little in addition to what the user disclosed to each verifier. Similarly, a single verifier cannot collect multiple selectively disclosed attributes and link them to the same user beyond what is possible solely based on the disclosed attribute values. This requires removing correlatable data (such as the signature) in the presentation to each verifier.

EXAMPLE 2: If presentations are unlinkable, then the bar (who knows that the user is over 21 years) cannot cooperate with the car parking (who knows that the user lives in City B) to link the user's age to the citizenship.

EXAMPLE 3: If presentations are unlinkable, then the user may visit the university library multiple times and present proofs of different courses (Course D, Course E, etc.) over time. The university library cannot link the different courses to the same user.

The concept of issuer unlinkability means that the issuer cannot collude with one or more verifiers to discover where the user is using the issued (Q)EAA. Most ZKP-based systems discussed in the present report provide full unlinkability, i.e. verifier unlinkability and issuer unlinkability.

Descriptions of selective disclosure and unlinkability

The preceding text introduced the terms 'selective disclosure' and 'unlinkability' without providing precise definitions. These terms often have varied interpretations, and these interpretations significantly influence the choice of an appropriate privacy preserving technique. Despite their apparent similarity, selective disclosure and unlinkability are distinct concepts, and their relationship to privacy is complex:

- Selective disclosure involves revealing specific attributes, or claims about these attributes, from a larger dataset. Selective disclosure, on its own, does not guarantee the highest privacy guarantees but may be a key part of a privacy preserving solution.
- Unlinkability relates to the difficulty or cost of linking multiple electronic attestation of attribute presentations. Unlinkability does not inherently ensure privacy but can be a vital element thereof.

Furthermore, the two concepts (selective disclosure and unlinkability) are not binary; they exist on a spectrum or scale, where various degrees or levels exist. And different privacy-preserving techniques are required at different degrees or levels. For selective disclosure, it is possible to understand these levels through a set of requirements:

1) The ability to selectively disclose a minimum of one attribute from a single (Q)EAA.
2) The ability to selectively disclose a minimum of two attributes from at least two distinct (Q)EAAs, with at least one attribute from each (Q)EAA. This ability is sometimes referred to as 'combined presentation'.
3) The user can disclose statements about an attribute rather than the attribute itself. This ability is sometimes referred to as predicate support.

Note that the attributes disclosed do not necessarily have to describe the identity subject. For instance, a disclosure can disclose the EAA type to reveal only that the user has a certain attestation (e.g. passport) without revealing any attribute about the identity subject. Furthermore, the above three requirements relate to other requirements to ensure important capabilities like holder binding (e.g. the verifier has to be assured that the: a) presented attributes cannot be combined in ways that make them appear to be part of another EAA than they originally were, b) presented attributes describe the same identity subject, and c) identity subject is the same entity as is presenting the attributes) and unlinkability.

Relatedly, unlinkability can be understood through a set of requirements. The general requirement relates to the ability to determine whether at least two EAA presentations describe the same identity subject. More precisely, presentations (p1, p2) are unlinkable if a set of entities cannot decide, with a non-negligible probability better than pure guessing based on the presentations and attributes received, whether the two presentations describe the same identity subject. The following cases are possible as unlinkability criteria:

1) The set is a single verifier who seeks to learn whether the attributes describe the same identity subject.

2) The set consists of at least two colluding verifiers who share the respective presentations they received in order to determine whether the attributes describe the same identity subject.

3) The set consists of signers (issuers) and verifiers, who share information to determine if the attributes describe the same identity subject.

4) The set consists of signers, verifiers, or any other party, who share information to determine if the attributes describe the same identity subject.

Throughout the rest of the present report, criteria 1 and 2 above will be combined and referred to as verifier unlinkable, whilst criteria 3 and 4 will be combined and referred to as fully unlinkable.

Neither the requirements for selective disclosure nor unlinkability are exhaustive; they are meant to clarify the non-binary nature of these concepts. What matters is the extent to which the technical solutions and formats presented in the present report can fulfil some or all of the above requirements.

Furthermore, the relationship between selective disclosure, unlinkability, and privacy is not straight forward. It is incorrect to assume that a (Q)EAA capable of selective disclosure also has to be privacy preserving. Similarly, it is not necessarily so that a (Q)EAA with unlinkability features guarantees that the privacy is preserved. If the verifier requires certain information for business or regulatory reasons, privacy may not be possible but minimizing the amount of information conveyed by the user may still be desirable to technically maximize privacy within the boundaries of the use case. Consider the following examples:

EXAMPLE 4: A user discloses that they are below the age of 65, and that they have a tertiary education. The verifier is able to determine that these two attributes describe the same identity subject. The user's privacy is still protected because the verifier does not have enough information to learn the user's identity (roughly 32 % of citizens aged 25-74 years in the EU have a tertiary education).

EXAMPLE 5: A user discloses that their first name is Peter, that they live in Sweden, and that they are below the age of 21 in three separate presentations. Each attribute roughly represents 10 million possible entities. If any party is able to learn that these three attributes represent the same identity subject (i.e. is able to link them) they can narrow down the candidates to about 300. Unlinkability here is crucial to prevent a subset of attributes from becoming personally identifying.

EXAMPLE 6: A doctor books a physical meeting with a patient, and when the patient arrives, they selectively disclose only the meeting time and meeting location. The user did not reveal any identifying or linkable information. The verifier can still easily identify the patient through the context of the presentation.

EXAMPLE 7: The verifier has access to user data sufficient for a behavioural profile in another context, e.g. browsing data over time. The user then presents unrelated data to the verifier that allows the verifier to quantify similarities in sequential data and thus identify the user.

These examples serve as a transition to a more insightful approach to understanding privacy beyond the capacity for selective disclosure or unlinkability. It delves into quantifying the extent to which each presentation diminishes the uncertainty surrounding the identity subject. Both selective disclosure and unlinkability can contribute to privacy, but their effectiveness depends on the extent of uncertainty reduction, which often is influenced by other factors. And it is unlikely that technical solutions alone can eliminate all such factors, especially considering the rapid evolution of behavioural profiling and identification techniques.

As established, user control and privacy are influenced by factors extending beyond the technical aspects of selective disclosure, unlinkability, or even predicates. Nonetheless, it is the legal text that guides the choice of privacy-preserving techniques and when and how selective disclosure and unlinkability will be supported.

Legal definitions in eIDAS2 about selective disclosure, unlinkability, and ZKP

The provisional agreement on the amending Regulation (EU) No 910/2014 (hereafter called eIDAS2) [i.86] mandates support for privacy in Recital 15 and article 5a.4(a) and provides the following definition of selective disclosure in recital 59:

"Selective disclosure is a concept empowering the owner of data to disclose only certain parts of a larger data set, in order for the receiving entity to obtain only such information as is necessary for the provision of a service requested by a user. The European Digital Identity Wallet should technically enable the selective disclosure of attributes to relying parties. It should be technically possible for the user to selectively disclose attributes, including from multiple, distinct electronic attestations, and to combine and present them seamlessly to relying parties. This feature should become a basic design feature of European Digital Identity Wallets, thereby reinforcing convenience and the protection of personal data, including data minimisation."

The definition in eIDAS2 recital 59 clarifies that disclosed information may come from multiple distinct electronic attestations of attributes, similar to the second selective disclosure requirement for combined presentations. This scenario requires additional considerations related to holder binding and proper pairing of attributes as compared to single attestation disclosures.

Moreover, the definition specifies the ability to disclose a subset of a larger data set as disclosing only such information that is necessary for the provision of a service. It is possible to interpret this clarification as a requirement that users are able to assert and prove statements about their attributes without disclosing the actual attribute data. This interpretation is aligned with Recital 14 [i.86] that states that "*cryptographic methods should allow a relying party to validate that a given statement based on the person's identification data and attestation of attributes is true, without revealing any data this statement is based on*". If this interpretation holds true, it aligns with the concept of the third selective disclosure requirement concerning predicate support. One method for implementing predicate support is through the utilization of ZKP-capable attestations, although alternatives exist. ZKPs could also be used to prove the equality (a predicate) of highly linkable identity attributes (e.g. name and date of birth or a cryptographic public key) from different attestations without revealing the identity attributes, thus increasing holder binding guarantees without reducing privacy.

Relatedly, eIDAS2 article 5a.16 lists the requirements related to unlinkability as follows:

"The technical framework of the European Digital Identity Wallet shall:

(a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user;

(b) enable privacy preserving techniques which ensure unlinkability, where the attestation of attributes does not require the identification of the user."

This article elucidates the specific entities whose linking efforts the solution aims to make more difficult. Note how (a) encompasses all parties, including issuers, verifiers, and third parties. Note also how and when the requirement in (b) mandates privacy preserving techniques to ensure unlinkability. Together, (a) and (b) seemingly correspond to either the third or fourth unlinkability requirement, which mandates unlinkability even in cases of collusion between an issuer (who signs the attestation) and a verifier (who sees a presentation of the attestation) or any other party. No salted attribute digest based solution can satisfy this unlinkability requirement as issuers are always able to link user behaviour through the disclosure of the highly linkable issuer's digital signature.

Moreover, (b) appears to suggest that unlinkability is only obligatory when the (Q)EAA does not require user identification. One plausible interpretation is that unlinkability may not be obligatory in cases where an (Q)EAA presentation includes user identifying attributes.

It is not clear if (a) is a restriction to the acquisition of data, or if it is a requirement that the data are unlinkable. If the article is a restriction on the acquisition of data, then contractual terms that prevent data sharing may be enough even in cases where the data are linkable (e.g. using salted attribute hashes approach such as ISO mDL MSO and SD-JWT). Conversely, if the data has to be unlinkable then technical solutions are required that ensure unlinkable (Q)EAAs. This may require that issuers issue a (Q)EAA in such a way that even a coalition of colluding issuers and verifiers has no ability of linking together attribute presentations on the basis of the data shared with a greater probability than pure guessing (e.g. using signature blinding and ZKP of valid signature).

It is also possible that the legal text intended unlinkable data without fully considering its technical feasibility or the relationship between unlinkable data and privacy. For instance, consider recital 14:

"Member States should integrate different privacy-preserving technologies, such as zero knowledge proof, into the European Digital Identity Wallet. Those cryptographic methods should allow a relying party to validate whether a given statement based on the person's identification data and attestation of attributes is true, without revealing any data on which that statement is based, thereby preserving the privacy of the user."

There are two main issues with this recital and the strong focus on unlinkable data. Firstly, the recital presumes that cryptographic unlinkability can ensure privacy. Cryptographic methods can only guarantee unlinkability of the data itself, and do not guarantee anything with regards to the unlinkability of an identity subject. While unlinkability of data can be achieved using cryptographic operations, the unlinkability of the identity subject requires that the user's presentation is devoid of any information (contextual or auxiliary) that reduces the verifier's uncertainty of who the identity subject is. Secondly, advanced ZKP schemes (see clause 4.5) are not yet standardized in a way that can be referenced by the eIDAS2 implementing acts. Moreover, eIDAS2 article 5a.14 states:

"Users shall have full control of the use of and of the data in their European Digital Identity Wallet. The provider of the European Digital Identity Wallet shall neither collect information about the use of the European Digital Identity Wallet which is not necessary for the provision of European Digital Identity Wallet services, nor combine person identification data or any other personal data stored or relating to the use of the European Digital Identity Wallet with personal data from any other services offered by that provider or from third party services which are not necessary for the provision of European Digital Identity Wallet services, unless the user has expressly requested otherwise."

Hence, eIDAS2 article 5a.14 puts a requirement on the EUDI Wallet Providers to not gather unnecessary personal data, which in turn could be used for issuer collusion of linkable user information.

In conclusion, selective disclosure and unlinkability are potential components in a privacy-by-design solution. Their impact on privacy depends, however, on an entity's ability to reduce uncertainty about a user's identity from the attribute presentation. When an entity relies solely on linking attributes to reduce uncertainty (and few do), selective disclosure and unlinkability are vital. However, when the entity controls the context or requires user identifying attributes for service provision, non-technical measures (e.g. contractual, economic, and/or regulatory) may be necessary to ensure user privacy and data control.

No technical solution can offer complete control over data and privacy, which requires a more comprehensive approach. Determined, potentially malicious, and well-resourced entities can identify a user and map their behaviour regardless of technical countermeasures employed. This stems from the inherently leaky nature of (Q)EAA presentations, even presentations that do not contain identity subject attributes. For instance, in the context of the European Digital Identity Wallet, a presentation and the associated flow reveals, among other things, that the user has a certified and capable device, often an IP address, attestation issuance dates, identifies an actor the user has been in contact with, and reveals an attestation type the user is eligible to request.

The above regulatory discussion notwithstanding, the present report focuses on various technical solutions that can increase the cost associated with uncertainty reduction (and thus e.g. on verifier and issuer unlinkability). Any (Q)EAA solution that seeks to ensure user privacy has to consider these technical solutions as part of a more comprehensive approach.

Identity matching in eIDAS2

The proposed eIDAS2 regulation [i.86] also includes recitals and articles on identity matching. Recital 55 in eIDAS2 defines identity matching as follows:

"'identity matching' means a process where person identification data, or electronic identification means are matched with or linked to an existing account belonging to the same person;"

Furthermore, eIDAS2 article 11a.2 states:

"Member States shall provide for technical and organisational measures to ensure a high level of protection of personal data used for identity matching and to prevent the profiling of users."

High level protection of personal data for identity matching can be achieved with selective disclosure of attributes.

EXAMPLE 8: Assume that a relying party requests a user to get identified based on the attributes Name, Date of birth and Place of birth. The relying party will need these attributes only to perform identity matching. Instead of revealing the entire PID, which will provide superfluous person identification data to the relying party, the user can select to disclose only the requested attributes Name, Date of birth and Place of birth, which the relying party can use to perform the identity matching.

Descriptions of selective disclosure and unlinkability in the ARF

The ARF [i.59] also defines the term selective disclosure as follows in clause 2:

"The capability of the EUDI Wallet that enables the User to present a subset of attributes provided by the PID and/or (Q)EAAs."

Furthermore, in the ARF outline [i.58] the term unlinkability is also introduced as follows in clause 5:

"The Wallet shall ensure an appropriate level of privacy, implementing policies about non-traceability and unlinkability of user's activities for third parties as appropriate considering:

- the applicable legal context for identity providers and attestation providers;
- the need to retain evidence for dispute resolution purpose;
- the right for the user to be informed of the use of their EUDI Wallet."

More specifically, the ARF [i.59] mandates ISO/IEC 18013-5 [i.140] Mobile Security Object (MSO) and IETF SD-JWT to enable selective disclosure of the EUDI Wallet PID formats. In the ARF section 5.1.2 "Issuing requirements for PID" it is stated:

"PID attestation MUST enable Selective Disclosure of attributes by using Selective Disclosure for JWTs (SD-JWT) and Mobile Security Object (ISO/IEC 18013-5) scheme according to the data model."

The ISO mDL MSO and IETF SD-JWT are mandatory as PID selective disclosure mechanisms in use cases where the Relying Party relies on LoA High as defined in EU CIR 2015/1502 [i.83], to enable cross border identification using PID attributes at LoA High. Hence, the requirements in EU CIR 2015/1502, in conjunction with Regulation (EU) No 1025/2012 on European standardisation [i.88] and the SOG-IS catalogue of approved cryptographic algorithms [i.188], have resulted in this restricted selection of PID formats for the EUDI Wallet.

However, the ARF also specifies the EUDI Wallet support for additional (Q)EAA formats and proof mechanisms, which aims at enabling flexibility and additional feature support for use cases that cannot be met by ISO mDL MSO and IETF SD-JWT (such as in the areas of health, education credentials, etc.). Hence, the EUDI Wallet allows for other selective disclosure techniques based on multi-message signature schemes or proofs for arithmetic circuits but does not mandate support for these.

It should be observed that the ARF holds no legal value and does not prejudge the forthcoming legislative process and the final mandatory legal requirements for EUDI Wallets. Nor does it discuss unlinkability to the same extent as selective disclosure. Only the finally adopted eIDAS2 regulation, and the implementing and delegated acts adopted under that legal basis, will be mandatory. The ARF will be aligned to the final adoption of eIDAS2. Hence, the ARF provides guidelines to the present report for the PID formats to be analysed with respect to selective disclosure in the context of eIDAS2, although the present document may also provide recommendations for additional selective disclosure and ZKP schemes for future versions of the ARF or to be considered for further ETSI standardization.

1 Scope

The present document analyses cryptographic schemes for selective disclosure and their potential application for privacy of electronic attestation attributes in line with the expected requirement of the proposed regulation amending Regulation (EU) No 910/2014 (commonly called eIDAS2) [i.86].

NOTE 1: The term selective disclosure is a collective term that may also include various concepts of unlinkability, and predicates such as range proofs, depending on the context of the specific cryptographic scheme. The scope of the present document is primarily to describe selective disclosure and unlinkability properties of each analysed cryptographic scheme.

NOTE 2: Range proofs, and more general predicate proofs as well as general-purpose ZKPs are out of scope in the ARF [i.59]. If an analysed cryptographic scheme relies on any of these features, they will be described in the context of that particular cryptographic scheme.

The present document aims at providing a comprehensive overview of existing cryptographic schemes for selective disclosure and the formats and protocols associated with these cryptographic schemes.

The aim of the present document is first to provide input to ETSI standardization relating to how selective disclosure may be applied to the eIDAS2 (Qualified) Electronic Attribute Attestations ((Q)EAA) and Person Identification Data (PID). More specifically, the present report may serve as input to (Q)EAA issuance policies as being specified in ETSI TS 119 471 [i.80] and (Q)EAA profiles as being specified in ETSI TS 119 472-1 [i.81].

Second, the present document will also analyse the policy requirements for (Q)TSPs and PID providers issuing (Q)EAAs or PIDs with selective disclosure capabilities to EUDI Wallets.

Third, the present document analyses how the user of an EUDI Wallet can present selected attributes of a (Q)EAA or PID to relying parties (or (Q)TSPs acting as relying parties). Consequently, the present document can highlight needs that may require future standardization efforts.

The present document analyses the concepts of selective disclosure, unlinkability, and predicates (including range proofs) in the following main clauses:

- Selective disclosure signature schemes (clause 4): This clause describes the academic research of the cryptographic algorithms and schemes that shape the foundation for selective disclosure signature schemes.
- Selective disclosure (Q)EAA formats (clause 5): This clause describes the (Q)EAA formats that have been developed and standardized based on the aforementioned selective disclosure signature schemes.
- Selective disclosure protocols and systems (clause 6): This clause describes the complete protocols and /or systems that have been developed and standardized based on the aforementioned selective disclosure signature schemes and (Q)EAA formats.

Since the ARF [i.59] specifies the PID to be issued to an EUDI Wallet as ISO mDL [i.140] (with ISO mDL MSO for selective disclosure) or W3C Verifiable Credentials (with SD-JWT for selective disclosure), these formats and protocols are analysed in more detail in clause 7.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Adida: "Helios: Web-based Open-Audit Voting".

[i.2] Paquin-Zaverucha: "U-Prove Cryptographic Specification V1.1".

[i.3] Alikhani-Brunner-Crépeau et al: "Experimental relativistic zero-knowledge proofs".

[i.4] Altmann: "Inequality tests in salted attribute digest based attestations".

[i.5] Ames-Hazay-Ishai-Venkitasubramaniam: "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup".

[i.6] Arapinis-Cortier-Kremer-Ryan: "Practical Everlasting Privacy".

[i.7] Argo-Güneysu-Jeudy et al: "Practical Post-Quantum Signatures for Privacy".

[i.8] Argo-Jeudy-Land: "Lattice Anonymous Credentials".

[i.9] Asghar: "A Survey on Blind Digital Signatures".

[i.10] Au-Susilo-Mu: "Constant-size dynamic k-TAA".

[i.11] Au-Tsang-Susilo-Mu: "Dynamic Universal Accumulators for DDH Groups and Their Application to Attribute-Based Anonymous Credential Systems".

[i.12] Babel-Sedlmeir: "Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs".

[i.13] Barki-Brunet-Desmoulins-Traoré: "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials".

[i.14] Bellare-Goldreich-Goldwasser: "Incremental Cryptography: The Case of Hashing and Signing".

[i.15] Ben-Sasson-Bentov-Horesh-Riabzev: "Scalable, transparent, and post-quantum secure computational integrity (zk-STARK)".

[i.16] Ben-Sasson-Bentov-Horesh-Riabzev: "Scalable zero-knowledge with No Trusted Setup".

[i.17] Ben-Sasson-Chiesa-Genkin-Tromer-Virzs: "SNARKs for C: Verifying Program Executions Succinctly and in zero-knowledge".

[i.18] Ben-Sasson-Chiesa-Riabzev-Spooner: "Aurora: Transparent Succinct Arguments for R1CS".

[i.19] Ben-Sasson-Tromer: "Succinct Non-Interactive zero-knowledge for a von Neumann Architecture".

[i.20] Benhamouda-Lepoint-Loss et al: "On the (in)Security of ROS".

[i.21] Benjumea-Lopez-Montenegro-Troya: "A First Approach to Provide Anonymity in Attribute Certificates".

[i.22] Bennett-Brassard: "Quantum cryptography: Public key distribution and coin tossing".

[i.23] Bitcoin: "BIP-32 Bitcoin Improvement Proposal 32".

[i.24] Boneh-Bortz et al: "Private Information Retrieval".

[i.25] Boneh-Boyen-Shacham: "Short Group Signatures".

[i.26] Boneh-Lynn-Shacham: "Short Signatures from the Weil Pairing".

[i.27] Bootle-Lyubashevsky-Nguyen-Sorniotti: "A Framework for Practical Anonymous Credentials from Lattices".

[i.28] Bowe: "BLS12-381: New zk-SNARK Elliptic Curve Construction".

[i.29] Bowe-Grigg-Hopwood: "Recursive Proof Composition without a Trusted Setup".

[i.30] Brands: "A Technical Overview of Digital Credentials".

[i.31] Brands: "Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy".

[i.32] Brassard-Chaum-Crépeau: "Minimum disclosure proofs of knowledge".

[i.33] Broadbent-Ji-Song-Watrous: "Zero-knowledge proof systems for QMA".

[i.34] BSI TR-03110: "Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS token".

[i.35] Bull-Stanski-Squire: "Content extraction signatures using XML digital signatures and custom transforms on-demand".

[i.36] Bünz-Bootle-Boneh: "Bulletproofs: Short Proofs for Confidential Transactions and More".

[i.37] Bünz-Fisch-Szepieniec: "Transparent SNARKs from DARK Compilers".

[i.38] Camenisch-Drijvers-Lehmann: "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited".

[i.39] Camenisch-Drijvers-Lehmann-Neven-Towa: "Short Threshold Dynamic Group Signatures".

[i.40] Camenisch-Lysyanskaya: "A Signature Scheme with Efficient Protocols".

[i.41] Camenisch-Lysyanskaya: "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation".

[i.42] Camenisch-Lysyanskaya: "Dynamic accumulators and application to efficient revocation of anonymous credentials".

[i.43] Camenisch-Lysyanskaya: "Signature Schemes and Anonymous Credentials from Bilinear Maps".

[i.44] Camenisch-Mödersheim-Sommer: "A Formal Model of Identity Mixer".

[i.45] Campanelli-Fiore-Querol: "LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs".

[i.46] Canard-Coisel-Jambert-Traoré: "New Results for the Practical Use of Range Proofs".

[i.47] Carney: "On Zero-Knowledge Proofs over the Quantum Internet".

[i.48] CEN TC/224 WG17: "EN 419 211: Protection profiles for secure signature creation device" (produced by CEN).

[i.49] CEN TC/224 WG20: "New work item: PID onboarding technical standard" (produced by CEN).

[i.50] Chadwick: "The Use of FIDO2 and Verifiable Credentials".

[i.51] Chalkias-Cohen-Lewi-Moezinia-Romailler: "HashWires: Credential-Based Range Proofs".

[i.52] Chase-Meiklejohn-Zaverucha: "Algebraic MACs and keyed-verification anonymous credentials".

[i.53] Chaum: "Blind signatures for untraceable payments".

[i.54] Chaum-van Heyst: "Group Signatures".

[i.55] Chen-Page-Smart: "On the Design and Implementation of an Efficient DAA Scheme".

[i.56] Chiesa-Bitansky-Canetti: "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again".

[i.57] Chiesa-Hu-Maller-Mishra: "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS".

[i.58] Common Union Toolbox for a Coordinated Approach Towards a European Digital Identity Framework: "The European Digital Identity Wallet Architecture and Reference Framework, Outline".

[i.59] Common Union Toolbox for a Coordinated Approach Towards a European Digital Identity Framework: "The European Digital Identity Wallet Architecture and Reference Framework, Version 1.3.0".

NOTE: The European Digital Identity Wallet Architecture and Reference Framework is commonly called the ARF. The ARF version 1.3.0 is used as reference in the present document.

[i.60] Costello-Fournet-Howell et al.: "Geppetto: Versatile Verifiable Computation".

[i.61] Crites: "Delegatable Anonymous Credentials from Mercurial Signatures".

[i.62] Crites-Lysyanskaya: "Mercurial Signatures for Variable-Length Messages".

[i.63] CRYSTALS: "Dilithium digital signature scheme".

[i.64] Damgård and Triandopoulos: "Supporting Non-membership Proofs with Bilinear-map Accumulators".

[i.65] Delignat-Lavaud-Fournet-Kohlweiss-Parno: "Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation".

[i.66] DIF: "Blind Signatures extension of the BBS Signature Scheme".

[i.67] DIF: "Presentation Exchange 2.0.0".

[i.68] DIF: "Wallet Security Working Group".

[i.69] Dutto-Margaria-Sanna-Vesco: "Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity".

[i.70] Eberhardt-Tai: "ZoKrates - Scalable Privacy-Preserving Off-Chain Computations".

[i.71] EBA (European Banking Association): "Register of payment and electronic money institutions under PSD2".

[i.72] Ebrahimi: "Post-quantum Efficient Proof for Graph 3-Coloring Problem".

[i.73] ETSI EN 319 162-1: "Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC); Part 1: Building blocks and ASiC baseline containers".

[i.74] ETSI EN 319 401: "Electronic Signatures and Trust Infrastructures (ESI); General Policy Requirements for Trust Service Providers".

[i.75] ETSI EN 319 411-1: "Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements".

[i.76] ETSI TR 119 001: "Electronic Signatures and Infrastructures (ESI); The framework for standardization of signatures; Definitions and abbreviations".

[i.77] ETSI TS 119 495: "Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Certificate Profiles and TSP Policy Requirements for Open Banking".

[i.78] ETSI TS 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists".

[i.79] ETSI TS 119 462: "Electronic Signatures and Trust Infrastructures (ESI); Wallet interfaces for trust services and signing".

[i.80] ETSI TS 119 471: "Electronic Signatures and Trust Infrastructures (ESI); Policy and Security requirements for Providers of Electronic Attestation of Attribute Services".

[i.81] ETSI TS 119 472-1: "Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestations of Attributes; Part 1: General requirements".

[i.82] European Banking Authority: "Regulatory Technical Standards on strong customer authentication and secure communication under PSD2".

[i.83] European Union: "Commission Implementing Regulation (EU) 2015/1502 of 8 September 2015 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market".

[i.84] European Union: "Commission Implementing Decision (EU) 2015/1505 of 8 September 2015 laying down technical specifications and formats relating to trusted lists pursuant to Article 22(5) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market".

[i.85] European Union: "Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC".

NOTE: The Directive (EU) 2015/2366 is commonly called PSD2.

[i.86] European Union: "European Digital Identity Framework - European Parliament legislative resolution of 29 February 2024 on the proposal for a regulation of the European Parliament and of the Council amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity (COM(2021)0281 - C9-0200/2021 - 2021/0136(COD))".

NOTE 1: The Proposal for a regulation of the European Parliament and of the Council amending Regulation (EU) No 910/2014 is commonly called eIDAS2.

NOTE 2: The European Commission issued the first proposal of eIDAS2 in June 2021. The European Council issued an amended edition of eIDAS2 in December 2022 and the European Parliament issued another amended edition of eIDAS2 in February 2023. The eIDAS2 proposal, which is based on the agreement in the eIDAS2 trialogue, was published in December 2023. Finally, the EU Parliament voted to approve the eIDAS2 regulation in February 2024. Unless stated otherwise, the eIDAS2 adopted text, which was issued in February 2024, is by default the referenced version of the eIDAS2 regulation.

[i.87] European Union: "Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC".

NOTE: The Directive (EU) 910/2014 is commonly called eIDAS.

[i.88] European Union: "Regulation (EU) No 1025/2012 of the European Parliament and of the Council on European standardisation".

[i.89] Eurosmart PP-0117: "Protection Profile for Secure Sub-System in System-on-Chip (3S in SoC)".

[i.90] Evans-Angeris: "Succinct Proofs and Linear Algebra".

[i.91] Federal Public Key Infrastructure Policy Authority: "X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework".

[i.92] Fett: "SD-JWT VC DM Credential Format".

[i.93] Gabison-Williamson-Ciobotaru: "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge".

[i.94] Garcia-Rodriguez-Moreno-Bernabe-Skarmeta: "Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures".

[i.95] Global Platform: "TEE Protection Profile".

[i.96] Goldreich-Micali-Wigderson: "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems".

[i.97] Goldwasser-Micali-Rackoff: "The knowledge complexity of interactive proof systems".

[i.98] Grontas-Pagourtzis-Zacharakis-Zhang: "Towards everlasting privacy and efficient coercion resistance in remote electronic voting".

[i.99] Groth: "Short pairing-based non-interactive zero-knowledge arguments".

[i.100] Grover: "A fast quantum mechanical algorithm for database search".

[i.101] Haines-Gritti: "Improvements in Everlasting Privacy: Efficient and Secure Zero Knowledge Proofs".

[i.102] Haines-Mosaheb-Müller-Pryvalov: "SoK: Secure e-voting with everlasting privacy".

[i.103] Heath-Yang-Devecsery-Kolesnikov: "Zero Knowledge for Everything and Everyone: Fast ZK Processor with Cached ORAM for ANSI C Programs".

[i.104] Hyperledger Foundation: "AnonCreds Specification v1.0".

[i.105] Hyperledger Foundation: "Hyperledger Aries".

[i.106] Hyperledger Foundation: "Hyperledger Fabric".

[i.107] Hyperledger Foundation: "Hyperledger Indy".

[i.108] Hyperledger Foundation: "Hyperledger Ursa SDK".

[i.109] IBM® Research: "Identity Mixer (IDEMIX)".

[i.110] ICT Trust and Security Research: "Attribute based Credentials for Trust (ABC4Trust)".

[i.111] IETF: "Authentic Chained Data Containers (ACDC)".

[i.112] IETF: "SD-JWT-based Verifiable Credentials (SD-JWT VC)".

[i.113] IETF: "Self-Addressing IDentifier (SAID)".

[i.114] IETF: "The Gordian Envelope Structured Data Format".

[i.115] IETF CFRG: "Asynchronous Remote Key Generation (ARKG) algorithm".

[i.116] IETF CFRG: "The BBS Signature Scheme".

[i.117] IETF: "Key Blinding for Signature Schemes".

[i.118] IETF CFRG: "Pairing-Friendly Curves".

[i.119] IETF IESG: "JOSE and COSE Encoding for Post-Quantum Signatures".

[i.120] IETF JOSE: "JSON Web Proof (JWP)".

[i.121] IETF OAUTH: "OAuth Status List".

[i.122] IETF OAUTH: "SD-JWT-based Verifiable Credentials with JSON payloads (SD-JWT VC)".

[i.123] IETF OAUTH: "Selective Disclosure for JWTs (SD-JWT)".

[i.124] IETF RFC 5652: "Cryptographic Message Syntax (CMS)".

[i.125] IETF RFC 5755: "An Internet Attribute Certificate Profile for Authorization".

[i.126] IETF RFC 6066: "Transport Layer Security (TLS) Extensions: Extension Definitions".

[i.127] IETF RFC 6960: "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP)".

[i.128] IETF RFC 6818: "Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[i.129] IETF RFC 7049: "Concise Binary Object Representation (CBOR)".

[i.130] IETF RFC 7515: "JSON Web Signature (JWS)".

[i.131] IETF RFC 7516: "JSON Web Encryption (JWE)".

[i.132] IETF RFC 7519: "JSON Web Token (JWT)".

[i.133] IETF RFC 8152: "CBOR Object Signing and Encryption (COSE)".

[i.134] IETF RFC 8235: "Schnorr Non-interactive Zero-Knowledge Proof".

[i.135] IETF RFC 8259: "JavaScript Object Notation (JSON) Data Interchange Format".

[i.136] IETF RFC 8610: "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures".

[i.137] IETF RFC 9162: "Certificate Transparency Version 2.0".

[i.138] IRMA: "Revocation".

[i.139] ISO/IEC 9796 series: "Information technology - Security techniques - Digital signature schemes giving message recovery".

[i.140] ISO/IEC 18013-5: "Personal identification - ISO-compliant driving licence - Part 5: Mobile driving licence (mDL) application".

[i.141] ISO/IEC CD 18013-7: "Personal identification - ISO-compliant driving licence - Part 7: Mobile driving licence (mDL) add-on functions".

[i.142] ISO/IEC 18370 series: "Information technology - Security techniques - Blind digital signatures".

[i.143] ISO/IEC 20008 series: "Information technology - Security techniques -Anonymous digital signatures".

[i.144] ISO/IEC PWI 24843: "Privacy-preserving attribute-based credentials".

[i.145] ISO/IEC 23220-3: "Cards and security devices for personal identification - Building blocks for identity management via mobile devices - Part 3: Protocols and services for installation and issuing phase".

[i.146] ISO/IEC CD 23220-4: "Cards and security devices for personal identification - Building blocks for identity management via mobile devices - Part 4: Protocols and services for operational phase".

[i.147] ISO/IEC CD 23220-6: "Cards and security devices for personal identification - Building blocks for identity management via mobile devices - Part 6: Mechanism for use of certification on trustworthiness of secure areas".

[i.148] ISO/IEC 27001: "Information security, cybersecurity and privacy protection - Information security management systems - Requirements".

[i.149] ISO/IEC 27002: "Information security, cybersecurity and privacy protection - Information security controls".

[i.150] ISO/IEC CD 27565: "Guidelines on privacy preservation based on zero knowledge proofs".

[i.151] Jeudy-Roux-Langlois-Sanders: "Lattice Signature with Efficient Protocols, Application to Anonymous Credentials".

[i.152] Kampanakis-Panburana-Daw-Van Geest: "The Viability of Post-Quantum X.509 Certificates".

[i.153] Kosba-Papadopoulos-Papamanthou-Song: "MIRAGE: Succinct Arguments for Randomized Algorithms with Applications to Universal zk-SNARKs".

[i.154] Kosba-Papmanthou-Shi: "xJsnark: A Framework for Efficient Verifiable Computation".

[i.155] Lapon-Kohlweiss-Decker-Naessens: "Analysis of Revocation Strategies for Anonymous Idemix Credentials".

[i.156] Libert-Ling-Mouhartem et al: "Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions".

[i.157] Maller-Bowe-Kohlweiss-Meiklejohn: "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings".

[i.158] Menezes: "An Introduction to Pairing-Based Cryptography".

[i.159] Microsoft® Research: "Spartan: High-speed zkSNARKs without trusted setup".

[i.160] Microsoft® Research: "U-Prove".

[i.161] Morales-Agudo-Lopez: "Private set intersection: A systematic literature review".

[i.162] Mouris-Tsoutsos: "Zilch: A Framework for Deploying Transparent Zero-Knowledge Proofs".

[i.163] Nguyen: "Accumulators from Bilinear Pairings and Applications".

[i.164] Nitulescu: "zk-SNARKs: A Gentle Introduction".

[i.165] NIST: "Digital Identities - Mobile Driver's License (mDL)".

[i.166] NIST: "FIPS 204 - Module-Lattice-Based Digital Signature Standard".

[i.167] NIST: "FIPS 205 - Stateless Hash-Based Digital Signature Standard".

[i.168] NIST: "Post-Quantum Cryptography (PQC)".

[i.169] OpenAttestation: "Document Integrity".

[i.170] OpenID Foundation: "OpenID Connect Core 1.0".

[i.171] OpenID Foundation: "OpenID for Verifiable Presentations".

[i.172] OpenID Foundation: "OpenID4VC High Assurance Interoperability Profile with SD-JWT VC".

[i.173] OpenID Foundation: "Self-Issued OpenID Provider v2".

[i.174] Parno-Howell-Gentry et al: "Pinocchio: Nearly Practical Verifiable Computation".

[i.175] Petkus: "Why and How zk-SNARK Works: Definitive Explanation".

[i.176] Pointcheval-Sanders: "Short Randomizable Signatures".

[i.177] PrimeLife: "Identity Mixer".

[i.178] Proos-Zalka: "Shor's discrete logarithm quantum algorithm for elliptic curves".

[i.179] Pussewalage-Oleshchuk: "An Efficient Multi-Show Unlinkable Attribute Based Credential Scheme for a Collaborative E-Health Environment".

[i.180] Radboud University Nijmegen: "IRMA project".

[i.181] Rivest-Shamir: "PayWord and MicroMint: Two simple micropayment schemes".

[i.182] Roetteler-Naehrig-Svore-Lauter: "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms".

[i.183] Rosenberg-White-Garman-Miers: "zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure".

[i.184] Sanders: "Efficient Redactable Signature and Application to Anonymous Credentials".

[i.185] Sanders: "Improving Revocation for Group Signature with Redactable Signature".

[i.186] Setty: "Spartan: Efficient and general-purpose zkSNARKs without trusted setup".

[i.187] Shor: "Algorithms for quantum computation: discrete logarithms and factoring".

[i.188] SOG-IS Crypto Working Group: "SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms".

[i.189] SPHINCS+: "Stateless hash-based signature scheme".

[i.190] Steinfeld-Bull-Zheng: "Content Extraction Signatures".

[i.191] Tan-Groß: "MoniPoly - An Expressive q-SDH-Based Anonymous Attribute-Based Credential System".

[i.192] Trusted Computing Group: "TPM 2.0 Library".

[i.193] U.S. Department of Homeland Security: "Cryptography Review of W3C Verifiable Credentials Data Model (VCDM) and Decentralized Identifiers (DIDs) Standards and Cryptography Implementation Recommendations".

[i.194] Verheul: "SECDSA: Mobile signing and authentication under classical 'sole control'".

[i.195] Vitto-Biryukov: "Dynamic Universal Accumulator with Batch Update over Bilinear Groups".

[i.196] Wahby-Setty-Ren-Blumberg-Walfish: "Efficient RAM and Control Flow in Verifiable Outsourced Computation".

[i.197] Wahby-Tzialla-Shelat: "Doubly-Efficient zkSNARKs Without Trusted Setup".

[i.198] Watrous: "Zero-knowledge against quantum attacks".

[i.199] Webber-Elfving-Weidt-Hensinger: "The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime".

[i.200] W3C®: "Bitstring Status List v1.0 - Privacy-preserving status information for Verifiable Credentials".

[i.201] W3C®: "Data Integrity ECDSA Cryptosuites v1.0".

[i.202] W3C®: "Decentralized Identifiers (DIDs) v1.0".

[i.203] W3C®: "Json Web Proofs for Binary Merkle Trees".

[i.204] W3C®: "Merkle Disclosure Proof 2021".

[i.205] W3C®: "Remove securing JSON, VC-JWT issue #88".

[i.206] W3C®: "Securing Verifiable Credentials using JSON Web Tokens".

[i.207] W3C®: "Universal Wallet 2020".

[i.208] W3C®: "Verifiable Credentials Data Integrity 1.0".

[i.209] W3C®: "Verifiable Credentials Data Model v1.1".

[i.210] W3C®: "Verifiable Credential Data Model v2.0 (working draft)".

[i.211] W3C®: "Web Authentication: An API for accessing Public Key Credentials Level 2".

[i.212] W3C® CCG: "BBS Cryptosuite v2023".

[i.213] ZKProof: "HashWires: Range Proofs from Hash Functions".

[i.214] Zhang-Xie-Zhang-Song: "Transparent Polynomial Delegation and Its Applications to zero-knowledge Proof".

[i.215] Zhao-Liu-Wu et al: "A Tutorial on Quantum Key Distribution".

[i.216] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[i.217] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TR 119 001 [i.76], ETSI EN 319 401 [i.74] and the following apply:

atomic (Q)EAA: (Qualified) Electronic Attestation of Attribute with a single attribute claim

attribute: feature, characteristic or quality of a natural or legal person or of an entity, in electronic form

NOTE: As defined in the ARF [i.59].

authentic source: repository or system, held under the responsibility of a public sector body or private entity, that contains attributes about a natural or legal person and is considered to be the primary source of that information or recognized as authentic in national law

NOTE: As defined in the ARF [i.59].

blind signature: type of digital signature in which the content of a message is disguised (blinded) before it is signed

EXAMPLE: The concept of blind signatures can be exemplified by a voting system in the physical world. The voter encloses an anonymous ballot in a carbon envelope with the voter's name written on the outside. An official verifies the voter's identity and signs the envelope, such that the ballot inside the carbon envelope gets signed with the official's signature. The voter moves the signed ballot to a new unmarked envelope. Hence, the signing official does not see the content of the vote, but a third party can later verify its signature and know that the vote is valid.

NOTE 1: Blinded signatures cater for unlinkability, since the verifier cannot link the signed messages back to the user.

NOTE 2: The U-Prove scheme (clause 6.6.2) utilizes blinded signatures when issuing the credentials.

NOTE 3: Blind signatures are specified in the ISO/IEC 18370 series [i.142], which allow a user to obtain a digital signature as specified in the ISO/IEC 9796 series [i.139]. ISO/IEC 18370-1 [i.142] also introduces a model of selectively disclosing attributes by using blind signatures.

NOTE 4: Sometimes blind signature schemes leverage ZKPs to ensure the signer that the blindly signed content is well-formed (adheres to some requirements).

Electronic Attestation of Attributes (EAAs): attestation in electronic form that allows the authentication of attributes

NOTE: As defined in the ARF [i.59].

EUDI Wallet Instance: instance of an EUDI Wallet Solution belonging to and which is controlled by a user

NOTE: As defined in the ARF [i.59].

EUDI Wallet Provider: organization, public or private, responsible for the operation of a eIDAS-compliant EUDI Wallet Solution that can be instantiated, e.g. through installation and initialization

NOTE: As defined in the ARF [i.59].

EUDI Wallet Solution: EUDI Wallet Solution is the entire product and service owned by an EUDI Wallet Provider, offered to all users of that solution. An EUDI Wallet solution can be certified as being EUDI-compliant by a CAB

NOTE: As defined in the ARF [i.59].

ISO mDL: ISO mobile driving license (mDL) according to ISO/IEC 18013-5 [i.140] and ISO/IEC CD 18013-7 [i.141].

Issuing Authority Certification Authority (IACA): certification authority in the context of ISO mDL that issues certificates for the creation of ISO mDL MSOs and auxiliary certificates for revocation services or securing online services (such as TLS servers)

issuer: issuing authority that is accredited or supervised for issuing certificates, attested attributes, ISO mDL or credentials

NOTE 1: In the context of eIDAS2, the issuer can be a Person Identification Data Provider issuing PIDs or a (Qualified) Trust Service Provider issuing (Q)EAAs (as defined in the ARF [i.59]).

NOTE 2: In the context of ISO mDL, the issuer is an IACA that issues certificates for the creation and operation of ISO mDL MSOs.

MSO: ISO mobile driving license Mobile Security Object (MSO), with salted attribute hashes of the user's elements in the ISO mDL mdoc

Person Identification Data (PID): set of data enabling the identity of a natural or legal person, or a natural person representing a legal person to be established

NOTE: As defined in the ARF [i.59].

Person Identification Data Provider (PIDP): Member State or legal entity providing Person Identification Data to users

NOTE: As defined in the ARF [i.59].

predicate proof: verifiable Boolean assertion (true or false) about the value of another attribute claim in the attestation without disclosing the claim value itself

EXAMPLE 1: Predicate proofs are often in the form of range proofs (greater than, less than), equal to, set member, etc.

EXAMPLE 2: A user can prove to a verifier that he/she is an EU citizen, without revealing in which Member State.

NOTE 1: Predicate proofs are often employed in ZKP systems aimed at limiting information disclosure.

NOTE 2: The definition of predicate proof above is quoted from the Hyperledger AnonCreds specification [i.104].

Qualified Electronic Attestations of Attributes (QEAAs): Electronic Attestation of Attributes, which is issued by a Qualified Trust Service Provider and meets the requirements laid down in eIDAS Regulation amendment proposal Annex V [i.86]

NOTE: A (Qualified) Electronic Attestation of Attribute is abbreviated as (Q)EAA, and is a collaborative term that is used when either a QEAA or an EAA could be applicable for the context.

Quantum-Safe Cryptography (QSC): cryptographic algorithms (typically public-key algorithms) that are expected to be secure against a cryptanalytic attack by a quantum computer

NOTE 1: NIST conducts a research program [i.168] to identify candidates for QSC algorithms that can be standardized. The signature scheme finalists (December 2023) are FIPS 204 [i.166] (based on CRYSTALS Dilithium [i.63]) and FIPS 205 [i.167] (based on SPHINCS+ [i.189]). Unless stated otherwise, FIPS 204 and FIPS 205 are referred to as QSC signature schemes throughout the present document.

NOTE 2: The term post-quantum cryptography (PQC) is sometimes used in other literature, and is equivalent to the term quantum-safe cryptography (QSC) that is used throughout the present document.

NOTE 3: The post-quantum world is the era when quantum computers are expected to be capable of breaking asymmetric cryptographic algorithms based on the discrete logarithm problem (DLP) or the difficulty of factoring large composite numbers. Asymmetric cryptographic algorithms that are plausibly vulnerable to such attacks are RSA, SDH, ECDSA, ECSchnorr, etc.

NOTE 4: The pre-quantum world is the era when quantum computers are not (yet) capable of breaking asymmetric cryptographic algorithms based on the DLP or the difficulty of factoring large composite numbers.

NOTE 5: Plausible quantum-safe cryptographic systems, protocols or signature schemes may be implemented either by introducing quantum-safe components, and/or by selecting a quantum-safe signature method like FIPS 204 or FIPS 205

range proof: method by which the user (prover) can prove to the relying party (verifier) that a number is in a given range (lower and upper bound) without disclosing the actual number

EXAMPLE: A 21 year old user can prove to a verifier that he/she is older than 18 years, without revealing their actual age.

NOTE: Range proofs are subsets of predicate proofs; a range proof for inclusion in an interval is typically generated by using two inequality tests, one for each boundary.

SD-JWT: W3C Verifiable Credential (VC) used in conjunction with a SD-JWT [i.123] with a list of salted hash values of the user's claims in the W3C VC

selective disclosure: capability of the EUDI Wallet that enables the user to present a subset of attributes provided by the PID and/or (Q)EAAs

NOTE 1: As defined in the ARF [i.59].

EXAMPLE: Assume that a user's EUDI Wallet includes a (Q)EAA with the attributes first name, last name, birth date, and address. The user can for example selectively disclose only its first name.

NOTE 2: ISO mDL MSO (clause 7.2) and IETF SD-JWT (clause 7.3) can present selectively disclosed attributes based on the design of salted attribute hashes.

unlinkability: lack of information required to connect the user's selectively disclosed attributes beyond what is disclosed

NOTE 1: Verifier unlinkable means that one or more verifiers cannot collude to determine if the selectively disclosed attributes describe the same identity subject.

NOTE 2: Issuer unlinkable means that one or more issuers cannot collude to determine if the selectively disclosed attributes describe the same identity subject.

NOTE 3: Fully unlinkable means that no party can collude to determine if the selectively disclosed attributes describe the same identity subject.

NOTE 4: Multi-show unlinkability means that a (Q)EAA can be used for multiple presentations, which cannot be used to connect the user's selectively disclosed attributes.

NOTE 5: The opposite of multi-show unlinkability means that a (Q)EAA can only be used once for a presentation, since the (Q)EAA will thereafter reveal information that can be used for linkability.

EXAMPLE 1: Assume that a user's EUDI Wallet includes a (Q)EAA with the attributes first name and last name. The user can disclose its first name to one relying party, and its last name to another relying party. The relying parties cannot exchange any information that allows them to link the user's first name disclosure to the last name disclosure.

EXAMPLE 2: The same principle applies if the user discloses its first name to a relying party and later discloses its last name to the same relying party and the single relying party cannot link the user's first name disclosure to its last name disclosure.

EXAMPLE 3: The same principle applies if the issuer colludes with the verifier without being able to link the user's first name disclosure to its last name disclosure.

user: natural or legal person using an EUDI Wallet

NOTE 1: As defined in the ARF [i.59].

NOTE 2: In the context of selective disclosure, the user is also the prover of the attributes it presents from its EUDI Wallet.

NOTE 3: The user is sometimes also denoted as holder in other specifications.

verified issuer certificate authority list (VICAL) provider: ISO mDL provider that can compile, operate and provide trust anchors (such as IACA trust anchors) in the form of a service to mDL participants

W3C VCDM: W3C Verifiable Credential (VC) Data Model (DM)

NOTE: The W3C VCDM v1.1 exists as a recommendation [i.209], whilst W3C VCDM v2.0 [i.210] is a working draft.

W3C VCDI: W3C Verifiable Credential (VC) Data Integrity (DI)

NOTE: The W3C VCDI v1.0 [i.208] exists as a recommendation.

Zero-Knowledge Proof (ZKP): method by which the user (prover) can prove to the relying party (verifier) that a given statement is true while the user does not provide any additional information apart from the fact that the statement is true

NOTE 1: There are special-purpose ZKPs that can only prove very specific statements (e.g. knowledge of a pre-image of a hash or knowledge of a signature under a specific digital signature scheme) and general-purpose or programmable ZKPs that allow to prove any statement. Programmable ZKPs usually involve a compiler from some programming language that describes the statement to be proved (e.g. program returns a certain public value upon correct execution on a private input) into a ZKP proving and verification program.

NOTE 2: A ZKP protocol should meet the following three criteria: Completeness (if the statement is true then a user can convince a verifier), soundness (a fraudulent user can not convince a verifier of a false statement beyond negligible probability - how small is a parameter choice, e.g. 2^{-128}), and zero-knowledge (the interaction only reveals if a statement is true and nothing else beyond what can trivially be inferred from the statement itself).

NOTE 3: A ZKP system provides predicate proofs, selective disclosure and unlinkability per definition, provided the verifier does not specifically ask for all (Q)EAA or linkable data.

EXAMPLE: zk-SNARKs (clause 4.5.2) are examples of programmable ZKP protocols, whereas CL-signatures and BBS+ are examples of special-purpose ZKP protocols

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TR 119 001 [i.76] and the following apply:

3S	Secure Sub-System
AA	Attribute Authority
ABC	Attribute Based Credentials
AIR	Algebraic Intermediate Representation
ARF	Architecture and Reference Framework
ARKG	Asynchronous Remote Key Generation
BBS	Boneh-Boyen-Shacham
BLE	Bluetooth Low Energy
BLS	Barreto-Lynn-Scott (pairing-friendly elliptic curves)

BIP-32	Bitcoin Improvement Proposal 32
BSI	Bundesamt für Sicherheit in der Informationstechnik
CBOR	Concise Binary Object Representation
CCG	Credentials Community Group
CD	Committee Draft (in the context of ISO)
CDDL	Concise Data Definition Language
CES	Content Extraction Signatures
CFRG	Crypto Forum Research Group
CIR	Commission Implementing Regulation
CL	Camenisch-Lysyanskaya
CLRSA	Camenisch-Lysyanskaya signatures based on RSA
CMS	Cryptographic Message Syntax
COSE	CBOR Object Signing and Encryption
CRL	Certificate Revocation List
CRYSTALS	Cryptographic Suite for Algebraic Lattices
CS	Computationally Sound
CWT	CBOR Web Tokens
DAA	Direct Anonymous Attestation
DAG	Directed Acyclic Graph
DIF	Digital Identity Foundation
DLP	Discrete Logarithm Problem
DLREP	Discrete Logarithm Representation
dp-ABC	distributed privacy-preserving Attribute Based Credentials
EAA	Electronic Attestation of Attributes
EBA	European Banking Association
ECDL	Elliptic Curve Discrete Logarithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDSA-SD	Elliptic Curve Digital Signature Algorithm with Selective Disclosure
ECSDSA	EC Schnorr DSA
eMRTD	Electronic Machine Readable Travel Document
EPID	Enhanced Privacy ID
EUDI	European Union Digital Identity
EUDIW	European Union Digital Identity Wallet
FIDO	Fast Identity Online
FIPS	Federal Information Processing Standards
FPKIPA	Federal Public Key Infrastructure Policy Authority
FRI	Fast Reed Solomon Interactive Oracle Proof
G3C	Graph 3-Colouring
HAIP	High Assurance Interoperability Profile
HDK	Hierarchical Deterministic Key
HNDL	Harvest Now Decrypt Later
IACA	Issuing Authority Certification Authority
ICAO	International Civil Aviation Organization
IDEMIX	Identity Mixer
IEC	International Electrotechnical Commission
IOP	Interactive Oracle Proof
JAdES	JSON Advanced Electronic Signatures
JOSE	JSON Object Signing and Encryption
JSON	JavaScript Object Notation
JSON-LD	JSON for Linking Data
JWS	JSON Web Signature
JWT	JSON Web Token
KBSS	Key Blinding for Signature Schemes
KDF	Key Distribution Function
k-TAA	k-Times Anonymous Authentication
KVAC	Keyed-Verification Anonymous Credentials
LLVM	Low Level Virtual Machine
MAC	Message Authentication Code
MAC_BBS	Message Authentication Code based Boneh-Boyen-Shacham signatures
mDL	mobile Driving License
MSO	Mobile Security Object
NCCoE	National Cybersecurity Center of Excellence

NP	Nondeterministic Polynomial-time
NTRU	Number Theory Research Unit
OCSP	Online Certificate Status Protocol
OID4VC	OpenID for Verifiable Credentials
OID4VP	OpenID for Verifiable Presentations
OIDC	OpenID Connect
p-ABC	privacy-preserving Attribute Based Credentials
PCP	Probabilistically Checkable Proofs
PCS	Polynomial Commitment Scheme
PID	Person Identification Data
PIDP	Person Identification Data Provider
PII	Personal Identifiable Information
PIOP	Polynomial Interactive Oracle Proof
PIR	Private Information Retrieval
PKD	Public Key Directory
PKIX	Public-Key Infrastructure (X.509)
PQC	Post-Quantum Cryptography
PSD2	Payment Services Directive v2
PSI	Private Set Intersection
PS-GS	Pointcheval-Sanders Group Signatures
PS-MS	Pointcheval-Sanders Multi-Signatures
PWI	Preliminary Work Item
QAP	Quadratic Arithmetic Program
QEAA	Qualified Electronic Attestation of Attributes
QKD	Quantum Key Distribution
QMA	Quantum Merlin Arthur
QSC	Quantum-Safe Cryptography
qSDH	q-Strong Diffie-Hellman
QTSP	Qualified Trust Service Provider
QWAC	Qualified Website Authentication Certificate
RDF	Resource Description Framework
RL	Revocation List
ROM	Random Oracle Model
ROS	Random inhomogeneities in a Overdetermined Solvable system of linear equations
RSAREP	RSA Representation
RTS	Regulatory Technical Standard
R1CS	Rank-1 Constraint System
SAID	Self-Addressing IDentifier
SD	Selective Disclosure
SDH	Strong Diffie-Hellman
SD-JWT	Selective Disclosure JSON Web Token
SECDSA	Split-ECDSA
SEP	Signatures with Efficient Protocols
SIOP2	Self-Issued OpenID Provider v2
SL	Status List
SoC	System on Chip
SOG-IS	Senior Officials Group Information Systems Security
SSP	Square Span Program
TCG	Trusted Computing Group
TLS	Transport Layer Security
TPM	Trusted Platform Module
UI	User Interface
UUID	Universal Unique Identifier
VC	Verifiable Credential
VCDI	Verifiable Credential Data Integrity
VCDM	Verifiable Credential Data Model
VDR	Verifiable Data Registry
VICAL	Verified Issuer Certificate Authority List
VP	Verifiable Presentation
W3C	World Wide Web Consortium
WG	Working Group
XAdES	XML Advanced Electronic Signatures

YAML	Yet Another Multicolumn Layout
ZKP	Zero-Knowledge Proof
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge
zk-STARK	Zero-Knowledge Scalable Transparent Argument of Knowledge
zkVM	zero-knowledge virtual machine

4 Selective disclosure signature schemes

4.1 General

The present clause provides an analysis of a set of selective disclosure signature schemes.

The topics for the analysis of each selective disclosure signature scheme are:

- Underlying cryptographic algorithms for selective disclosure, unlinkability and optionally ZKP.
- Maturity of the selective disclosure signature scheme's specification and deployment.
- Cryptographic aspects, more specifically if the cryptographic algorithms used for the selective disclosure signature schemes are approved by SOG-IS and allows for QSC algorithms for future use.

There exist four main categories to enable selective disclosure:

- The first category is using atomic (Q)EAAs, which is described in clause 4.2.
- The second category is using a selective disclosure capable multi-message signature scheme, which typically relies on commitments. This category is explained in clause 4.3.
- The third category is signing a collection of salted attribute digests; this category is described in clause 4.4.
- There is also a fourth category of methods that can ensure the privacy of any computable proof (e.g. Bulletproofs, zk-SNARKS, zk-STARKS, etc.). This category is elaborated in clause 4.5. These methods could support additional selective disclosure mechanisms beyond the three main ones listed above.

NOTE: An argument can be made for a selective disclosure mechanism that relies on trusted components for storage and computation. It is possible to store unsigned attribute claims on trusted storage and transport only the requested claims over a secure messaging channel. It is also possible in these setups to associate each storage partition with a unique key and only store a single (Q)EAA per partition in order to ensure the proper pairing of attributes. A solution based on these principles is detailed in BSI TR-03110 [i.34]. The solutions described in the present document, however, include only signature based selective disclosure schemes.

Each of the four main ways are described in the clauses below.

4.2 Atomic (Q)EAAs schemes

An atomic electronic attribute attestation is a (Q)EAA with a single attribute claim, which can be issued by a (Q)TSP upon request or as part of a batch to an EUDI Wallet. The atomic (Q)EAAs can be selected by the user and be included in a verifiable presentation that is presented to a verifier.

An example of a solution based on atomic (Q)EAAs is illustrated in Figure 2. In this scenario, the user needs a parking ticket to enter a car parking. For that purpose, the user enrols for atomic (Q)EAAs from a transport authority (with the car registration number), from a civil registry (with the address), and from a payment service provider (with the paid amount). The user's EUDI Wallet can then combine these atomic (Q)EAAs into a verifiable presentation, which is the parking ticket that is presented to the car parking clerk.

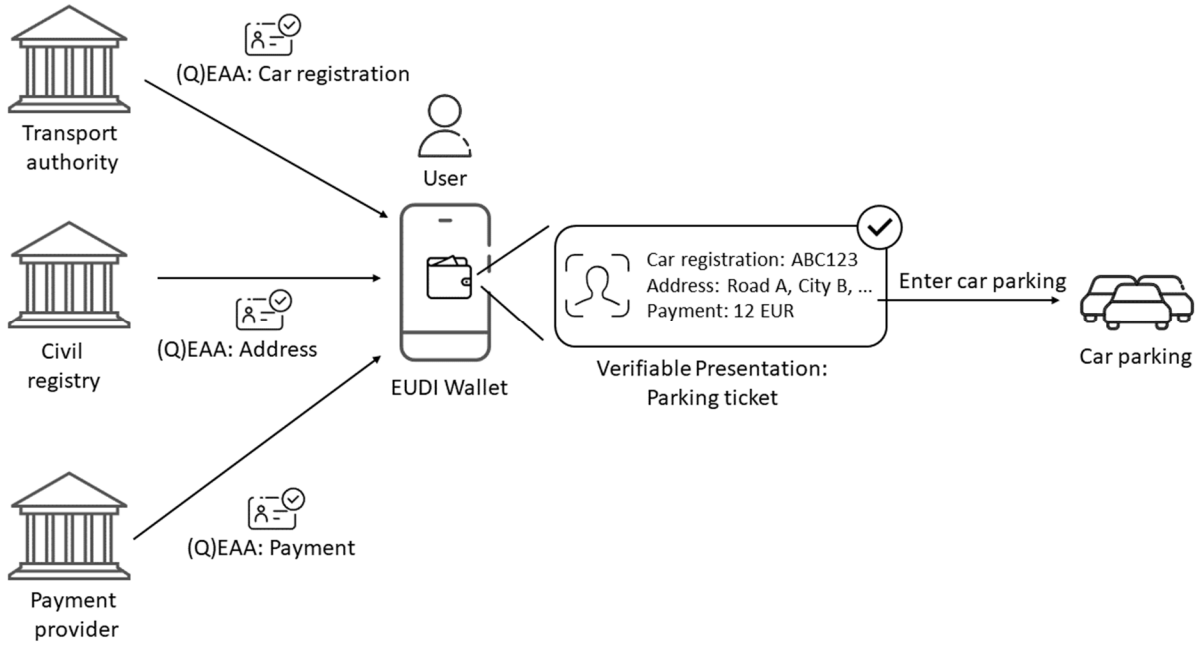

Figure 2: Example of atomic attribute credentials

The underlying cryptographic algorithms depend on the (Q)TSPs' signing algorithms of the (Q)EAAs and the proof key when signing the verifiable presentation. Hence, it is possible to select signature algorithms that are approved by SOG-IS and/or allow for QSC. (More information on the specific (Q)EAA formats X.509 attribute certificates and W3C Verifiable Credentials is available in clauses 5.2.1 and 5.2.2).

By enrolling for atomic (Q)EAAs on demand it is possible to achieve verifier unlinkable attestations which results in an unused set of (Q)EAAs with new signatures that cannot be correlated with any previous signatures. Fully unlinkable (Q)EAAs are, however, not possible.

NOTE 1: If the atomic (Q)EAAs are issued batchwise to an EUDI Wallet, it is recommended to keep track of the atomic (Q)EAAs that have been used for presentations, and replace them with new atomic (Q)EAAs.

NOTE 2: Atomic attribute credentials cannot alone guarantee that the claims are paired properly in a presentation. For instance, if the user has a credential from the civil registry with an address, and one for their company they are the legal representative of, there is nothing preventing the user from creating a presentation that improperly pairs the company's address with the user's private car registration. Verifiers cannot trust that verifiable presentations containing multiple atomic attribute credentials are properly paired without additional mechanisms preventing improper pairing.

4.3 Multi-message signature schemes

4.3.1 The BBS+ signature scheme

4.3.1.1 Background: Boneh-Boyen-Shacham (BBS04) signature scheme

Initially, the term group signatures was introduced in 1991 by Chaum and van Heyst in their paper "Group signatures" [i.54] as a scheme that provides anonymity for signers. This means that any member of the group can sign a message, but the resulting signature keeps the identity of the signer secret. The Stanford cryptography researchers Boneh, Lynn and Shacham continued the research on group signatures with respect to bilinear pairings, and published the results in their paper "Short signatures from the Weil pairing" [i.26] in 2001, where the Weil pairing refers to elliptic curve bilinear pairings [i.158].

Three years later the BBS04 signature scheme was published 2004 in the paper "Short Group Signatures" [i.25] by Boneh, Boyen and Shacham, who also named the BBS04 signature scheme after their initials. The BBS04 is a group signature scheme that is based on the Strong Diffie-Hellman assumption in conjunction with bilinear groups called the Decision Linear assumption.

4.3.1.2 Introducing the BBS+ signature scheme

Based on the BBS04 signature scheme, the cryptographic research has continued with BBS+, which allows for multi-messages to be selectively disclosed and signed with group signatures. One major contribution was Camenisch and Lysyanskaya and their 2004 work on signature schemes and anonymous credentials from bilinear maps [i.43]. The BBS+ signature scheme was described for the first time in 2006 by Au et al in the paper "Constant-size dynamic k-TAA" [i.10]. Furthermore, the BBS+ signature scheme is proven to be secure in the type-3 pairing setting in the paper "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited" [i.38] published by Camenisch et al in 2016.

The BBS+ signature scheme is a multi-message digital signature protocol, that proves knowledge of a signature while selectively disclosing any subset of the signed messages. Similar to CL-signatures (see clause 4.3.2.2), BBS+ signatures preserve the algebraic structure of the messages and rely on commitments. Specifically, the message $M = (m_1, m_2, \ldots, m_L)$ is used in a commitment as follows:

$A = (g_1 h_1^{m_1} h_2^{m_2} \ldots h_L^{m_L})^{(1/(x+e))}$, where *h_1*, …, *h_L* are generators of the group *G_1*.

The signature on *M* is *(A,e)*. The proof generation and verification then involves disclosing the messages and generators that the user wishes to present.

NOTE: The IETF CFRG BBS draft [i.116] differs from the above in subtle ways but the core selective disclosure mechanism is the same.

The BBS+ scheme allows for signing multiple messages whilst producing a single, constant size, digital (group) signature. BBS+ is a ZKP protocol, hence the BBS+ proofs do not reveal any information about the undisclosed messages or the original signature. A user who possesses a signature is able to generate multiple, unlinkable proofs that selectively disclose subsets of the originally signed messages, yet preserving the authenticity and integrity of the messages.

A user can generate a ZKP proof of knowledge of a valid BBS+ signature, which makes BBS+ signatures suitable in cases that seek to prevent linkability through the issuer's signature.

4.3.1.3 Overview of BBS+

The BBS+ signature scheme is illustrated in Figure 3.

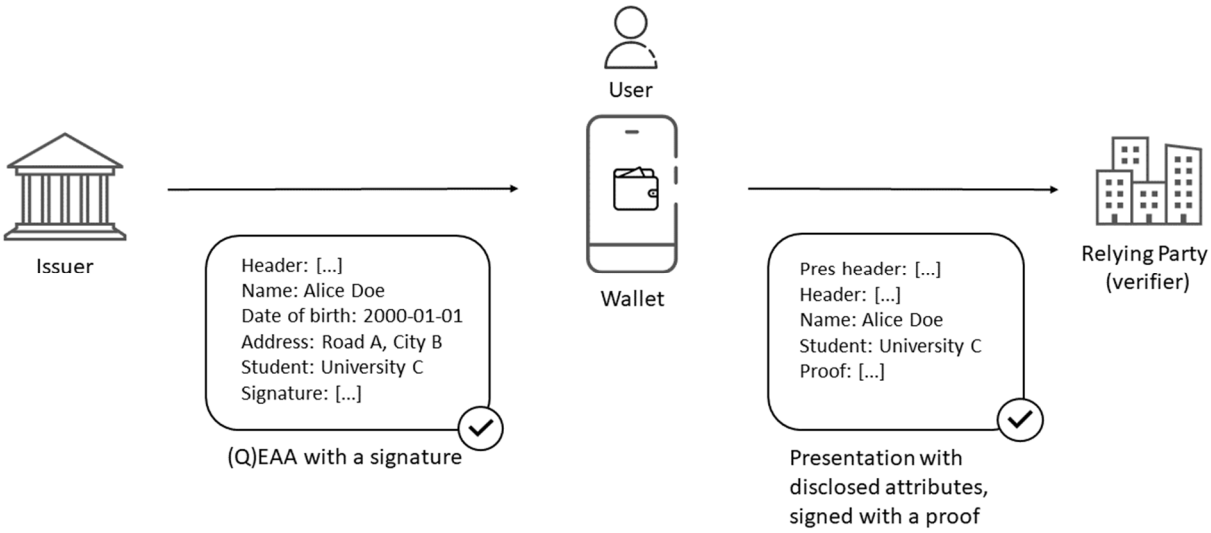

Figure 3: Overview of the BBS+ signature scheme

The issuer issues a (Q)EAA, with a header and a complete set of attributes, which is signed by the issuer. The (Q)EAA is stored in the user's wallet.

The user selects the attributes to disclose to a relying party, and the wallet generates a presentation with the disclosed attributes. The presentation contains a presentation header, the original header, the selectively disclosed attributes, and a proof. The proof reveals the user's knowledge of the original signature, but does not reveal the actual signature.

4.3.1.4 IETF CFRG BBS specification

The IETF Crypto Forum Research Group (CFRG) has created the internet draft specification "The BBS Signature Scheme" [i.116], which specifies an internet profile of the BBS+ scheme. The specification describes the following topics:

- Scheme Definition ([i.116], clause 3) defines the core operations and parameters for the BBS+ signature scheme.
- Utility Operations ([i.116], clause 4) defines utilities used by the BBS+ signature scheme.
- Security Considerations ([i.116], clause 5) describes a set of security considerations associated with the signature scheme.
- Ciphersuites ([i.116], clause 6) define the format of a ciphersuite.

More specifically, the IETF BBS+ draft specifies pairing-friendly ECC curves [i.118] alongside a concrete ciphersuite based on the BLS12-381 curve.

NOTE: The IETF CFRG draft specification has the title "The BBS Signature Scheme", although it describes the BBS+ scheme. The term BBS+ is however used throughout the present report to describe the multi-message signature scheme, whilst the term BBS04 describes the original single-message signature scheme.

In relation to the IETF CFRG BBS draft, DIF (Decentralized Identity Foundation) has initiated the project to specify "Blind Signatures extension of the BBS Signature Scheme" [i.66]. That draft specification defines an extension to the IETF CFRG BBS scheme, which will enable blind signing capabilities in order to provide the ability for a signer to blindly sign a set of messages.

4.3.1.5 Cryptographic analysis of the BBS+ signature scheme

In a post-quantum world, SDH algorithms based on bilinear pairings are vulnerable against quantum computing attacks [i.193]. This is an identified weakness of the BBS+ signature scheme, which has been described in a cryptographic review [i.193] prepared for the U.S. Department of Homeland Security [i.193]. The report [i.193] claims that BBS+ signatures are not standardized by NIST, and are unlikely to be standardized, since they rely on ECC with BLS12-381 curves that are not considered quantum-safe in a post-quantum world. The European standardization organization SOG-IS has not approved the BLS12-381 [i.28] curves either. The U.S cryptographic review [i.193] gives the following recommendations for the IETF CFRG BBS draft specification to move closer to government compliance: use the SHAKE256 hash function from SHA-3 and an approved random number generator in the BBS+ signature implementation.

While the strong Diffie-Hellman assumption is not quantum resistant, the threat from an attacker utilizing a quantum computer is more difficult to assess. In general, the parts of a BBS+ secured (Q)EAA that are ZKP are secure against a computationally unbounded adversary, whereas the parts that can be attacked based on public knowledge (e.g. a signature or a public key) need to either be frequently rotated, used once only, or replaced with quantum resistant alternatives. Put differently, an attacker can use a quantum computer to reveal the signer's private key from the public key and thereafter forge proofs and signatures. But an attacker will not be able to break data confidentiality, meaning that undisclosed messages are safe in a post-quantum world, as are undisclosed signature values. For a more general discussion on the Post Quantum Computer implications, see clause 9.

The BBS+ signature scheme is fully unlinkable (i.e. to issuers, verifiers, and any other party, when blinded).

4.3.2 Camenisch-Lysyanskaya (CL) signatures

4.3.2.1 Introduction to CL-signatures

In their paper "A signature scheme with efficient protocols" [i.40] (2002), Camenisch and Lysyanskaya introduce the CL-signature. The authors explicitly sought to design signature schemes that would be "suitable as building blocks for other applications".

Of particular relevance to this text is that the CL-signature allows for the implementation of two additional protocols. The first protocol is a secure multiparty computation protocol that allows an issuer to issue a signed attestation to the user, without the issuer learning all the message content or the final signature value. The ability for a signer to obliviously sign a user provided commitment to a message is enables, among other things, the user to convenience a verifier that two attestations were issued to the same identity subject simply by providing an equality proof between the two (blinded) commitments in the two attestations. Relatedly, it allows the user to generate a proof of possession of the commitment value in a privacy preserving way. The second protocol enables the user to prove possession of a, potentially hidden and blinded, message-signature pair (in CL-signatures, this proof is done in a ZKP manner). This ability for the user to present different looking presentations based on the same underlying issuer signed attestation is an important property when seeking to achieve privacy across distinct authentications.

Together, the two protocols above are introduced to achieve what Camenisch and Lysyanskya describe as an anonymous credential system. Such a system has two important requirements:

1) The user is required to demonstrate to a verifier that they possess the right attributes for a specific service, without the verifier being able to infer anything other than the fact that the user has the right attributes.

2) The user is required to obtain attribute attestations without revealing their identity to the issuer (in the paper "A signature scheme with efficient protocols" [i.40], the authors consider the user's secret key to be equivalent to the user's identity).

A signature scheme that can meet the above two requirements is one that allows the design of protocols that can prove statements in the form of "I have a valid signature" and where these signatures are over blinded committed values.

4.3.2.2 The CL-signature scheme

CL-signatures enable the signing of messages without affecting the message's algebraic structure; a property that allows a user to prove statements about messages even if these messages are hidden in some way (e.g. using a commitment).

For key generation, the first CL-scheme relies on a special RSA modulus n=pq, where (p, q) are safe primes, and the quadratic residues mod n (a,b,c). The public key is (n,a,b,c) and the secret key is (p). The message space consists of the integers in range $[0,2^{l\_m})$ for the parameter l_m. The signing algorithm takes as input a message m, selects a random prime number e and a random value s of suitable lengths (the paper "A signature scheme with efficient protocols" [i.40] details how to select the proper parameters) and computes the value v such that $v^e = a^m b^s c \ (mod\ n)$. The signature verification is done using the tuple (e,s,v), where it is the user that completes the value for s based on input from the issuer, and the message m by checking that $v^e = a^m b^s c \ (mod\ n)$ and that e is within the suitable range.

Later versions rely on bilinear pairings and are more efficient.

As aforementioned, the CL-signature scheme preserves the message's algebraic structure. As such, when signing a block of messages, $(m_1, m_2, \ldots, m_L)$ it is not permitted to simply sign the hash over the block of messages $H(m_1, m_2, \ldots, m_L)$ as this would make it impossible to both prove relations among the message components, the oblivious signature demand, and to prove predicates. Instead, the previous signing algorithm is modified to allow for multi-message signing as follows:

$$v^e = a_1^{m_1} a_2^{m_2} \ldots a_L^{m_L} \ \ b^s c \ (mod\ n)$$

As such, in a sense, each message is signed with an individual key by the issuer, and all the signatures are combined to a single one. Next it will be described how the CL-signature scheme enables selective disclosure.

4.3.2.3 The CL-signature scheme and selective disclosure

In essence, the CL-signature includes a commitment vector of messages $a_1^{m_1} a_2^{m_2} \ldots a_L^{m_L}$. The following characteristics can now be observed:

- All the quadratic residues are public.
- The commitment $a_i^{m_i} \ (mod\ n)$ prevents the verifier from learning m_i as long as solving the DLP in that group is hard.
- The user can present any combination of the commitment and the cleartext message.

The last point is what enables selective disclosure. Basically, the user will present in cleartext all the messages they wish to reveal, and the commitments to the messages they wish to keep secret. For instance, if a user wants to present m_1 but keep m_2 hidden, the user would present $((a_1, m_1), a_2^{m_2})$.

4.3.2.4 The CL-signature scheme, predicates, and knowledge proofs

Since the algebraic structure of the messages is preserved, it is possible to generate various proofs using CL-signatures.

In their original paper, Camenisch and Lysyanskya list the following protocols known to be secure under the strong RSA assumption:

- Proof of knowledge of discrete logarithm representation modulo a composite. Under specific conditions, this can be used to prove knowledge of exponents $(m_1, m_2, \ldots, m_L)$ in the commitments $a_1^{m_1} a_2^{m_2} \ldots a_L^{m_L}$ without revealing the exponents.
- Proof of knowledge of equality of representation modulo two (possibly different) composite moduli. This one is similar to the one above, but can prove knowledge of exponents even if the bases are different and the composite moduli are different.
- Proof that a committed value, $g^{ab} h^{r_3} (mod\ n)$, is the product of two other committed values, ($g^a h^{r_1} (mod\ n)$, $g^b h^{r_2} (mod\ n)$), without revealing any of the values.
- Proof that a committed value, $g^x h^r (mod\ n)$, lies in a given integer interval $a \leq x \leq b$. This builds on other known proofs that a committed value is a square (i.e. a positive number) and greater than or equal to proofs.

The above support the various predicate proofs that attestation systems based on CL-signatures are capable of, set (non-) membership tests, enable the property where the user can provide a proof of a valid signature as opposed to presenting the signature itself, and allows the user to request a signature over blinded messages. By extension, these properties provide unlinkability for the user as issuer and verifiers cannot collude to track use of an attestation.

EXAMPLE: A positive number proof can be easily constructed using other proofs. Lagrange's four-square theorem states that every natural number can be represented as the sum of four non-negative integer squares. Remember that there exists a way for the user to prove that a committed value is a square. A user could then send over the commitments to the square values, together with their corresponding proofs. The verifier can then easily check that another number is a positive number using the four commitments of a square number proof.

Recently, Thomas Groß extended the CL-signature scheme to obtain a signature on a committed graph and demonstrated that there exists a proof system on graph 3-colorability, meaning that there exists a CL proof system for all NP problems. However, like the common proof that every NP problem has a ZKP based on 3-colorability, this does not yield an efficient protocol for practical statements.

4.3.2.5 Cryptographic analysis of the CL-signature scheme

Since the first CL-signature scheme is based on the strong RSA assumption, and later versions are based on bilinear-pairings, they are not considered as being plausible quantum-safe in a post-quantum world. The CL-signature schemes are also not possible to construct using SOG-IS approved inputs. As with BBS+ signatures, the data confidentiality properties of a CL signatures remain safe even against a computationally unbounded attacker, but such an attacker can recover the signer's private key and forge signatures and proofs. For a more general discussion on the Post Quantum Computer implications, see clause 9.

The CL-Signature scheme is fully unlinkable when blinded.

4.3.3 Mercurial signatures

Mercurial signatures [i.62] cater for privacy preserving schemes, such as anonymous credentials, delegatable anonymous credentials, and related applications. They allow a signature s0 on a message m0 under a public key pk0 to be transformed into a signature s1 on an equivalent message m1 under an equivalent public key pk1. For example, pk0 and pk1 may be unlinkable public keys of the same user, and m0 and m1 may be unlinkable pseudonyms of a user to whom some capability is delegated. Mercurial signatures were presented by Crites-Lysyanskaya [i.61] in 2019.

Mercurial signatures are based on Decisional Diffie-Hellman (DDH) over equivalent groups, and are therefore not considered as plausible quantum-safe cryptography in a post-quantum world. Mercurial signatures can however be considered to be secure in a pre-quantum world, and the ZKP of knowledge of Mercurial signatures that are generated in a pre-quantum world will also remain plausible quantum-safe in a post-quantum world (see clause 4.3.1.5).

The Mercurial signature scheme is fully unlinkable when blinded.

4.3.4 Pointcheval-Sanders Multi-Signatures (PS-MS)

Pointcheval-Sanders Multi-Signatures (PS-MS) [i.176] have certain properties that can be used for distributed privacy-preserving Attribute Based Credentials (dp-ABC). The PS-MS signatures are based on a variant of CL-signatures with pairing-friendly curves such as BLS12-461. There is a formal definition of PS-MS signatures by Camenisch et al in the paper "Short Threshold Dynamic Group Signatures" [i.39] (2020), which are secure under bilinear group model and random oracle model.

An dp-ABC scheme based on PS-MS signatures has been designed by García-Rodríguez et al in their paper "Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures" [i.94] (2021).

The workflow of a dp-ABC scheme is illustrated in Figure 4.

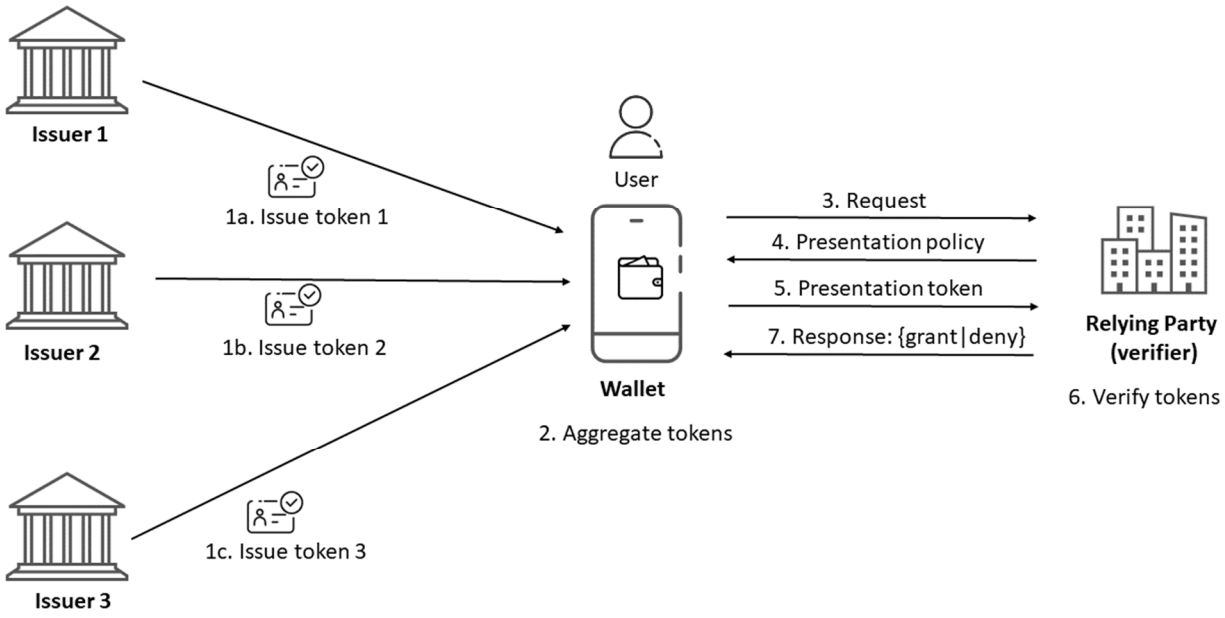

Figure 4: Overview of PS-MS signatures used for dp-ABC flow

More specifically, the PS-MS signatures are used when aggregating the issued tokens in step 2. Selective disclosure and unlinkability is an integral feature of the PS-MS signatures.

NOTE: The identity systems Idemix (clause 6.6.1) and U-Prove (clause 6.6.2) are also based on p-ABC schemes, however, they are based on CL-Signatures and the DLP.

Since the PS-MS signature scheme is based on bilinear-pairings, it is not approved by SOG-IS or considered as being plausible quantum-safe cryptography in a post-quantum world. ZKP of knowledge of PS-MS signatures can however be considered to be secure in a pre-quantum world, and the ZKP of knowledge of PS-MS signatures that are generated in a pre-quantum world will also remain plausible quantum-safe in a post-quantum world (see clause 4.3.1.5).

The PS-MS signature scheme is fully unlinkable when blinded.

4.3.5 ISO standardisation of multi-message signature schemes

4.3.5.1 ISO/IEC 20008 - Anonymous digital signatures

The ISO/IEC 20008 [i.143] series specify anonymous digital signature mechanisms (algorithms) as follows:

- ISO/IEC 20008-1 [i.143] specifies a general model with principles, entities, processes, and requirements for anonymous digital signature mechanisms.
- ISO/IEC 20008-2 [i.143] specifies anonymous digital signature mechanisms, for which a verifier can use a group public key to verify a digital signature. For each mechanism, this part of the standard specifies the processes for generating group member signature keys, producing signatures, verifying signatures, opening signatures, linking signatures, and revocation of group members.
- ISO/IEC 20008-3 [i.143] extends ISO/IEC 20008-2 [i.143] by specifying anonymous digital signature mechanisms using multiple public keys.
- ISO/IEC 20008-2 [i.143]/AMD1 and ISO/IEC 20008-2 [i.143]/AMD2 are amendments to ISO/IEC 20008-2 [i.143] with additional details about certain mechstandardizationanisms.

More specifically, ISO/IEC 20008-2 [i.143] mechanism 3 specifies the cryptographic primitives of a qSDH scheme, which corresponds to BBS04 with single messages as described 2004 by Boneh, Boyen and Shacham in their paper on short group signatures [i.25]. Since ISO 20008-2 [i.143] mechanism 3 is designed as a single message signature scheme, it requires an extension to support multi-message signature protocols.

BBS+ is an extension of BBS04 (including the Pedersen commitments) to cater for a multi-message signature scheme. Formally, BBS+ relies upon the same security model as the qSDH assumption that is described in ISO 20008-2 [i.143] mechanism 3. More precisely, it is shown (for example in [i.13]) that if an attacker can forge BBS+ signatures then it can also forge BBS04 signatures. In other words, if the BBS04 cryptographic primitives are deemed secure as specified in ISO 20008-2 [i.143], so is BBS+.

Furthermore, the Pointcheval-Sanders Group Signature scheme (PS-GS) [i.176] is specified in ISO 20008-2 [i.143] amendment 2.

4.3.5.2 ISO/IEC PWI 24843 - Privacy-preserving attribute-based credentials

In October 2023, ISO/IEC JTC 1/SC 27 initiated the Preliminary Work Item (PWI) 24843 "Privacy-preserving attribute-based credentials" [i.144] to formally standardize the multi-message signature scheme version of ISO/IEC 20008-2 [i.143].

At the time of writing in April 2024, this ISO/IEC Preliminary Work Item (PWI) 24843 is a preliminary study to investigate the interest in ISO/IEC JTC 1/SC 27 for creating a standard on privacy-preserving attribute-based credentials mechanisms. The final agreement to start the work on this standard is planned to be made in April 2024, with the intent to publish a first working draft in October 2024.

In other words, the ISO/IEC PWI 24843 proposal has the potential to result in an ISO standardized version of BBS+ as well as other multi-message signature schemes capable of both selective disclosure and full unlinkability.

4.3.5.3 ISO/IEC CD 27565 - Guidelines on privacy preservation based on ZKP

In addition to the aforementioned ISO standards on anonymous digital signatures and the PWI on privacy-preserving attribute-based credentials, ISO/IEC JTC 1/SC 27 are also working on the common draft ISO/IEC CD 27565 [i.150] "Guidelines on privacy preservation based on zero knowledge proofs". This draft document provides guidelines for how to use ZKPs to improve privacy by minimizing unnecessary information disclosure when sharing personal data between organizations and users.

More specifically, Annex C of ISO/IEC CD 27565 [i.116] includes an example of selective disclosure by using BBS+, with a reference to the IETF CFRG BBS draft specification.

4.3.6 Extensions of multi-messages signature schemes

The multi-messages signature schemes described in clauses 4.3.1 to 4.3.4 are based on the classic approach for building (Q)EAAs from a set of advanced cryptographic mechanisms such as BBS+, CL or PS-MS signatures. While this approach does support selective disclosure, it comes with the cost of concealing the undisclosed attributes in a zero-knowledge proof whose complexity grows linearly with the number of such attributes. In order to minimize the size of the (Q)EAAs and their verifiable presentations, more elaborate approaches have been proposed for BBS+ and PS-MS, where undisclosed attributes have no impact on the proof size, which is beneficial for selective disclosure. Below are three cryptographic research papers that describes such approaches:

- "MoniPoly: An Expressive q-SDH-Based Anonymous Attribute-Based Credential System" [i.191] published by Syh-Yuan Tan and Thomas Gross (2020).
- "Efficient Redactable Signature and Application to Anonymous Credentials" [i.184] published by Olivier Sanders (2020).
- "Improving Revocation for Group Signature with Redactable Signature" [i.185] published by Olivier Sanders (2021).

4.4 Salted attribute hashes

4.4.1 Overview of salted attribute hashes

Salted attribute hashes are a widely deployed concept in many solutions capable of selective disclosure. The salted hash approach computes a cryptographic digest over at least one attribute and an attribute specific random salt, e.g. a SHA256 digest over a concatenation of a salt and an attribute, SHA256 (salt||attribute).

In the context of a (Q)EAA, each attribute is salted and a hash digest is included as a value in the attestation. The specific way to include the digest in the attestation varies between various solutions. Some include salted attribute hashes in an indexed list, others in an array, others structure these as a Directed Acyclic Graph (DAG). Common to all is that the issuer needs to issue the (Q)EAA with the attributes in clear text, along with the logical ordering of salted attribute hashes.

An illustrative example of salted attribute hashes is illustrated in Figure 5.

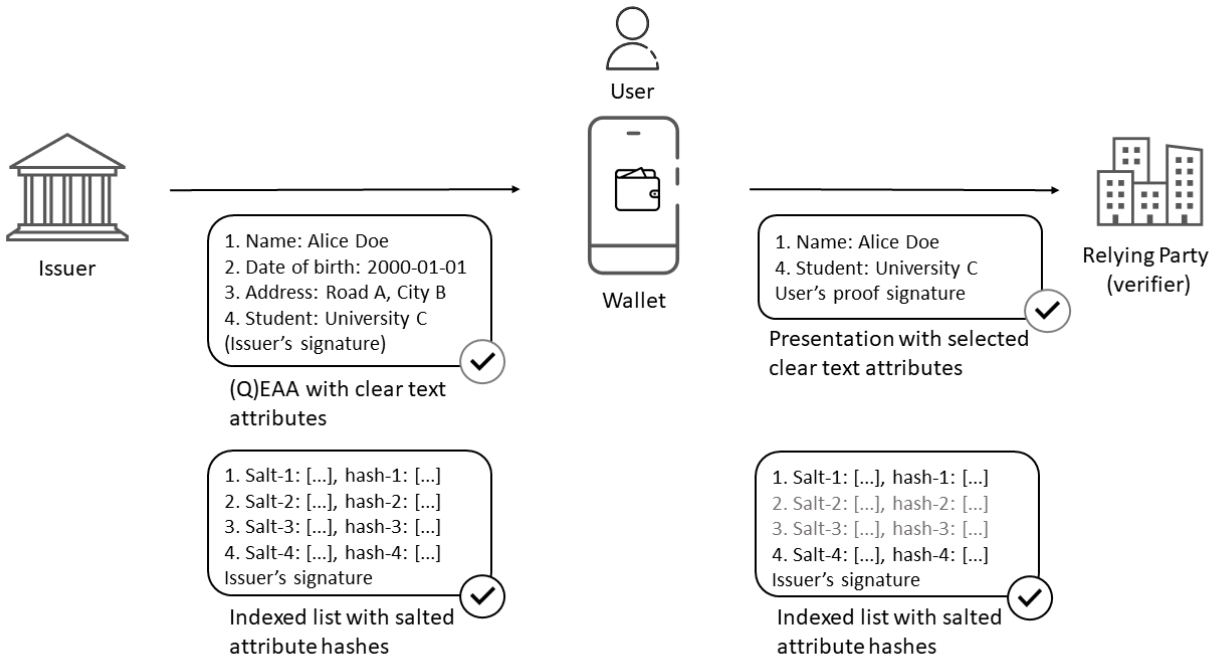

Figure 5: Illustrative example of salted attribute hashes

In the example above, the issuer issues a (Q)EAA with all attributes in clear text. The issuer also issues an indexed hash list in which each (Q)EAA attribute is represented as a key (index), a random salt, and a hash value over the salt and attribute. The (Q)EAA and indexed hash list are signed by the issuer.

NOTE 1: Exactly how the random salts are combined with the attributes and hashed, and how the lists of salted attributes hashes are signed by the issuer, differs between various specifications and standards. The relevant standards that are described and analysed in the present document are ISO mDL MSO (see clause 7.2) and IETF SD-JWT (see clause 7.3).

NOTE 2: The salts may be included in the indexed list with salted attribute hashes, or be provided separately from the indexed list. If the salts are provided separately (on a need to know basis) to the verifier, this is the most privacy preserving option.

NOTE 3: The (Q)EAA can be either signed or unsigned. It depends on the standard if the (Q)EAA is signed or not.

EXAMPLE 1: The ISO mDL mdoc (with the attributes) is unsigned, whilst the corresponding ISO mDL MSO (with the salted attribute hashes) is signed by the issuer.

EXAMPLE 2: The W3C Verifiable Credentials (with the attributes) is signed, and the corresponding IETF SD-JWT (with the salted attribute hashes) is also signed.

The (Q)EAA and indexed hash list are stored in the user's wallet. The user selects the attributes to disclose to a relying party, and the wallet generates a presentation with the disclosed attributes; the user signs the presentation with its proof key.

The wallet submits the presentation with selected attributes (in clear text) along with the indexed hash list. The relying party parses out the salted hashes from the indexed hash list, and compares them with the salted hashes of the presented attributes.

Solutions based on the concept of salted attribute hashes have been standardized as IETF SD-JWT and ISO mDL MSO. More information on the specific formats IETF SD-JWT and ISO mDL MSO that use salted attribute hashes for selective disclosure is available in clauses 5.4.2 and 5.4.3.

4.4.2 Issuance phase

The issuance phase of this selective disclosure scheme is in principle based on the following algorithm:

1) Parse out each attribute from a user's (Q)EAA.

2) Concatenate each attribute set with a salt, denoted as (salt||attribute).

3) Hash each (salt||attribute), denoted as hash(salt||attribute).

4) Order all the hash(salt||attribute) values and the salts in e.g. an indexed hash list (could also be an array, DAG etc.), which is signed. The indexed hash list can be expressed as this formula: signed({key-1, salt-1, hash(salt-1||attribute-1)}, … {key-*n*, salt-*n*, hash(salt-*n*||attribute-*n*)}).

5) Store the (Q)EAA in an EUDI Wallet along with the indexed list from step 4.

NOTE 1: The hash algorithm used in step 3 should be listed in the SOG-IS list of approved hash algorithms [i.188], such as SHA-256 or higher.

NOTE 2: The signature algorithm used in step 4 should be listed in the SOG-IS list of approved signature algorithms [i.188], such as ECDSA with BrainpoolP256r1.

NOTE 3: The signature format used in step 4 should allow for QSC algorithms. For example, JOSE and COSE allows for QSC algorithms.

4.4.3 Presentation and verification phase

When presenting selective disclosed attributes in the (Q)EAA along with the indexed list, the relying party can perform the following verification process:

1) The EUDI Wallet parses out the disclosed attribute with key-*x* from the (Q)EAA.

2) The EUDI Wallet submits the disclosed (Q)EAA attribute with key-*x* from step 1 along with the indexed hash list to the relying party. The indexed hash list has the format: signed({key-1, salt-1, hash(salt-1||attribute-1)}, … {key-*n*, salt-*n*, hash(salt-*n*||attribute-*n*)}).

3) The relying party verifies the signature of the indexed hash list from step 2. If the signature check fails, the verification process is stopped, else it continues at step 4.

4) The relying party parses out salt-*x* from the indexed hash list.

5) The relying party parses out hash(salt-*x*||attribute-*x*) from the indexed hash list.

6) The relying party concatenates the disclosed (Q)EAA attribute from step 2 with the corresponding salt-*x* from step 4, and hashes the result.

7) The relying party checks if the result in step 6 is equal to the hash(salt-*x*||attribute-*x*) from step 5. If the values match, the verification process has succeeded.

4.4.4 Salted attribute hashes and unlinkability

4.4.4.1 General criteria of unlinkability for salted attribute hashes

By default, salted attribute hash based approaches cannot offer any unlinkability. Both issuers and verifiers, and any other party, can trivially link together disclosures and attestations based on either the signature value or the salt values. There are, however, some workarounds that can offer verifier unlinkability at added cost for issuers.

To achieve verifier(s) unlinkability, two important criteria have to be met. The first relates to the salts in issuance step 2. Each salt value should be a randomly generated unique value and each attestation should only be presented once. Consequently, the indexed list in issuance step 4 is also updated. Using unique salts will prevent the verifier(s) from comparing the issuer signatures and/or salts of previously disclosed attributes.

NOTE: Using unique salts, an issuer can always uniquely identify a user from a single disclosed salted attribute even without the attestation signature. Consequently, salted attribute hashes represent a tradeoff between issuers' and verifiers' ability to link together attestation usage. That tradeoff is unproblematic in contexts where issuers are assumed trusted, but represents a great risk in contexts where issuer collusion is possible.

Hence, schemes based on uniquely salted attribute hashes can be designed to be verifier unlinkable but do not support fully unlinkable attestations.

The second criteria important for verifier(s) unlinkability relates to the information about the key material that the user can prove control over. Two attestations that contain the same user public key are trivially linked. To prevent such linkability, batch issuance requires the user to generate a unique public key for each attestation.

4.4.4.2 Hierarchical Deterministic Keys and blinded key proof of possession

One way to generate unique public keys for each attestation is to rely on Hierarchical Deterministic Keys (HDKs). With HDKs, keys are not generated at random but deterministically as part of a cryptographically linked chain. The generated keys are also hierarchical in that the keys on a particular level, *i*, contain no information whatsoever about the keys in the level above them, *i-1*, but can be used to generate new keys in a level below them, *i+1*.

The major benefit of HDKs is that the user only has to manage a single private public key pair. For instance, the user only has to communicate this single public key to the issuer, who can then derive all other public keys it needs deterministically. These unique public keys can then be used for the batch that the issuer issues. Each of the user's derived public keys will be unique and can be designed to be unlinkable. The keys are unique and act as regular private public keys. A symmetric secret between the issuer and the user is used as input to the HDK, ensuring that only the issuer and the user can generate the derived keys.

The issuer does not need to save any of the public keys it generates; it only needs to save the single public key sent to it by the user. Correspondingly, the user does not need to save any other private keys beyond the one that belongs to the public key it sent to the issuer. The user can derive the required private key when asked to prove control over the private key behind a particular derived public key.

There are several options for HDK. One widely adopted HDK is the one used by BIP-32 [i.23] compatible cryptocurrency wallets. While not necessarily suitable for a PID/(Q)EAA context, BIP-32 is helpful to describe how it works since most HDKs work in a similar fashion. Specifications targeting PID/(Q)EAAs are underway with one notable mention being the proposed IETF draft "The Asynchronous Remote Key Generation (ARKG) algorithm" [i.115]. In BIP-32, the key derivation function depends on the following two main functions (serialization and conversion functions are ignored for brevity):

- `random(l)`: Generate a cryptographically secure pseudorandom byte string of length `l` bytes. Suitable values for n are between 128 to 512 bits.
- `HMAC-SHA-512(key, data)`: Generates a 512 bit cryptographic hash digest over a keyed message.
- `contact(a,b)`: Concatenate byte strings.

For a suitably selected curve, with order n and generator G, the master private key can be computed as follows (note that BIP-32 uses different values for the generation of `I`):

```
1. seed = random(32)
2. I = HMAC-SHA-512(key=b'', data=seed)
3. master_private_key, master_salt = I[:32], I[32:]
4. master_public_key = master_private_key * G
```

Derived keys can be either derived from the parent public key or parent private key. For the first level, the master keys and master salt act as parents. For the parent public key to child public key derivation:

```
Input:

   -  parent_public_key, the encoded parent public key from which to derive new keys
   -  parent_salt, the encoded salt value for the parent level
   -  i, child number

Output:

   -  i_salt, the rightmost 32 bytes of the HMAC-SHA-512 digest is the corresponding
      salt for the derived key pair
   -  i_public_key, the returned public key as constructed by the leftmost 32 bytes of
      the HMAC-SHA-512 digest

def child_key_derivation_pub(parent_public_key, parent_salt, i):

  I_i = HMAC-SHA-512(key=parent_salt, data=concat(parent_public_key, i))

      i_salt = I_i[32:]

      i_public_key = (I_i[:32] * G) + parent_public_key

      return i_public_key, i_salt
```

The above HDK allows the issuer to generate any number of public keys from a single public key, and the user to compute the corresponding private key. The user follows the same function steps as the Issuer, but instead of computing `i_public_key` they compute `i_private_key = (I_i[:32]+parent_private_key) % p` where `p` is a field parameter for the chosen curve. The above function does not include serialization and a way to structure the derived keys and is only meant to illustrate the potential for the Issuer to deterministically derive new unlinkable public keys from a single public key.

While interesting, there are two points to note with BIP-32. The first has to do with the generation of the salt value. In the above example, the salt needs to be communicated to the issuer (in BIP-32 this is done using a 78 byte extended public key). This is neither necessary nor ideal in a setting where it is assumed that a relationship exists between issuers and users.

An alternative way to generate a salt would be to rely only on regular public keys and let the issuer and user derive a deterministic shared secret using ECDH. The shared secret can then derive a derived key using a suitable KDF (e.g. HKDF-Extract). It is then possible to use HDKF-Expand together with an info parameter that includes a domain separation tag, an index, and other values to generate derived keys. Note that this example does not use HMAC directly. Unlike BIP-32, where the HMAC input is a 32 byte uniformly random seed, the ECDH output is not suitable for HMAC. It is however possible to use HMAC on the derived key from a HKDF-Extract. An approach based on ECDH and HKDF would create a different key derivation hierarchy that is arguably more suitable for an issuer-holder relationship. Here, the derived keys are grouped by issuer and index.

The second point with BIP-32 is that it requires the user to perform an additive operation on the private key and to sign with the resulting sum. If the addition required for `i_private_key` and the subsequent signature can be performed and generated in a secure cryptographic environment, then the HDK may be suitable for use at Level of Assurance High. However, most secure cryptographic environments do not support the additive or multiplicative operation required. Support may be added in the near future given how ARKG and the related IETF draft "Key Blinding for Signature Schemes (KBSS)" [i.117] that both require additive and/or multiplicative operations.

The additive and/or multiplicative signature key blinding approach is one out of (at least three) possible ways to use HDK output for proof of possession. The second asymmetric way to use HDK output for proof of possession is by using a discrete logarithm based signature with a malleable signature that enables operations that change the private key. Of the discrete logarithm based signatures that support the technique, ECDSA and ECSDSA are both SOG-IS approved.

In ECDSA, the key malleability stems from the way the `s` value is computed in a signature pair `(r,s)`. Given a generator `G`, private key `x`, a random integer, `k`, and `r` set to the x-coordinate of `k*G`, the value `s = k^(-1) * (H(m) + r*x)` can be blinded using a HDK serving as a blind, `b`, to generate the value `s_blind = k^(-1) * b(H(m) + r*x)` to generate a valid signature over `H(m)*b` using the private key `x*b`, which can be verified using the public key `(x*b)G`. To transform the ECDSA signature to a valid signature over H(m) with blinded private key (`x*b`), the value of `H(m)` is multiplied with the inverse of the blind before the raw ECDSA. The inventor, Eric Verheul, refers to this approach as SECDSA in the paper "SECDSA: Mobile signing and authentication under classical 'sole control'" [i.194], and a patent is pending for its use for blind key signing.

In ECSDSA, the blinding can be done either using multiplicative blinding or additive blinding. Additive blinding is far simpler and only requires a single operation on the Schnorr signature. With a Schnorr signature `(s,e)`, where `e` is derived from hashing the message together with the x-coordinate of `k*G`, and where s is computed as `s=k-xe`, a blind, `b`, can be added to generate `s_blind=s-be`. A signature `(s_blind, e)` can be verified with a blinded public key `(x+b)G`. A blinded ECSDSA is similar to how multisignature works in Bitcoin, but for PID/(Q)EAA the set of allowed keys is that from the HDK and the user derives the two key shares required for the signature.

A third option to complement the asymmetric option that requires support in the secure cryptographic environment for addition and/or multiplication (as required by BIP-32, ARKG, and KBSS), and the asymmetric options of SECDSA (forthcoming patent) and ECSDSA that are supported by existing secure cryptographic environments, is a symmetric proof of possession that relies on ECDH.

In blinded ECDH, the user first computes the scalar multiplication between the verifier public key and the blind, `b`. The resulting point is then sent as the public point input to a secure cryptographic environment protected ECDH. The resulting shared key is then input to a KDF. The challenge can either be input directly to a suitable KDF, e.g. HKDF, using the info parameter, or as the data to a HMAC that uses the HKDF derived key as key material. The verifier can compute the same shared secret using the blinded public key and its private key.

Note that:

1) in all three cases, the issuer can compute the blinded public key using only the user's public key;

2) it is only the user who can compute the corresponding blinded private key or use this key to generate a proof a possession;

3) generating the blinded public key requires access to either the issuer or user private key; and

4) the derived blinded public keys are cryptographically linked to the private key that can be protected on a secure cryptographic environment.

4.4.5 Cryptographic analysis

The (Q)EAA and indexed hash list are separate objects that can be signed with cryptographic algorithms that are approved by SOG-IS [i.188]. In other words, there are no specific requirements on ECC curves for bilinear pairings.

This concept also caters for the (Q)EAA and indexed hash list to be signed in the future with QSC algorithms as discussed in the IETF report "JOSE and COSE Encoding for Post-Quantum Signatures" [i.119].

4.4.6 Predicates based on computational inputs

Salted attribute hashes do not support dynamic calculation of predicates (e.g. to compute a proof for age over 18 given only the birth date and current date). The recommendation is to include Boolean claims such as `"age_over_NN": "True"`. Hence, salted attribute hashes do not support dynamic predicates according to the theoretical definition.

However, there is a possibility for the issuer to sign the parameters and the inputs to an inequality test. This would enable the user and the verifier to compare numbers and perform range proofs. For an (Q)EAA system, there is normally a) a trusted issuer, and b) a limited need to perform operations between hashed values (thus eliminating the need for commitment homomorphism and the ability to perform general algebraic manipulations).

It is normally interesting to prove that an attribute claim satisfies a threshold or inequality and absolutely nothing else. Furthermore, there is a trusted issuer and there is also only the need to hide the exact amount of the values. Thus, ensuring the ZKP property may not be necessary.

EXAMPLE: The issuer could compute the commitment s = H (seed) and assign this to the user's birth year. The issuer then computes the commitment $c = H^k (salt | s)$, which is k repeated iterations of H. The value for k can be computed e.g. based on the maximum year supported in the calculation. The issuer includes s and c in the signed attestation both as disclosures (the user should never reveal s, only c). The user can now generate an age over 18 proof by constructing a hash chain where the length of the chain equals the k iterations used to arrive at the signed commitment c if and only if the user is above a certain age. Example code is provided in Appendix B. Research on efficient protocols for hash chain based range proofs is underway with one notable example being HashWires [i.213]. And variations of the technique exist that would allow a user to generate a valid age_over_N proof from an age_over_M proof where M > N. The algorithm for HashWires in combination with salted attribute hashes is described in clause 4.4.7.4.

4.4.7 HashWires

4.4.7.1 Introduction

In their 2021 paper "HashWires: Hyperefficient Credential-Based Range Proofs", Chalkias et al. [i.51] present a hash based protocol for performing inequality tests (and by extension range proofs) in contexts where a trusted issuer can sign commitments to computational inputs. The computational inputs in HashWires are a commitment *c* to a hash chain, and the parameter is the hashing algorithm used to create the chain.

HashWires are inherently less flexible than general ZKP inequality tests and range proofs, and do not support homomorphic operations on commitments. However, the commitment and proof conditions, together with the adversarial assumptions in their deployed contexts (e.g. cryptocurrencies), often makes ZKP inequality tests and range proofs unsuitable for resource constrained environments and unnecessarily complex given the presence of a trusted PID/(Q)EAA Provider (as opposed to self signed claims). Put differently, many existing ZKP inequality tests and range proofs were designed to cater for highly adversarial cryptocurrency contexts without any trusted parties or central authorities, and where the user self issues a signed intent to perform a certain transaction. In contrast, HashWires were designed to specifically cater for the needs of the issuer-holder-verifier model. The authors introduce the concept of "Credential-based range proofs" to distinguish these inequality tests and range proofs from their ZKP counterparts.

HashWires is based on the core idea that the trusted third party, i.e. the PID/(Q)EAA Provider, generates and signs the commitment needed for an inequality test. The idea to rely on a trusted third party to sign a commitment can be traced back to Rivest and Shamir's 1996 work on micro-payments. In their paper "PayWord and MicroMint: Two simple micropayment schemes" [i.181], Rivest and Shamir describe how issuer signed hash chains type commitments can be used for payments. A description of their original idea follows.

4.4.7.2 Using a hash chain for inequality tests

A fundamental building block in HashWires is hash chains. Given two collision-resistant hash functions (H, G), a maximum integer value N, and a random value r, the issuer computes the commitment $c = H^k(G(r))$. Here, $H^k(\cdot)$ represents k iterations of the function H such that the digest of H^i is the pre-image to H^{i+1}. The issuer signs c and sends (c, r) to the user (optionally also k). The user can now produce a hash chain of the same length as a threshold t by computing the range proof $\pi = H^{k-t}(G(r))$. The user signs a presentation containing (π) and the verifier checks if $c = H^t(\pi)$. If the check passes, the verifier knows that c is the commitment to some value $t \leq x$ but does not learn k.

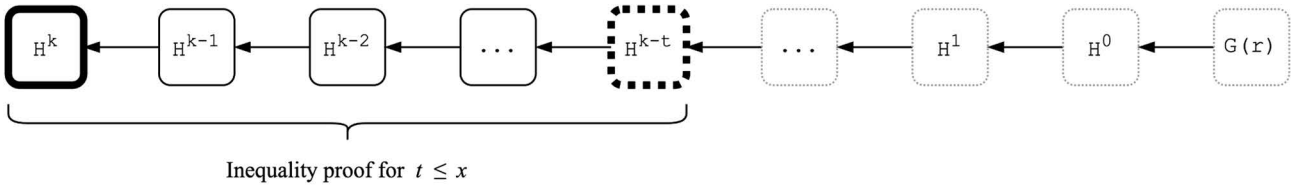

Figure 6: A hash chain based inequality test

In Figure 6, the issuer signs the leftmost bold box representing the commitment $c = H^k(G(r))$. The user presents the dotted bold lined box representing the threshold value $\pi = H^{k-t}(G(r))$. The verifier accepts π as a proof for the inequality $t \leq x$. Note that for an age proof, the value H^0 should represent the user's actual age k at the time of issuance and that H^k represents the minimum age value 0.

NOTE 1: The hash functions (H, G) should be listed in the SOG-IS table of agreed hash functions [i.188].

NOTE 2: The digital signature scheme should be listed in the SOG-IS table of agreed signature schemes [i.188].

NOTE 3: The use of digital signatures that are QSC should be possible.

NOTE 4: The verifier does not learn the value k, $G(r)$ and any $H^m(\cdot)$ where $m > t$.

NOTE 5: A single hash function with two different salts, or a keyed HMAC with two keys, are both alternatives to (H, G).

When considering non-negative integers, one obvious representation is that the H^0 digest represents the maximum value, and each subsequent digest represents a decrement by 1. The problem with that approach is that it does not scale. Take for instance age over or equal to proofs. Here, the user should be able to prove that their age is equal to or above 18 the very day they turn 18, but not before. A hash chain for 18 years in days requires roughly 6 575 digests. This is further exacerbated by the batch issuance requirement for PIDs and (Q)EAAs to prevent verifier collusion (the Provider would need to create a new hash chain for every attestation since the commitment would be correlatable even with a salt). Also, each verifier needs to recompute the threshold length of the chain at every presentation. With ~450 million EU citizens, and potentially multifold more inequality tests for age based services, optimization is required.

4.4.7.3 Using multiple hash chains for inequality tests

The optimization presented in the HashWires paper ensures that the commitment generation, proof and verification, and proof size all scale well even for very large *n*-digit numbers. The core idea is to rely on multiple hash chains. However, instead of representing decrements starting from the maximum number, each digest represents the commitment to the digits x_i of a number $x = x_n \cdot 10^n + x_{n-1} \cdot 10^{n-1} + \ldots + x_1 \cdot 10^1 + x_0$.

For instance, using the commitments to the coefficients in $22 = 2 \cdot 10^1 + 2 \cdot 10^0$ a user could generate a proof for the inequality $x \geq 10$. Note, however, that the user would not be able to use that commitment to prove $x \geq 13$ without revealing a lot more information than necessary (more specifically, the user would need to reveal commitments to 20).

Chalkias et al. here describe the idea of Minimum Dominating Partitions (MDP) to address the above problem. In the HashWires paper, there is a formal definition of MDP, which relies on the idea that a number x dominates another number y if each digit $x_i \geq y_i$. The authors present an algorithm that takes a non-negative integer as input and outputs one or more non-negative integers that represent numbers that dominate other numbers, where the collection of numbers output can dominate any other number in the entire range of the requested inequality.

A simpler explanation is that the MDP is generated using a recursive function that takes as input a number, and outputs the first number that the input cannot dominate. That new output number then becomes the new input number, and the MDP outputs the value it cannot dominate. For instance, using base 10, the number 84 can dominate $\{84,83,82,81,80\}$ but not 79. Subsequently, 79 can dominate all numbers down to 0. So the $MDP(84) = \{84,79\}$. Similarly, $MDP(3413) = \{3413,3409,3399,2999\}$.

Given a set of MDP partitions, the user can use hash chains to dominate any number that up to and including the first element by simply picking the element that can dominate the requested threshold value. For instance, given $MDP(3413) = \{3413,3409,3399,2999\}$ the user can use the $\{2999\}$ element to prove $x \geq 376$. When the user can use more than a single element from the MDP to dominate the threshold number, the user picks the number that reveals the least amount of information.

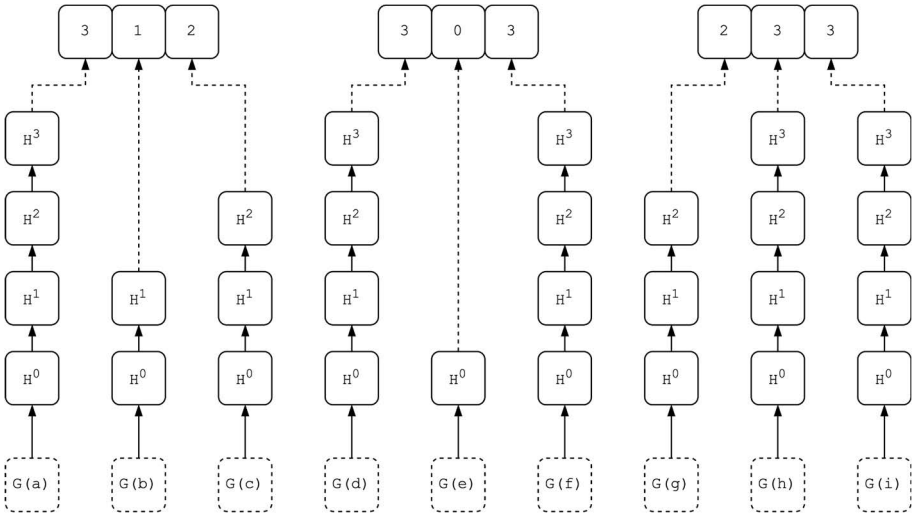

Figure 7: Basic HashWires commitment

Figure 7 illustrates a basic HashWires commitment to the number 312 in base 4 with $MDP_4(312) = \{312,303,233\}$. Each hash chain represents a commitment to a specific digit in each MDP partition.

A further optimization can be made by reusing the same hash chain for multiple different commitments. The idea here is to generate one hash chain per digit in the largest number, with the length of the hash chain being the largest value of any digit in any MDP partition.

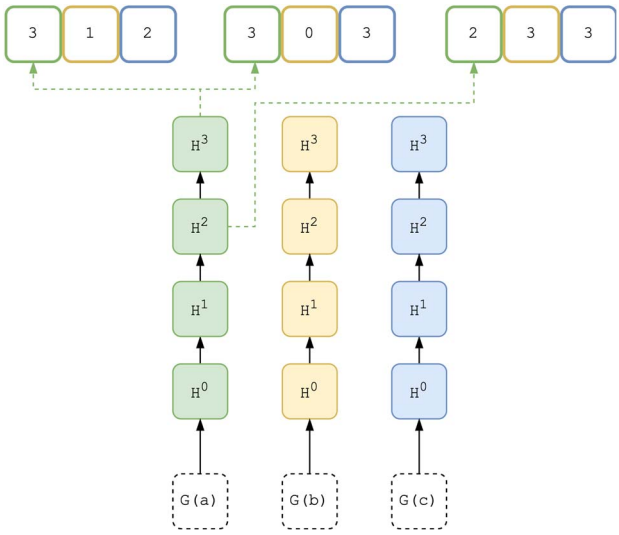

Figure 8: Optimized HashWires commitment

Figure 8 shows an optimized HashWires commitment to the number 312 in base 4 with $MDP_4(312) = \{312,303,233\}$. Each hash chain represents the commitments to the digit values of each partition. Green dotted line illustrates how the values are sourced for the third digit in each MDP partition. Hash chains are coloured to correspond to their commitments, i.e. the second digit in each MDP partition would source their commitment from the middle hash chain, and the first digit in each partition would source commitments from the rightmost hash chain.

The optimized HashWires approach is orders of magnitude more efficient than using a single hash chain. Specifically, the $MDP(6575) = \{6575,6569,6499,5999\}$ (18 years in days), requires $3 + 6 + 9 + 9 + 9 = 36$ hash operations (three for the seeds, and then 6 for the fourth digit, and then 9 for each subsequent digit). In fact, using base 10, the maximum possible number of hash chains will never exceed the number of digits multiplied by 10.

One concern with the optimized HashWires approach is that it may leak information about the partitions, and thus reveal the users actual number. To avoid such leaks, the authors of the HashWires paper suggest the use of an accumulator that can hide the actual commitments. While the use of an accumulator addresses the concern, it is also not necessary when the attestation format is capable of selectively disclosing the particular commitment that the user needs to prove the inequality, and when attestations are batch issued and used only once (that is not to say that the issuer cannot select to include the accumulator value as a selectively disclosable value).

4.4.7.4 Protecting optimized HashWires with SD-JWT or MSO

The MDP partitions leak information about the number in several ways. Therefore, it is important that the user only reveals the exact commitment that is required for the request threshold inequality proof. The original HashWires paper achieves this using an accumulator, but it is also possible to rely on the selective disclosure capabilities of SD-JWT and MSO. For reasons of readability, illustrative examples will be done using SD-JWT and without an accumulator, but the concept is equally applicable for MSO and every other salted attribute hashes based approach.

NOTE: Combining HashWires range proofs with selectively disclosed salted hashes of attributes is suggested by Peter Lee Altmann (Swedish Digitalization Agency) and Sebastian Elfors (IDnow) to the present ETSI technical report. The idea is not peer reviewed and is meant primarily to illustrate the idea of a PID/(Q)EAA Provider signing computational inputs and parameters to enable dynamic predicates e.g. inequality tests. With modifications, the proposal could enhance the ISO mDL MSO [i.140] and IETF SD-JWT [i.123] standards to cater for predicate proofs in addition to selectively disclosing claims.

Consider an optimized HashWire for an *n*-digit number, $HW = \{[c_n, c_{n-1}, \ldots, c_0], [r_n, r_{n-1}, \ldots, r_0]\}$ where c_i denotes the hash chain root for digit position *i* in each MDP partition for a value *x* and r_i denotes the seed used in $G(\cdot)$ to generate the first value of the hash chain for each digit position *i*. Each MDP partition is a combination of hash roots.

For instance, the $MDP(6575) = \{6575,6569,6499,5999\}$ would require four seeds, resulting in four hash chains, one for each digit. The corresponding hash chains lengths for $MDP(6575)$ are $6 \cdot 10^3 + 9 \cdot 10^2 + 9 \cdot 10^1 + 9$. More precisely:

- 6575 requires the commitment: $H^6(G(r_3)), H^5(G(r_2)), H^7(G(r_1)), H^5(G(r_0))$
- 6569 requires the commitment: $H^6(G(r_3)), H^5(G(r_2)), H^6(G(r_1)), H^9(G(r_0))$
- 6499 requires the commitment: $H^6(G(r_3)), H^4(G(r_2)), H^9(G(r_1)), H^9(G(r_0))$
- 5999 requires the commitment: $H^5(G(r_3)), H^9(G(r_2)), H^9(G(r_1)), H^9(G(r_0))$

Each commitment is required to be included in a disclosure, and then signed as part of the SD-JWT or MSO. The PID/(Q)EAA Provider is required to also include a number of decoy digests to hide the number of MDP partitions, or alternatively commit only an accumulator value (e.g. a Merkle Tree as proposed in the original HashWires paper or the digest over the concatenation of all the decoys and commitments). In Figure 8, and in the example below, the commitments are included as separate disclosures for illustrative purposes only.

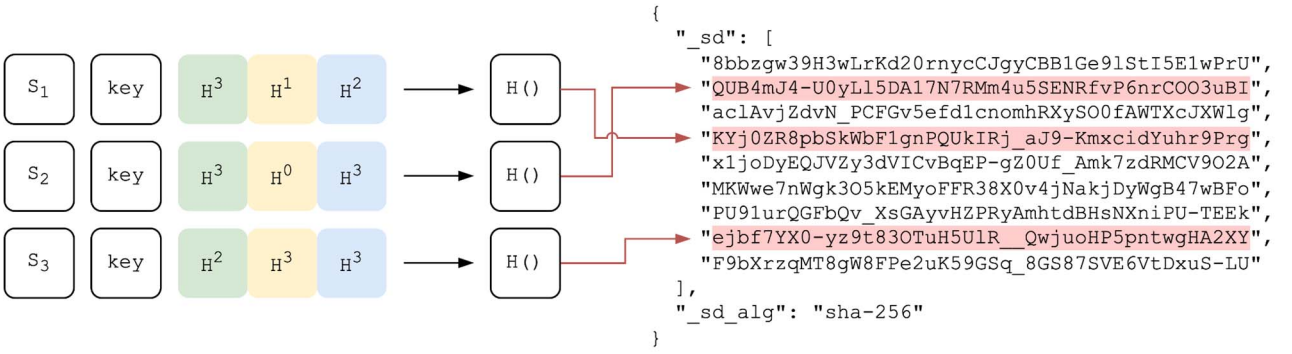

Figure 9: Optimized HashWires commitment using SD-JWT

Figure 9 illustrates an optimized HashWires commitment to the number 312 in base 4 protected by the `_sd` object suitable for an SD-JWT. Each commitment to the three partitions is salted (box with S), contain a MDP partition identifier key, and the hash chain roots for each MDP partition. The hash over the salt, key, and commitment is included in the `_sd` (red highlights). The other digests in the `_sd` object are decoys to hide the number of MDP partitions the user has. Each commitment is included as a disclosable value for illustrative purposes. Optionally, an issuer could instead add the commitments to an accumulator, which would be disclosable. This is an illustration of HashWires, although implementations may differ.

EXAMPLE: The random values needed to initiate each hash chain with $G(\cdot)$. The values are not sent to the verifier.

```
{
  "10^0": "f6a23b90b9f07f34f33dfd4e5de87adab167b6ea9eb060163e741ac26f16edc1",
  "10^1": "3026950fd2d2c6c7e23c8a8b0a80928d5cdac0f953699a96e02c1033379ed392",
  "10^2": "d942fdb1d9c3274a257154ef2f6f66161ea5872163dbb8daa40c7496e5365242",
  "10^3": "ba0acaf18a6a966a3eecbb791e9e22bc45d3a1183ff47342ab9cbde4635a828c",
  "10^4": "f32da5b457d45e0e6113d744fff316a1882f77fbf6ef5f92456faf84dfc8bd02"
}
```

The disclosure of the commitment to the partition 13699 using the format `["salt", "key", <value>]`.

```
["TpPrKdZ73ZR7JoUU-FCiTYvlQ4-QQ5ab9V2Z-cXze8E", "0",
["927eb07e71c648f73bec94e03d29cb41a0efc4f247a999d49f1318e3e8afbb84",
"b4b2a297499d63dd1ae5ee64c1aa21667b43b8974be3b3e17273005951413a56",
"854983f72c56c0102cac32edcce8b7c52365edc793cdba37d5603221b21d0a95",
"040be38408070da03bd6ca9e63999fac072adc20e1ba6f4513861db317a82a54",
"ad1a9492c27be7d33c7d00e33b0ca223e02a07440394b4036ded6f1f2c990c7a"]]
```

The base64url encoded SHA256 digest included in the `_sd`:

```
"zDHz3CX-akEjrDddMc8RYemeUCmEN0yjT1JIM_KXJd4"
```

NOTE 1: The user is required to only disclose the particular partition it uses to generate the inequality proof.

NOTE 2: The issuer can combine the disclosure digests into a single value using an accumulator or by concatenating the disclosure digests and the decoys. Implementation specific profiles are required.

The user, given a threshold value *t*, is required to select the partition that can generate the hash chains required for the inequality $x \geq t$. The user sends the disclosure of the commitment required for the inequality test, and the threshold values for each digit. The verifier can compute the hash chain using the threshold value for each digit and compares the root hash with the issuer signed commitments in the SD-JWT or MSO. If the signature is verified, the verifier accepts the inequality test.

4.4.7.5 Less than or equal to and range proofs

Any range proof, $a \leq x \leq b$, can be constructed using two inequality tests, one proving the inequality at the lower bound and the other at the upper bound. The above demonstrates an inequality test of type $a \leq x$. To generate a less than or equal to $x \leq b$ proof, it is necessary to extend the above described approach. Using whole number *K*, the issuer can generate a commitment to the inequality $K - x \geq K - b$. Both inequality tests rely solely on hash digests and combined they can generate any valid range proof using issuer signed commitments.

EXAMPLE:

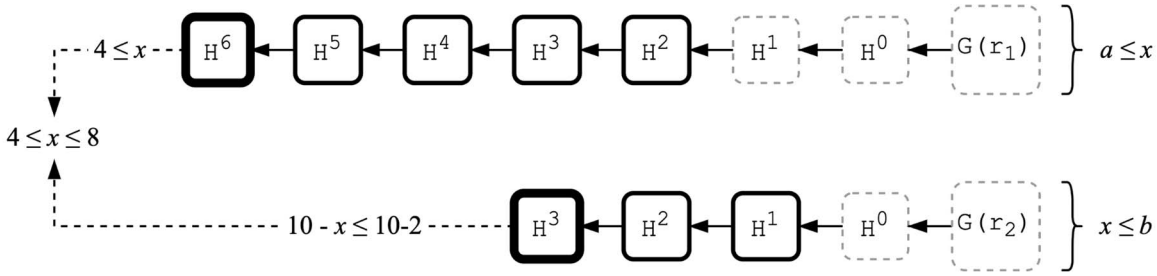

Figure 10: Hash chain based range proof

Figure 10 illustrates a hash chain based range proof for the range $4 \leq x \leq 8$. The issuer signs the bold commitments to both the lower bound test $4 \leq x$ and the upper bound test $x \leq 8$. The user presents both inequality tests to the verifier. The verifier combines the two proofs for inequality tests into range proof and accepts the range proof if the issuer's signature over the commitments is valid.

NOTE 1: For a range proof, the issuer is required to sign the parameter *K* used for the inequality $K - x \geq K - b$.

NOTE 2: The attestation issuance date impacts the proof that the user generates. A user generates a proof on an inequality test not for the request threshold, *t*, but subtracts the difference between the issuance date and the presentation date. A similar logic applies for age under or equal to proofs, as well as for range proofs.

HashWires represent an efficient way to generate inequality tests and range proofs using only SHA256. Running 70 000 loops on a dual core 2,2 GHz processor, it takes 72 µs ± 5,58 µs to generate the commitment for a 3 digit inequality test, and 156 µs ± 31,7 µs for a 6 digit one. The proof size is constant and the verification is faster than the generation.

4.4.7.6 Cryptographic analysis of HashWires

HashWires are considered as plausible quantum safe since they are based on hash chains. If the used hash functions are designed as QSC, the HashWires scheme becomes quantum-safe.

Since the HashWires scheme is based on chained salted attribute hashes, it can be designed to be unlinkable for verifier(s) collusion, but is not fully unlinkable (see clause 4.4.4).

4.4.8 Authentic Chained Data Containers (ACDC)

Authentic Chained Data Containers (ACDC) are verifiable data structures designed to cater for (Q)EAAs with selective disclosure requirements based on Directed Acyclic Graph (DAGs). While a detailed account of ACDC would require describing a suite of related specifications and standards (that cover key management topics, identifier systems, protocols for introduction and exchange, encoding, proofs, schemas, and the use of various event logs), the text herein focuses on the selective disclosure mechanism that are detailed in the IETF ACDC draft specification [i.111], more specifically in sections 2, 5 and 13.

Every salted attribute hash based approach relies on some form of logical ordering or structuring of the salted attributes that are included in an attestation. In ACDC, that structure is a Directed Acyclic Graph (DAG), where a knowledge graph expresses the attributes of the identity subject. A user may disclose various parts of such a graph, e.g. a vertex identifier, without disclosing any attribute values contained in the vertex, and/or the entire vertex.

The IETF ACDC draft specification offers multiple different, but closely related, disclosure mechanisms. To understand these mechanisms it is helpful to distinguish between mechanisms that offer contractual protection of the disclosure (i.e. mechanisms that detail permissions), and mechanisms that are primarily technical in nature (i.e. mechanisms that allow the recipient to obtain the plaintext attribute).

In ACDC, the contractual mechanisms can be expressed in legal terms as the value to a key, "l". This allows the user to specify certain terms and conditions associated with a potential disclosure of attributes, and the ACDC can present a set of such contractual terms under its rule attribute, "r". These mechanisms are not in place to enable disclosures of data for privacy purposes, but instead to protect the identity subject from the unauthorized exploitation of the disclosed data. While essential for a comprehensive grasp of ACDC's contributions, the intricate details of its contractual mechanisms are beyond the scope of the present report. Interested parties should refer to Sections 2 and 5 of the IETF ACDC draft specification [i.111] for a comprehensive examination. Of particular relevance herein is that these contractual agreements are designed to be both machine-readable and cryptographically verifiable, and that they play a role in interactions where disclosures are successive and depend on agreements that enable yet additional disclosures.

The IETF ACDC draft specification outlines several technical mechanisms to enable sharing only the minimum amount of information about the identity subject that the verifier needs. These mechanisms do not represent different selective disclosure techniques; rather they detail what of the DAG is revealed to a verifier. Three options are detailed:

1) The verifier obtains only a cryptographic digest of a set of key value pairs. These digests are referred to as "compact disclosures". These can be considered as a type of cryptographic commitment to a future disclosure.

2) The verifier obtains a set of key value pairs, and this disclosure contains correlatable information. This mechanism is referred to as "partial disclosure".

3) The verifier obtains a set of key value pairs, and this disclosure is not correlatable to any other yet undisclosed but disclosable key value pair. This mechanism is referred to as "selective disclosure".

Option 1 is used to enable Option 2 and 3. Option 2 is closely linked with successive disclosures where a user can disclose information over time following the acceptance of contractual agreements (e.g. first a commitment, then a schema, then a full disclosure of all attributes in a particular attestation). In contrast, Option 3 allows a user to disclose only a subset of key value pairs without any correlation handles such as an issuer signature over the entire salted attribute hash set. The ability of Option 3 to do so in turn relates to the DAG structure of ACDC and how an ACDC compliant attestation needs to be understood as a graph (section 4 in the IETF ACDC draft specification [i.111] provides additional details).

The content of an ACDC depends on its particular type, but for the purposes of explaining the selective disclosure mechanism employed the following example of a so called "private compact" variant is used with two properties important for understanding selective disclosure highlighted in green:


```
{
  "v":"ACDC10JSON00011c_",
  "d":"EBdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "u":"0ANghkDaG7OY1wjaDAE0qHcg",
  "i":"did:keri:EmkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
  "ri":"did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
  "s":"E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a":"EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
  "e":"ERH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZIl3MOA",
  "r":"Ee71iheqcywJcnjtJtQIYPvAu6DZIl3MORH3dCdoFOLB"
}
```


The example is private because it contains a property "u", which is a unique high entropy unique salt. This salt effectively blinds the digest commitment to the ACDC so that an entity cannot derive any of an ACDC's content knowing only its identifier (i.e. the value of "d", which is a content addressable and self referential identifier, called UUID, as specified in the IETF Self-Addressing IDentifier (SAID) draft specification [i.113]). Note that if an ACDC attribute set does not include an UUID, then its content is not private, and consequently it does not make much sense to discuss disclosure of attributes that an entity can derive using a rainbow table attack.

The example is compact because only commitments to other key value pair sets are included. For instance, in the above example, the key "a" is the unique identifier for a set of attributes but the attributes themselves are omitted.

A user can disclose the above ACDC by presenting ("u":"0ANghkDaG7OY1wjaDAE0qHcg"), i.e. a verifiable UUID, to a verifier and then disclosing the rest of the attributes in the ACDC. The verifier can then use the rest of the attributes to compute the value of "u" and compare it with the previously disclosed commitment. Relatedly, the user can further disclose identity related attributes by presenting the uncompacted private attribute key value set.

```
{
  "a":{
    "d":"EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
    "u":"0AwjaDAE0qHcgNghkDaG7OY1",
    "i":"did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA",
    "score":96,
    "name":"Jane Doe"
  }
}
```

Note how disclosure of attributes in "a" discloses the entire set. A user who wants to disclose individual identity attributes needs to use a selective disclosable attribute ACDC. There, each attribute is blinded individually as follows:

```
{
  "A":[
    {
      "d":"ErzwLIr9Bf7V_NHwY1lkFrn9y2PYgveY4-9XgOcLxUde",
      "u":"0AqHcgNghkDaG7OY1wjaDAE0",
      "i":"did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA"
    },
    {
      "d":"ELIr9Bf7V_NHwY1lkgveY4-Frn9y2PY9XgOcLxUderzw",
      "u":"0AG7OY1wjaDAE0qHcgNghkDa",
      "score":96
    },
    {
      "d":"E9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PYgveY4-",
      "u":"0AghkDaG7OY1wjaDAE0qHcgN",
      "name":"Jane Doe"
    }
  ]
}
```

Note how each attribute is selectively disclosable independently. Note also the capital "A" as key.

As with any salted attribute hash based approach to selective disclosure, ACDC only offers selective disclosure ability and does not offer inherent protection against verifiers colluding and correlating the users use of an ACDC. The UUID is a perfect correlation handle that any entity can use to track the user's behaviour. To protect against such correlation, the IETF ACDC draft specification discusses bulk issuance, where correlation handles are removed (see section 13.5.2 of IETF ACDC draft specification [i.111]). Note that such an approach does not protect against malicious issuers that wish to track the user. Succinctly put, ACDC is verifier unlinkable but not fully unlinkable.

ACDC is considered as being plausible quantum safe since they are based on hashes in a Directed Acyclic Graph. If the used hash functions are designed as QSC, the ACDC scheme becomes quantum-safe.

4.4.9 Gordian Envelopes

The Gordian Envelope [i.114] is a structured format for verifiable hierarchical data. The approach relies on a graph to logically order and structure salted attributes included in an attestation. Hence, it can be used to create Directed Acyclic Graphs (DAGs) through references within or between Envelopes. Claims can be structured as subject-predicate-object triplets (the predicate and the object are in turn envelopes), e.g. subject:Alice, predicate:knows, object:Bob.

The envelope itself is not limited to such triplets. An Envelope can enclose various types of data, ranging from basic plaintext messages to ciphertext to semantic graphs. These can then be represented in different ways in an envelope. The ways include nodes, leaves, nestled structures among others; common to all is that the envelope is meant to contain deterministically encoded identity subject claims that may or may not be encrypted, compressed, or made disclosable. The user has multiple ways to limit disclosures:

1) A single part of the triplet can be hidden: subject:Alice, predicate:knows, object:.
2) Multiple parts of the triplet can be hidden: subject:Alice, predicate:, object:.
3) The existence of the claim can be hidden.

Each envelope produces a unique and content determined digest, meaning that envelopes that are semantically identical produce the same digest. By extension, an identical identity subject with an identical claims set will yield the same digest tree every time the (Q)EAA is enveloped. As with other salted attribute hash approaches, the issuer signs the digests, which allows the user to later reveal claims associated with the digests. In the case of Gordian Envelopes, selective disclosure is possible by revealing only those objects required to traverse a path of interest and to calculate the Merkle root that is involved in the verification of the attestation.

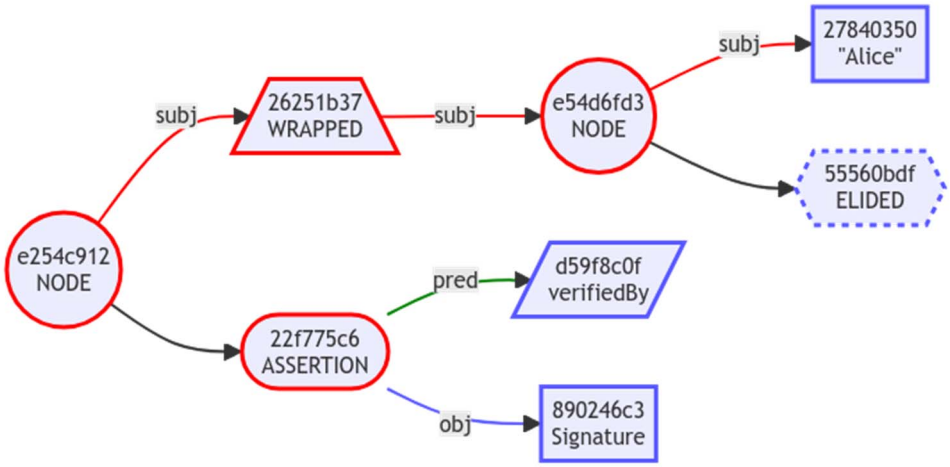

Figure 11: An example of a verifiable graph that selectively discloses only the subject

As with any salted attribute digest based approach to selective disclosure, a Gordian Envelope only offers selective disclosure ability and does not offer inherent protection against verifiers colluding and correlating the users based on the attestations they see. To prevent verifier collusion, Gordian envelopes support salting. Specifically, a unique salt is added as a predicate with a random number as the corresponding object to every envelope. As with any salted attribute hash approach, adding salts requires batch issuance, and does not protect against a malicious colluding issuer. In other words, Gordian Envelopes are verifier unlinkable but not fully unlinkable.

Gordian Envelopes are considered as being plausible quantum safe since they are based on hashes in a Directed Acyclic Graph. If the used hash functions are designed as QSC, the Gordian Envelopes scheme becomes quantum-safe.

4.5 Proofs for arithmetic circuits (programmable ZKPs)

4.5.1 General

Arithmetic circuits can represent any computational logic. Consequently, proofs for arithmetic circuits are "programmable": As every statement can be translated into an arithmetic circuit, a ZKP for any statement can be constructed. This offers a much higher degree of flexibility compared to "special-purpose ZKPs" such as BBS+ and CL-signatures, which rely on mathematical algorithms such as equality proofs for pre-images of commitments or proofs of knowledge of a private key as foundational components, from which more complex statements need to be constructed with much effort and no guaranteed success.

4.5.2 zk-SNARKs

4.5.2.1 Introduction to zk-SNARKs

The abbreviation zk-SNARK stands for "Zero-Knowledge Succinct Non-interactive ARgument of Knowledge", and is a collaborative term for a specific category of ZKP protocols. At the time of writing (in April 2024), eighteen zk-SNARK protocols have been published by cryptographic researchers; see clause A.4 for a list of all zk-SNARK protocols.

The zk-SNARK characteristics can be broken down as follows:

- zero-knowledge: As defined earlier, the proof gives no information beyond that the statement is correct, and any information that can be trivially derived from the statement (e.g. a ZKP that the statement that a holder is older than 19 is correct trivially proves also that the holder is older than 18).
- Succinct: the proof size grows sublinearly with the statement's size (e.g. logarithmically or even independent of statement size (constant proof size)).
- Non-interactive: randomness is not provided by the verifier (but by a random oracle). Consequently, a single message from the prover suffices to convince any verifier.

- ARgument: Cryptographic evidence (that relies on some battle-tested computational hardness assumptions such as DLP, as opposed to a full mathematical proof).
- of Knowledge: the proof demonstrates the user's knowledge of data (a witness) that proves the statement (not just its existence).

NOTE 1: A zk-SNARK system provides predicate proofs, selective disclosure and unlinkability by design.

The concept of zk-SNARK was initially described by Alessandro Chiesa et al in a paper [i.56] in 2012, which in turn was based on Jens Groth's work [i.99] from 2010. The first general or programmable zk-SNARK protocol Pinocchio [i.174] was designed and implemented in 2013. Hence, a zk-SNARK that is correctly executed (e.g. with a C program) can efficiently create specific ZKPs for any statement.

There is an important distinction between zk-SNARK proving systems that require a program (circuit)-specific preprocessing. So far, mainly preprocessing SNARKs have been used in practice (blockchain privacy and scaling projects) because they tend to have higher proving performance as they can be hand-optimized to the program. However, for different programs (e.g. patches) the preprocessing needs to be conducted again. On the other hand, so-called zero-knowledge virtual machines (zkVMs) can dynamically prove the correct execution of any program (represented by an instruction set received through compilation, e.g. a C or Rust program compiled with LLVM).

NOTE 2: In the zkVM case, there is also a preprocessing step, but it is only instruction set specific and, therefore, not program-specific.

A zk-SNARK protocol can be based on a trusted setup or as a transparent setup, as further described in clauses 4.5.2.2 and 4.5.2.3.

4.5.2.2 Trusted setup of zk-SNARKs

The trusted setup of a zk-SNARK involves three algorithms *KeyGen*, *CP*, *CV* as illustrated in Figure 12.

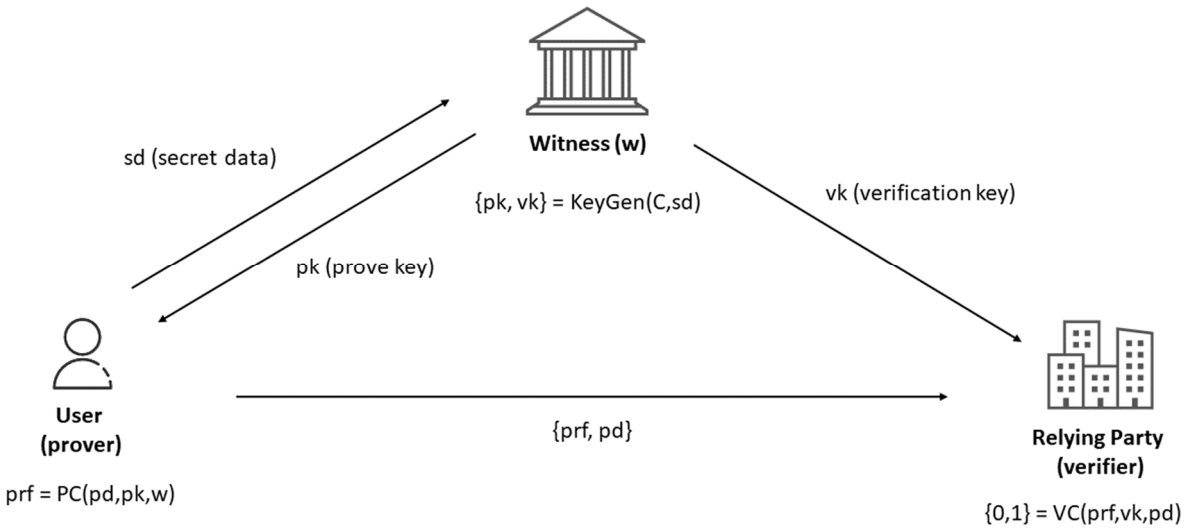

Figure 12: Overview of zk-SNARK with trusted setup

The key generator *KeyGen* takes a secret parameter *sd (secret data)*, also called "toxic waste", and the program *C* for which correct execution should be proven (the statement), and generates two publicly available keys, the user's proving key *pk*, and the relying party's verification key *vk*. These keys are public parameters that need to be generated once for a specific program *C*.

NOTE 1: The parameter *sd* used in the generator is a secret value. If this parameter is known to an attacker, it can generate fake proofs, i.e. without knowing a valid witness *w*. In other words, the soundness guarantees of the zk-SNARK would not be satisfied any more. However, the zero-knowledge property is not conditional on the secrecy of sd. In the context of digital attestations, even a citizen that does not trust the entity that ran the trusted setup need not to be afraid of a loss of privacy guarantees.

NOTE 2: To make sure that sd cannot be leaked, many projects (particularly on blockchains where whoever runs the trusted setup will unlikely be trusted by everyone), the trusted setup is usually operated in a multi-party computation by many entities, such that *sd* is only leaked if all of these entities collude. As such, if a verifier trusts only a single entity involved in the trusted setup, soundness of the zk-SNARK system is guaranteed, i.e. no fake proofs can be practically created.

NOTE 3: In principle, each relying party (verifier) could run their own trusted setup and distribute the corresponding pk to the holder: If the verifier protects their sd, they do not need to be afraid of receiving fake proofs. However, there are two significant drawbacks: pk tends to be large for practical presentations (tens to hundreds of MB), so real-time distribution is impractical and a pk that all verifiers accept is more desirable (particularly because different presentations correspond to different programs and, therefore, require different *pk*). Furthermore, as the holder cannot check the setup conducted by the verifier, additional certification of the pk to make sure it is derived from the correct program (and not some other program that outputs more information than stated), allowing a user to trust in the privacy guarantees.

The user executes the algorithm *CP* with the following input parameters: its (static) proving key *pk*, a (dynamic) public input *pd (public data)*, and a private witness *w*. The algorithm *CP* generates the proof value *prf = CP(pk, pd, w)*, as evidence that the user knows a witness *w*.

EXAMPLE 1: The public data *pd* could be the statement, for example that the user's age is above 18. It will also likely involve a nonce to avoid replay attacks and a set of public keys for accepted issuers against which the signature of the user's attestation (which represents part of the witness) is verified in the zk-SNARK.

The verifying relying party calculates the algorithm *CV(vk, pd, prf)* which returns true if the proof is correct and false otherwise. Hence, the function *CV* returns true if the user knows a witness *w* that satisfies the function *C(sd,w) = true*.

EXAMPLE 2: zk-SNARK protocols with trusted setup are Pinocchio [i.174], Geppetto [i.60], and TinyRAM [i.17]. For a complete list of zk-SNARK protocols with trusted setups, see table A.4 in clause A.4.

NOTE 4: Most zk-SNARKs with trusted setup actually involve a two-step trusted setup: one that is not dependent on C and a second one that is dependent on C. In 2019, PLONK [i.93] was introduced as a universal zk-SNARK protocol. In this approach, only the first step which is independent of C involves toxic waste that may compromise soundness; and the second, C-dependent step - while involving a computationally intensive preprocessing step - does not involve toxic waste anymore but only relies on the output of the first step. However, the "complexity" of the programs C that can be covered is bounded by the sizes covered by the first step.

Universal trusted setup: In 2019, PLONK [i.93] was introduced as the universal zk-SNARK protocol.

4.5.2.3 Transparent setup zk-SNARKs

In a transparent (public) setup of zk-SNARK there is no need for a trusted setup with a witness. As a tradeoff, the proving performance of a transparent zk-SNARK protocol may be lower than a zk-SNARK with trusted setup, and the proof size of a transparent zk-SNARK protocol may be larger than a zk-SNARK with trusted setup.

EXAMPLE: zk-SNARK protocols with transparent (public) setups are SuperSonic [i.157], Hyrax [i.197] and Halo [i.29]. For a complete list of zk-SNARK protocols with transparent setups, see table A.4 in clause A.4.

4.5.2.4 Cryptography behind zk-SNARKs

The cryptography that underpin the zk-SNARK schemes is highly complex and differs from protocol to protocol.

In brief, the zk-SNARK protocols can be constructed based on the following cryptographic building blocks [i.175]:

- Fiat-Shamir Heuristics, which in turn can be broken down into Sigma-Protocols, Random Oracle Models (ROM) and Fiat-Shamir-Compatible Hash Functions.
- Probabilistically Checkable Proofs (PCP): Merkle Trees and Hash Functions, Kilian Interactive Argument of Knowledge, and Micali's Computationally Sound (CS) Proof.

- Quadratic Arithmetic Programs (QAPs) and Square Span Programs (SSPs).
- Linear Interactive Proofs (LIPs).
- Polynomial Interactive Oracle Proofs (PIOPs).

A common construction involves three steps:

1) Arithmetization: Representing the program *C* as a sequence of simple algebraic operations, such as additions and multiplications. Common representations are Rank-1 Constraint Systems (R1CS), PLONKish, and Algebraic Intermediate Representation (AIR).

2) This representation is translated into one or multiple polynomials, such that knowledge of a witness, corresponding to a valid execution trace of *C*, corresponds to certain properties of the polynomials (e.g. roots at certain positions or equalities between one polynomial and a product of two other polynomials). Challenging this equality under the assumption of a truthfully answering prover corresponds to an Interactive Oracle Proof (IOP). The IOP is an information-theoretic object, i.e. it does not rely on cryptographic hardness assumptions. Because of the good error-amplification of polynomial encodings following the Schwartz-Zippel lemma (polynomials of low degree in a large field will either be equal or different in almost every point), few spot checks are sufficient, with the corresponding points for the spot checks determined using the Fiat-Shamir heuristic.

3) Using a cryptographic Polynomial Commitment Scheme (PCS), the prover can be forced to answer truthfully to queries of these polynomials (which are not shared by the prover). The PCS is responsible for the transparency properties of the setup (trusted or transparent) and the reason why a "proof" based on a PCS becomes an "argument".

NOTE 1: Depending on the IOP and PCS, some zk-SNARKs are not post-quantum secure, i.e. soundness guarantees rely on hardness assumptions such as DLP. As for the toxic waste, the zero-knowledge property is, by contrast, unconditional.

NOTE 2: Bulletproofs [i.36] - developed by Bünz et al. - are a family of zk-SNARKs with reduced succinctness properties (proof size is sublinear, but verification time is not).

NOTE 3: zk-STARKs [i.15] and [i.162] - developed by Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev [i.16] - are a family of transparent zk-SNARKs that are plausibly post-quantum secure, i.e. soundness guarantees hold against an adversary with a quantum computer. They are instantiated with a specific arithmetization (AIR) and IOP-PCS combination (Fast Reed Solomon IOP - FRI) that relies on low-degree testing of polynomials and Merkle trees for opening polynomials on small subgroups. Because of their FRI-based construction, proof sizes of zk-STARKs are around 100 to 1 000 times higher than proof sizes of the shortest zk-SNARKs.

Given the vast literature of zk-SNARK algorithms, a complete description of the cryptography for zk-SNARKs goes beyond the scope of the present document. For further reading about the cryptographic algorithms behind the zk-SNARK protocols, the following papers are recommended: Nitulescu "zk-SNARKs: A Gentle Introduction" [i.164], Petkus "Why and How zk-SNARK Works: Definitive Explanation" [i.175], and Evans "Succinct Proofs and Linear Algebra" [i.90].

4.5.2.5 Implementations

As regards to implementations, zk-SNARK was implemented in 2016 for the blockchain protocol ZeroCash for cryptocurrency ZCash, for which zk-SNARK caters for four different transaction types: private, shielding, deshielding, and public. Hence, zk-SNARK allows the users to determine how much data to be shared with the public ledger for each transaction. The blockchain Ethereum zk-Rollups also utilizes zk-SNARKs to increase its scalability. In doing so, they do not make use of the zero-knowledge property but the succinctness property, so some zk-rollups, in fact, are based on SNARKs and not on zk-SNARKs. Furthermore, zk-SNARKs have been implemented as general-purpose ZKP schemes in combination with existing digital identities, as described in clause 6.5.

4.5.2.6 Cryptographic analysis

Whether a zk-SNARK protocol is quantum-safe or not depends on the underlying cryptographic algorithms, as described in table A.4. The zk-SNARK protocols Aurora [i.18], Ligero [i.5], Spartan [i.159], and Virgo [i.214] are considered as plausible quantum-safe (related to soundness), whilst the others in table A.4 are not considered as quantum-safe.

It is possible to implement presentations of (Q)EAA using zk-SNARKs that support fully unlinkable attestations.

NOTE 1: Succinct proofs can typically be turned into ZKPs quite easily through adding blinding factors, since a succinct proof already eliminates a lot of superfluous information ("there cannot be much sensitive information left"). In the context of the EUDIW, the succinctness property is arguably not very relevant because the complexity of the statement to be proved is low enough to be handled directly by a mobile phone. Hence, it makes a lot of sense to look into programmable ZKPs beyond zk-SNARKs. Yet, because of their limited computational power, the focus of the blockchain project has lied on succinct proofs, such that progress and industry-grade tooling is arguably most advanced there.

NOTE 2: It is possible to combine ZKPs based on CL-signatures or BBS(+) with proofs for arithmetic circuits. For instance, BBS can be used for a proof of knowledge of the issuer's signature and reveal commitments to selected attributes. Then, a programmable ZKP (e.g. a zk-SNARK) can be used to prove certain properties of the identity attribute (the pre-image of the revealed hash), e.g. to compute a complex predicate. A well-known construction that follows this paradigm is LegoSNARK [i.45], implemented in the context of digital attestations, among others, by dock.io.

5 (Q)EAA formats with selective disclosure

5.1 General

The present clause provides an analysis of a set of formats for selective disclosure.

The topics for the analysis of each selective disclosure (Q)EAA formats are:

- Signature scheme(s) used for selective disclosure and optionally unlinkability, when applicable with references to clause 4.
- Encoding of the (Q)EAAs used for selective disclosure.
- Maturity of the (Q)EAA format's specification and deployment.
- Cryptographic aspects, more specifically if the cryptographic algorithms used for the selective disclosure (Q)EAA formats are approved by SOG-IS and allows for QSC algorithms for future use.

The (Q)EAA formats are categorized according to three of the main cryptographic schemes for selective disclosure:

- Atomic (Q)EAA formats, see clause 5.2. These (Q)EAA formats correspond to the (Q)EAA signature schemes described in clause 4.2.
- Multi-message signature (Q)EAA formats, see clause 5.3. These (Q)EAA formats correspond to the multi-message signature schemes described in clause 4.3.
- (Q)EAAs with hashes of salted attributes, see clause 5.4. These (Q)EAA formats correspond to the multi-message signature schemes described in clause 4.4.

NOTE 1: There is also a type of generic JSON container format (JSON WebProofs), which allows for a mix of the selective disclosure signature schemes in clause 4, and is therefore treated as a separate category of (Q)EAA formats.

NOTE 2: The proofs for arithmetic circuits (such as zk-SNARKs) do not rely upon (Q)EAA formats per se, as they can prove the correct execution of any credential verification program in zero-knowledge. Hence, proofs for arithmetic circuits are out of scope for this clause, which describes (Q)EAA formats. However, clause 6.5 describes solutions that are implemented based on a combination of programmable ZKPs (such as zk-SNARKs) with existing credentials (such as X.509 certificates).

5.2 Atomic (Q)EAA formats

5.2.1 Introduction to atomic (Q)EAA formats

The concept of atomic (Q)EAAs was introduced in clause 4.2. There are numerous (Q)EAA formats that can be issued with a single claim, so in principle a selective disclosure scheme based on atomic claims can be designed for a variety of types of (Q)EAA formats (ICAO DTCs, IETF JWTs, W3C Verifiable Credentials, X.509 certificates, etc.).

Clauses 5.2.2 and 5.2.3 are however focusing in more detail on two (Q)EAA formats that are used for atomic (Q)EAA schemes: PKIX X.509 attribute certificates and W3C Verifiable Credentials.

5.2.2 PKIX X.509 attribute certificate with atomic attribute

The PKIX X.509 Attribute Certificate (AC) profile is specified in IETF RFC 5755 [i.125]. An attribute certificate may contain attributes that specify group membership, role, security clearance, or other authorization attributes associated with the user. The attribute certificate is a signed set of attributes, although it does not contain a public key. Instead, the attribute certificate is linked to a X.509 Public Key Certificate (PKC), which can be used by the user for authentication. In order to preserve the user's privacy, the X.509 public key certificate may only include a pseudonym in the subject field.

The attribute certificates are issued by an Attribute Authority (AA), and they may be issued with a short lifetime and with an atomic (single) attribute. These characteristics make short-lived attribute certificates with atomic credentials suitable for an access control service with selective disclosure features.

A description of how to use PKIX X.509 attribute certificates for selective disclosure with an access control system is available in clause 6.2.1.

The X.509 attribute certificates are ASN.1/DER encoded as described in IETF RFC 5755 [i.125].

X.509 certificates can be signed by the QTSP using cryptographic algorithms (RSA with proper key lengths or ECC with approved curves) that are published by SOG-IS [i.188]. For future use, the X.509 certificates can be signed with quantum-safe cryptographic algorithms [i.152].

The maturity of X.509 attribute certificates can be considered as high, given that the IETF RFC 5755 [i.125] is a mature PKIX standard.

5.2.3 W3C Verifiable Credential with atomic attribute

As a preparation for enrolment of W3C Verifiable Credentials with atomic attributes, the EUDI Wallet would need to be equipped with Credential templates for the W3C Verifiable Credentials. The W3C Verifiable Credentials Data Model v1.1 [i.209] distinguishes between a Credential as "*a set of one or more claims made by an issuer*" and a Verifiable Credential as "*a verifiable credential is a tamper-evident credential that has authorship that can be cryptographically verified*". Put differently, a Verifiable Credential can be a signed Credential. Hence, the Credential(s) in the EUDI Wallet can consist of templates with the attribute properties that should be used for the enrolment of attribute values.

NOTE: The W3C Verifiable Credentials Data Model v1.1 [i.209] is a conceptual data model rather than a specific credential format. In this context of atomic attributes, however, the scope of W3C Verifiable Credentials can be limited to the JWT format.

A description of how to use the FIDO standard as an authentication protocol in conjunction with Verifiable Credentials with atomic attributes for selective disclosure is available in clause 6.2.2.

The encoding of the W3C Verifiable Credentials is specified as JWT or JSON-LD in the W3C Verifiable Credentials Data Model v1.1 [i.209].

W3C Verifiable Credentials can be signed by the QTSP using cryptographic algorithms (RSA with proper key lengths or ECC with approved curves) that are published by SOG-IS [i.188]. For future use, the W3C Verifiable Credentials can be signed with quantum-safe cryptographic algorithms as described in the IETF report on JOSE signatures with QSC algorithms [i.119].

The maturity of W3C Verifiable Credentials can be considered as high, given the wide deployment of issued W3C Verifiable Credentials.

5.3 Multi-message signature (Q)EAA formats

5.3.1 W3C VC Data Model with ZKP

The W3C Verifiable Credentials (VC) Data Model v1.1 [i.209] contains clause "5.8 Zero-Knowledge Proofs", which describes a data model that supports selective disclosure with the use of Zero-Knowledge Proof (ZKP) mechanisms.

The W3C Verifiable Credentials Data Model states two requirements for Verifiable Credentials when they are to be used in ZKP systems:

- The Verifiable Credential contains a proof, so that the user can derive a verifiable presentation that reveals only the information that the holder intends to reveal.
- The credential definition (if being used) is defined in the JSON credentialSchema property, so that it can be used to perform various cryptographic operations in zero-knowledge.

The following cryptographic schemes that support selective disclosure while protecting privacy across multiple presentations have been implemented for the W3C Verifiable Credentials Data Model [i.209]: IETF CFRG BBS [i.116], CL Signatures [i.40], Idemix [i.109], Merkle Disclosure Proof 2021 [i.204], Mercurial Signatures [i.43], PS Signatures [i.176], U-Prove [i.2] and Spartan [i.186].

More specifically, the W3C Verifiable Credentials Data Model standard includes examples of how to use Camenisch-Lysyanskaya (CL) signatures (see clause 4.3.2) with a W3C Verifiable Credential and a W3C Verifiable Presentation; see examples 24 and 25 in W3C Verifiable Credentials Data Model [i.209] for examples of these data structures.

An example of how to combine two W3C Verifiable Credentials into a W3C Verifiable Presentation with selected attributes is shown in Figure 13.

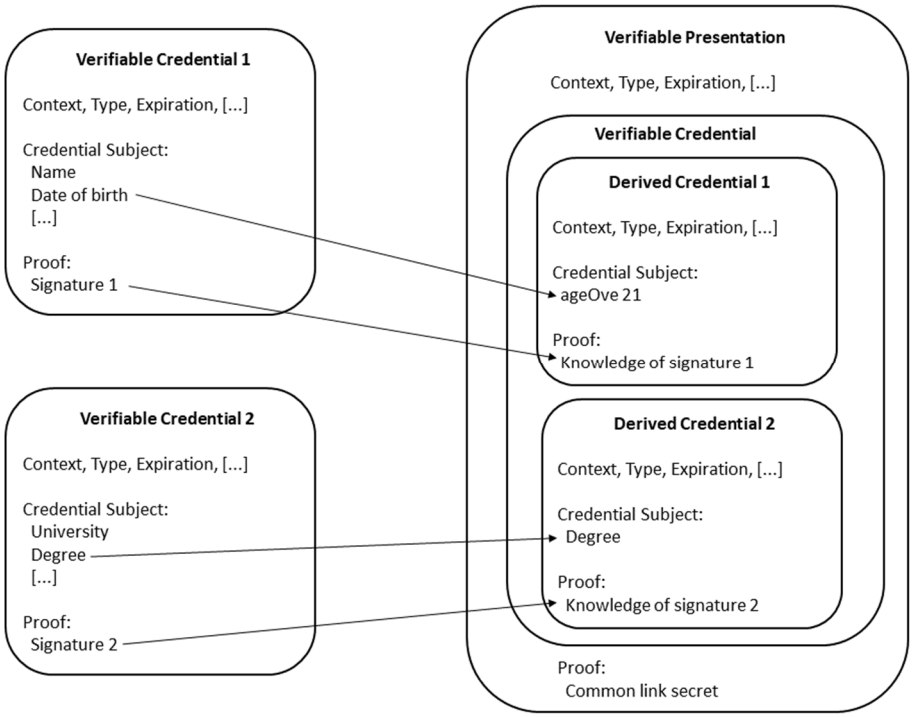

Figure 13: W3C Verifiable Credentials presented using ZKP

In Figure 13, selectively disclosed attributes from W3C Verifiable Credential 1 and W3C Verifiable Credential 2 are combined into a W3C Verifiable Presentation. CL-signatures are used in the Verifiable Presentation to create the proofs of knowledge of the original W3C Verifiable Credential signatures.

5.3.2 W3C VC Data Integrity with BBS Cryptosuite

5.3.2.1 W3C BBS Cryptosuite v2023

W3C BBS Cryptosuite v2023 [i.212] is an experimental draft specification, which defines a set of cryptographic suites for the purpose of creating, verifying and deriving proofs for the IETF CFRG BBS [i.116] draft signature scheme that specifies BBS+ (see clause 4.3.1.4). The BBS+ signatures are compatible with any pairing friendly elliptic curve, however the cryptographic suites defined in the W3C BBS Cryptosuite specification allow the usage of the BLS12-381 curve for interoperability purposes.

NOTE: The W3C draft specification has the title "W3C BBS Cryptosuite v2023", although it describes the BBS+ scheme. The term BBS+ is however used throughout the present document to describe the multi-message signature scheme, whilst the term BBS04 describes the original single-message signature scheme.

W3C BBS Cryptosuite v2023 [i.212] can be used in conformance with the W3C Verifiable Credentials Data Integrity v1.0 specification [i.208], which in turn describes mechanisms for ensuring the authenticity and integrity of JSON-LD encoded credentials according to W3C Verifiable Credentials Data Model v2.0, especially through the use of digital signatures and related cryptographic proofs.

As a result, the IETF CFRG BBS signature scheme (clause 4.3.1.4) can be applied on W3C Verifiable Credentials v2.0 and W3C Verifiable Presentations in order to disclose selected attributes, which are signed by the user's proofs without revealing the entire W3C Verifiable Credentials and their original signatures.

5.3.2.2 W3C VC Data Integrity with ISO standardized BBS04/BBS+

In this clause it is analysed whether the ISO/IEC standardization efforts of BBS04/BBS+ (see ISO/IEC 20008-2 [i.143], ISO/IEC PWI 24843 [i.144] and ISO/IEC CD 27565 [i.150], clause 4.3.5) are compatible with W3C BBS Cryptosuite v2023 and W3C Verifiable Credentials Data Integrity v1.1.

At the time of writing (April 2024), ISO/IEC 20008-2 [i.143] mechanism 3 is thus far the only ISO standard that specifies the qSDH cryptographic primitives for BBS04. However, ISO 20008-2 mechanism 3 is designed for single messages and is therefore neither compatible with W3C BBS Cryptosuite v2023 nor W3C Verifiable Credentials Data Integrity v1.1. It has been proven [i.13] that BBS+ with multi-messages has the same security features as BBS04 with single messages, although BBS+ is not yet standardized by ISO.

If the ISO/IEC PWI 24843 [i.144] is approved to standardize privacy-preserving attribute-based credentials schemes, the potentially new ISO standard may include a standardized version of BBS+ that has the potential to be compatible with W3C BBS Cryptosuite v2023 and W3C Verifiable Credentials Data Integrity v1.1.

Furthermore, ISO/IEC CD 27565 [i.150] refers to IETF CFRG BBS (clause 4.3.1.4), whilst W3C BBS Cryptosuite v2023 also refers to IETF CFRG BBS, so both ISO/IEC 27565 [i.150] and W3C BBS Cryptosuite v2023 share IETF CFRG BBS as a common reference for the BBS+ scheme.

Hence, if ISO/IEC PWI 24843 and/or ISO/IEC CD 27565 will standardize BBS+ according to IETF CFRG BBS in conjunction with DIF draft "Blind Signatures extension of the BBS Signature Scheme" [i.66], then W3C BBS Cryptosuite v2023 can be enhanced to reference such an ISO standard. In such a scenario, the W3C Verifiable Credential Data Integrity 1.0 specification will refer to an ISO compliant version of W3C BBS Cryptosuite v2023. Finally, the W3C Verifiable Credentials Data Model v2.0 can be deployed with W3C Verifiable Credential Data Integrity 1.0, which is underpinned with an ISO standardized version BBS+.

NOTE 1: W3C Verifiable Credentials Data Model v2.0 with JSON-LD encoding has the potential to be underpinned by an ISO standardized version BBS+.

NOTE 2: W3C Verifiable Credentials Data Model v1.1 with JWT encoding does not refer to W3C Verifiable Credential Data Integrity 1.0, and can therefore not be supported by an ISO standardized version of BBS+.

5.3.3 W3C Data Integrity ECDSA Cryptosuites v1.0

The W3C "Data Integrity ECDSA Cryptosuites v1.0" [i.201] specification describes a data integrity cryptosuite for use when generating a digital signature using the Elliptic Curve Digital Signature Algorithm (ECDSA). The data integrity cryptosuites are in conformance with the W3C Verifiable Credentials Data Integrity [i.208] specification.

More specifically, selective disclosure is described in generalized terms according to the ECDSA-SD-2023 functions. The function createDisclosureData is used to generate a derived proof. The inputs include a JSON-LD document, an ECDSA-SD base proof, an array of JSON pointers to use to selectively disclose statements, and any custom JSON-LD API options (such as a document loader). The disclosure data object is produced as output, which contains the selectively disclosed fields of the document along with the ECDSA-SD proof.

5.3.4 Hyperledger AnonCreds (format)

The Hyperledger AnonCreds [i.104] credentials are JSON-formatted according to public AnonCreds objects, which in turn are defined by Schemas, CredDefs, Revocation Registry Definitions and Rev_Reg_Entrys. These objects are published by the issuers to repositories called Verifiable Data Registries (VDRs), which are accessible to users and verifiers to enable presentation generation and verification. AnonCreds can also be issued in accordance with the W3C Verifiable Credentials Data Model.

AnonCreds are bound to the user with a non-correlatable secret only known to the user itself called a link secret. The link secret as a blind attribute that is sent to the issuer during credential issuance. The issuer signs every claim (including the blinded link secret) individually, enabling selective disclosure. The Pedersen Commitment is used for the link secret. It means the issuer does not know the exact value of the link secret, and the holder can prove the ownership of credentials to a verifier without disclosing a persistent identifier. A user can link two attestations by generating a proof that the two exponents in the Pedersen Commitments are equal, i.e. they contain the same link secret.

The cryptographic signature scheme used by AnonCreds is CLRSA-signatures (see clause 4.3.2), which caters for selective disclosure and full unlinkability.

More information about the AnonCreds protocols is available in clause 6.3.1.

5.3.5 Cryptographic analysis

The maturity of W3C Verifiable Credentials can be considered as high, given the wide deployment of issued W3C Verifiable Credentials. However, BBS+, CL signatures and ECDSA are not secure against quantum-safe cryptographic algorithms [i.193] (see also clause 9), and they are additionally not standardized by NIST in the US or by SOG-IS in the EU. Furthermore, since AnonCreds are based on CLRSA-signatures, the cryptographic algorithms are not considered as quantum-safe nor SOG-IS approved.

5.4 (Q)EAAs with salted attribute hashes

5.4.1 General

The general concept of selective disclosure based on salted attribute hashes is described in clause 4.4. As regards to credentials within this category, there are several noteworthy formats. The formats that are described more in-depth in the present report are:

- IETF SD-JWT, which is further described in clause 5.4.2.
- ISO mDL MSO (Mobile Security Object), which is elaborated in clause 5.4.3.

NOTE: ETSI EN 319 162-1 [i.73] specifies the Associated Signature Containers (ASiC), which is an XML-formatted manifest that binds together a number of hashed file objects into one single digital container. The principle of combining hashed objects in an ASiC manifest is similar to the IETF SD-JWT and ISO mDL MSO credentials with salted attribute hashes. There are however two main differences:

ETSI ASiC is intended for combining file objects in a signature container manifest, whilst IETF SD-JWT and ISO mDL MSO are designed for selective disclosure.

Furthermore, the ETSI ASiC hashes are not salted, whilst the hashed attributes in IETF SD-JWT and ISO mDL MSO are salted to cater for unlinkability. Hence, the comparison with ETSI ASiC is observed, but nevertheless out of scope for this clause.

In addition to the above two formats, the present document also includes a mention of disclosure mechanisms based on proof mechanisms detailed in JSON Web Proofs and describes a proposal that relies on Directed Acyclic Graphs (DAG).

5.4.2 IETF SD-JWT

To support selective disclosure in JWTs, IETF has specified Selective Disclosure JSON Web Token (SD-JWT) [i.123]. At its core, an SD-JWT is a digitally signed JSON document that can contain salted attribute hashes that the user can selectively disclose using disclosures that are outside the SD-JWT document. This allows the user to share only those attributes that are strictly necessary for a particular service. The technique of SD-JWT is based on salted attribute hashes as described in clause 4.4.

Each SD-JWT contains a header, payload, and signature. The header contains metadata about the token including the type and the signing algorithm used. The signature is generated using the issuer's private key. The payload includes the proof object that enables the selective disclosure of attributes. Each disclosure contains a salt, a cleartext claim name, and a cleartext claim value. The issuer then computes the hash digest of each disclosure and includes each digest in the attestation it signs and issues.

NOTE: The JOSE [i.135] signature format allows for SOG-IS approved cryptographic algorithms [i.188] and QSC algorithms [i.119] for future use.

The SD-JWT specification is still a draft, yet SD-JWT has been selected in the ARF [i.59] as the JSON-format for selective disclosure.

A thorough analysis of SD-JWT and how it can be applied for selective disclosure of the PID/(Q)EAA for the EUDI Wallet is available in clause 7.3.

5.4.3 ISO/IEC 18013-5 Mobile Security Object (MSO)

The Mobile Security Object (MSO) is specified in clause 9.1.2.4 of ISO/IEC 18013-5 [i.140] and contains the following attributes encoded in a CDDL [i.136] structure:

- digestAlgorithm: Message digest algorithm
- valueDigests: Array of digests of all data elements
- deviceKey: Device key in COSE_Key as defined in IETF RFC 8152 [i.133]
- docType: DocType as used in Documents
- validityInfo: validity of the MSO and its signature

The valueDigests are issued as IssuerSignedItems, which are the hash values of the ISO mDL attributes combined with random values (see ISO/IEC 18013-5 [i.140], clause 9.1.2.4). In other words, the MSO is a selective disclosure standard based on salted hashes of attributes (see clause 4.4), where the random values are the salts.

The deviceKey contains the mDL Authentication Key (see clause 7.2.2), which is protected by the user's PIN-code or biometrics (see clause 7.6).

The MSO is signed by the mDL Issuer Authority, which is an IACA X.509 CA (see clause 7.2.1.4), and the signature is COSE formatted.

NOTE 1: ISO/IEC 18013-5 [i.140], Table B.3 "Document signer certificate" lists the ECDSA curves BrainpoolP256r1, BrainpoolP384r1 and BrainpoolP512r1, which are also approved by SOG-IS [i.188].

NOTE 2: The COSE [i.129] signature format also allows for QSC algorithms [i.119] for future use.

An example of an ISO mDL MSO data structure is provided in ISO/IEC 18013-5 [i.140], annex D.5.2.

The MSO is stored and protected in the device's SE/TEE. The MSO is included in the mDL Response for the device retrieval flow (see clause 7.2.3).

ISO/IEC 18013-5 [i.140] is considered mature, and several ISO mDL device retrieval solutions with MSOs have been deployed in production, for example in a number of states in the US.

A thorough analysis of ISO mDL MSO and how it can be applied for selective disclosure of the PID/(Q)EAA for the EUDI Wallet is available in clause 7.2.

5.5 JSON container formats

5.5.1 IETF JSON WebProof (JWP)

The JOSE [i.120] standard is a widely adopted container format for JSON-formatted Keys (JWK), Signatures (JWS), and Encryption (JWE). For example, JWTs with JOSE-containers are used by the OpenID Connect standard and by W3C's Verifiable Credentials.

However, JOSE is not designed to cater for the growing number of selective disclosure and ZKP schemes. Most of these emerging cryptographic schemes require additional transforms, are designed to operate on subsets of messages, and have more input parameters than traditional signature algorithms.

Examples of selective disclosure signature schemes that would benefit from a more flexible JSON container format are:

- BBS+ [i.116];
- CL Signatures [i.40];
- Idemix [i.109];
- Merkle Disclosure Proof 2021 [i.204];
- Mercurial Signatures [i.43];
- PS Signatures [i.176];
- U-Prove [i.2]; and
- Spartan [i.186].

They adhere to the same principles of collecting multiple attributes and binding them together into a single issued token, which is transformed into a presentation that reveals only a subset of the original attributes, predicate proofs, or proofs of knowledge of the attribute.

In order to address these issues, the IETF JSON working group has drafted the JSON WebProof (JWP) specification. The JWP specification defines a new JSON container format similar in design to JSON Web Signature (JWS). However, JWS only integrity-protects a single payload, whilst JWP can integrity-protect multiple payloads in one message. JWP also specifies a new presentation form that supports selective disclosure of individual payloads, enables additional proof computation, and adds a protected header to prevent replay and support binding mechanisms.

5.5.2 W3C JSON Web Proofs For Binary Merkle Trees

In hash-based cryptography, the Merkle signature scheme is a digital signature scheme based on Merkle trees and one-time signatures such as the Lamport signature scheme. It was developed by Ralph Merkle in the late 1970s and is an alternative to traditional digital signatures such as DSA or RSA. An advantage of the Merkle signature scheme is that it is plausible quantum-safe.

The JSON Web Proofs For Binary Merkle Trees [i.203] specification defines a generic encoding of merkle audit paths that is suitable for combining with JWS to construct selective disclosure proofs. The specification is suitable for more generic applications and formats such as W3C Verifiable Credentials [i.209] and W3C Decentralized Identifiers [i.202].

JSON Web Proofs (see clause 5.5.1) are used as formats for the encoding binary merkle trees.

Selective disclosure is defined as the same as full disclosure with the exception that the rootNonce is not encoded in the compressed representation. The rootNonce is omitted in order to ensure that a selective disclosure proof does not reveal information that can be used to brute force siblings of disclosed members.

Merkle proofs are already being used to provide certificate transparency in IETF RFC 9162 [i.137]. The JSON Web Proofs For Binary Merkle Trees [i.203] specification is however independent of the certificate transparency specification.

6 Selective disclosure systems and protocols

6.1 General

The present clause provides an analysis of a set of systems and protocols for selective disclosure.

The topics for the analysis of each selective disclosure protocol are:

- Signature scheme(s) used for selective disclosure and optionally Zero-Knowledge Proofs, when applicable with references to clause 4.
- (Q)EAA format(s) for selective disclosure, when applicable with references to clause 5.
- Protocol(s) for presentation of the user's (Q)EAAs to a relying party (relying party).
- Maturity of the protocol's specification and deployment.
- Cryptographic aspects, more specifically if the cryptographic algorithms used for the selective disclosure protocol are approved by SOG-IS and allows for QSC algorithms for future use.

The protocols are first categorized according to the four main cryptographic schemes for selective disclosure:

- Atomic (Q)EAA protocols, see clause 6.2. These protocols correspond to the (Q)EAA signature schemes described in clause 4.2 and formats in clause 5.2.
- Multi-message signature protocols, see clause 6.3. These protocols correspond to the multi-message signature schemes described in clause 4.3 and formats in clause 5.3.
- Salted attribute hashes protocols, see clause 6.4. These protocols correspond to the multi-message signature schemes described in clause 4.4 and formats in clause 5.3.
- Proofs for arithmetic circuits protocols, see clause 6.5. These protocols correspond to the proofs for arithmetic circuits described in clause 4.5.

In addition to the traditional categories listed above, the following systems are described, which are based on a mix of selective disclosure schemes:

- Anonymous attribute based credentials systems, see clause 6.6.
- ISO mobile driving license (ISO mDL), see clause 6.7.

6.2 Atomic attribute (Q)EAA presentation protocols

6.2.1 PKIX X.509 attribute certificates with single attributes

An access control system based on PKIX X.509 certificates with atomic attributes is illustrated in Figure 14.

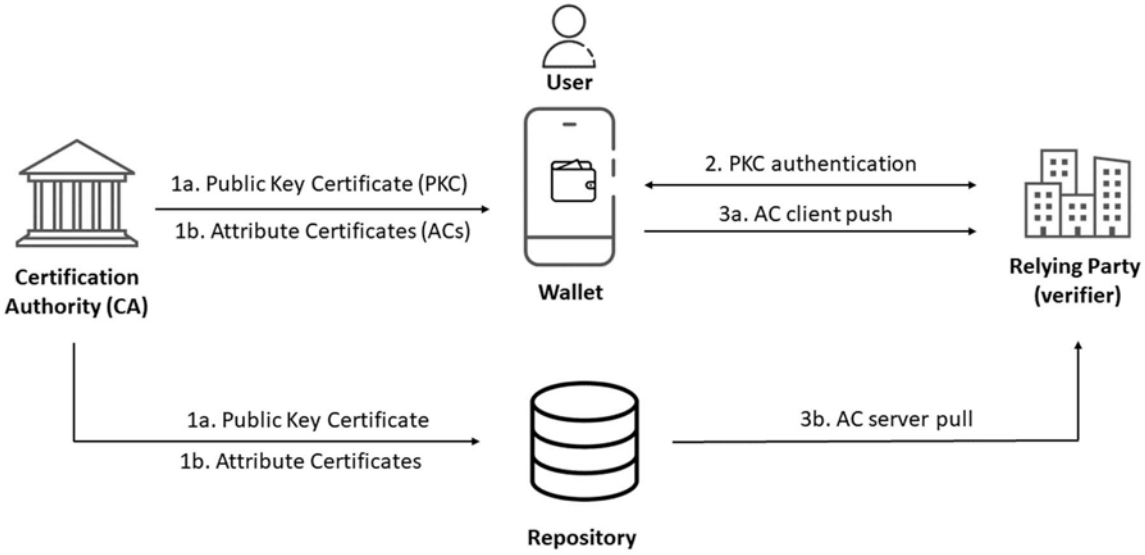

Figure 14: Overview of attribute certificate authorization

First, the system is configured by a Certification Authority (CA) that issues a PKIX X.509 public key certificate to a user's wallet. The user has a corresponding private key protected in the wallet, such that the user can be authenticated with the public key certificate. The public key certificate may only contain a pseudonym. The Certification Authority also issues short-lived PKIX X.509 attribute certificates with atomic attributes. The attribute certificates are associated with the public key certificate, and they may be stored in the user's wallet and/or in a central repository.

Second, the user authenticates to a relying party (with an access control system) by using the public key certificate. For example, TLS/SSL could be used for this authentication. If the public key certificate only contains a pseudonym of the user, the authentication protocol does not reveal the user's identity.

Third, the user's attribute certificate(s) are submitted to the relying party's access control system. The attribute certificate(s) may either be pushed from the client to the relying party, or pulled from the repository by the relying party.

For more information about attribute certificate architectures, see the IETF RFC 5755 [i.125].

An alternative design of using attribute certificates for anonymous authorization is described in the paper "A First Approach to Provide Anonymity in Attribute Certificates" [i.21] from 2004.

The PKIX X.509 certificates can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use, meaning that the attribute certificate access control solution meets the SOG-IS requirements on cryptographic algorithms.

6.2.2 VC-FIDO for atomic (Q)EAAs

Another example of a protocol for selective disclosure based on atomic (Q)EAAs is the VC-FIDO [i.50] integration that was invented at Kent University. The used atomic (Q)EAA format is W3C Verifiable Credential, which is described in clause 5.2.3.

In order to issue the atomic W3C Verifiable Credentials to an EUDI Wallet, the user needs to be identified or authenticated to a QTSP. The VC-FIDO integration is based on the W3C WebAuthn protocol in the FIDO2 standard. The WebAuthn [i.211] stack is extended with a W3C Verifiable Credentials enrolment protocol, resulting in a client that can enrol for multiple atomic short-lived W3C Verifiable Credentials based on W3C Credential templates. These atomic short-lived W3C Verifiable Credentials can then be (temporarily) stored in an EUDI Wallet, and be combined into a Verifiable Presentation that is presented to the relying party (verifier). Selective disclosure is achieved since the user can enrol for the atomic attributes it needs for a specific use case, and present only those atomic (Q)EAAs to a Relying Party.

The VC-FIDO integration was presented by David Chadwick at SHACK2020 [i.50]. This presentation explains the VC-FIDO architecture diagrams and shows a demo of how the client enrols for three atomic W3C Verifiable Credentials (address, driving license, and credit card) that are combined into a Verifiable Presentation as a parking ticket. The VC-FIDO integration is still a prototype, which is deployed as a pilot at National Health Services (NHS) in the UK.

The W3C Verifiable Credentials can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use, meaning that the VC-FIDO solution meets the SOG-IS requirements on cryptographic algorithms.

6.3 Multi-message signature protocols and solutions

6.3.1 Hyperledger AnonCreds (protocols)

The Hyperledger AnonCreds (Anonymous Credentials) specification [i.104] is based on the open source verifiable credential implementation of Hyperledger AnonCreds that has been in use since 2017. The Hyperledger AnonCreds software stack was initially implemented as a combination of the Hyperledger Aries [i.105] protocols, the Hyperledger Indy [i.107] credentials, and the Hyperledger Ursa [i.108] SDK with features for public/private key pair management, signatures and encryption. Since 2022 all Hyperledger AnonCreds features have been merged in the Hyperledger AnonCreds project. The Hyperledger AnonCreds credential format is described in clause 5.3.4.

Hyperledger AnonCreds are widely deployed, and are for example used by organizations such as the Government of British Columbia, IDunion, and the IATA Travel Pass.

6.3.2 Direct Anonymous Attestation (DAA) used with TPMs

Direct Anonymous Attestation (DAA) is a cryptographic protocol which enables remote authentication of a trusted computer yet preserving the privacy of the user.

ISO/IEC has standardized the DAA protocol in ISO/IEC 20008 [i.143]. The DAA protocol has been adopted by the Trusted Computing Group (TCG) in the Trusted Platform Module (TPM) v2.0 specification [i.192] to ensure the integrity of the computer yet addressing privacy concerns. Furthermore, Intel® has also adopted DAA in the Enhanced Privacy ID (EPID) 2.0 specification.

The primary scope of a TPM is to ensure the integrity of a computer and its operating system. The purpose is to ensure that the boot process starts from a trusted combination of hardware and software, and continues until the operating system has fully booted and applications are running in a trusted state. A computer that is running in a trusted state can be better controlled with respect to software licences and protection against computer viruses and malware.

The DAA eco-system consists of three entities: the DAA Member (i.e. TPM platform or EPID-enabled microprocessor), the DAA Issuer, and the DAA Verifier. The Issuer verifies the TPM platform during the Join step and issues a credential to the platform. The Member presents the credential to the Verifier during the Sign step; the Verifier can, based on a zero-knowledge proof, verify the credential without violating the platform's privacy. The DAA protocol also supports a blocklist such that Verifiers can prevent attestation attempts from TPMs that have been compromised.

Furthermore, the DAA protocol splits the signer role in two parts. In brief, a principal signer (a TPM) signs messages in collaboration with an assistant signer (the standard computer into which the TPM is embedded). This split aims to combine the high level of security provided by the TPM, and extend it by using the high level of computational and storage ability offered by the computing platform. Chen et al have specified the DAA protocol based on an ECC scheme [i.55] using Barreto–Naehrig curves, which is implemented by both TPM 2.0 and EPID 2.0.

The DAA protocol standardized in ISO/IEC 20008 [i.143], and implemented according to the TPM 2.0 and EPID 2.0 specifications, is considered mature and has been deployed at computers at a very large scale. Since the DAA protocol is based on an ECC scheme, it is however not considered as plausible quantum safe.

6.4 Salted attribute hashes protocols

6.4.1 OpenAttestation (Singapore's Smart Nation)

OpenAttestation, which is part of Singapore's Smart Nation initiative and developed within the GovTech's Government Digital Services, is an open source framework for verifiable documents and transferable records.

OpenAttestation allows a user to prove the existence and authenticity of a digital document. It makes use of smart contracts on the Ethereum blockchain to store cryptographic proof of individual documents. As an alternative to using the Ethereum blockchain, OpenAttestation can also be used to create verifiable documents using digital signatures.

More specifically, OpenAttestation provides Document Integrity [i.163] based on a target hash of salted attribute hashes. An overview of the OpenAttestation Document Integrity flow is illustrated in the figure below.

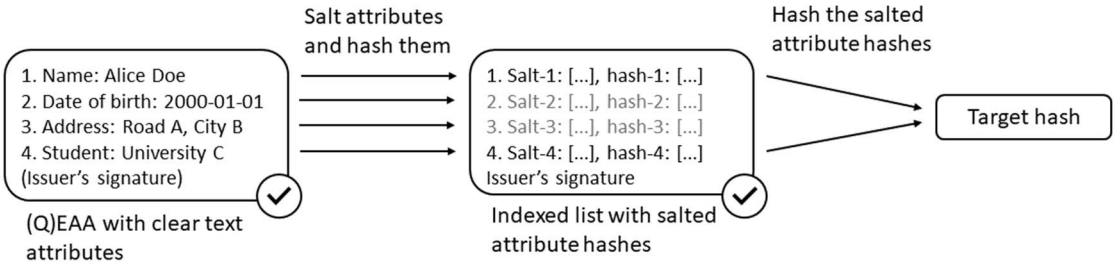

Figure 15: Overview of the OpenAttestation scheme

The target hash of the document is calculated as follows: Sort the selected salted attribute hashes from the previous step alphabetically and hash them all together. To compute the target hash the KECCAK256 algorithm is used.

During verification of the document, the same exact steps are performed again, and the result is compared to the target hash. If the two hash values match, the document integrity is intact.

Since the OpenAttestation scheme is based on salted attribute hashes, which can be signed with QSC algorithms, it can be considered as plausible quantum safe.

6.5 Proofs for arithmetic circuits solutions

6.5.1 Anonymous (Q)EAAs from programmable ZKPs and existing digital identities

6.5.1.1 Overview

This category is based on the principle of deriving anonymous (Q)EAAs by combining existing digital identities (such as X.509 certificates) with zero-knowledge proofs generated by general-purpose ZKP schemes (such as zk-SNARKs).

A generalized model of such systems is described in the paper "Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs" [i.12] by Babel and Sedlmeir. The solution, which can be divided in three phases, is illustrated in Figure 16.

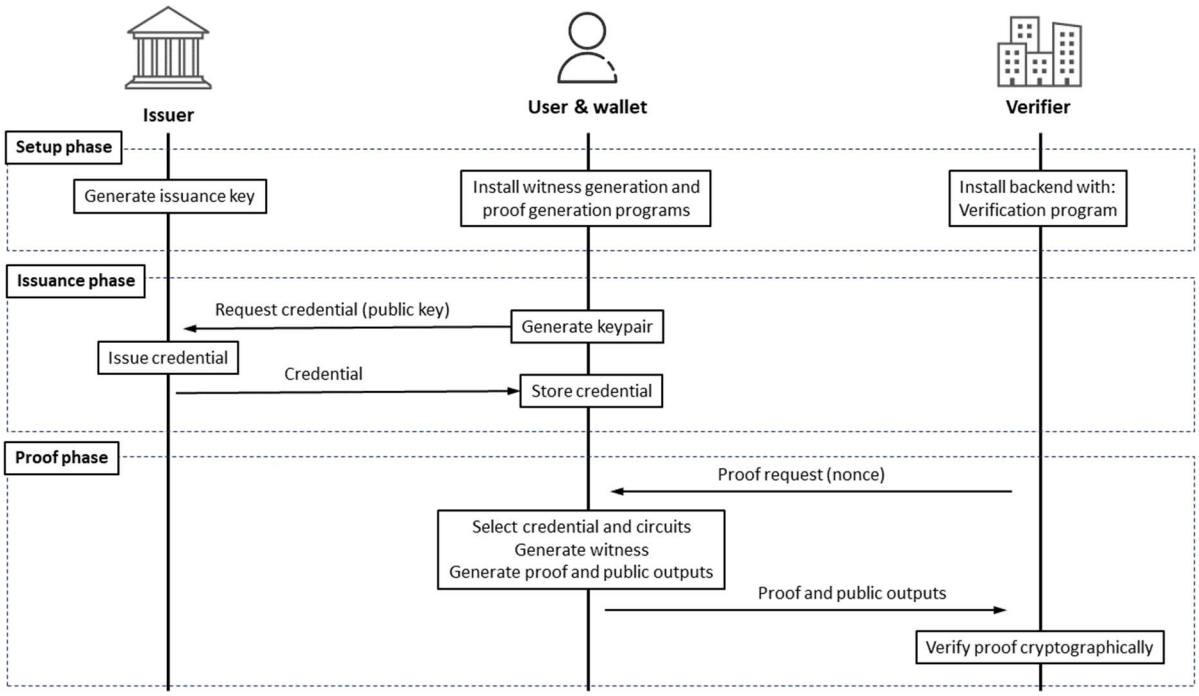

Figure 16: Overview of proofs used with credentials

6.5.1.2 Setup phase

In the setup phase, the issuer generates the issuance key. This could for example be a PKIX CA that issues X.509 certificates, or a PKD compliant CA that issues ICAO eMRTDs. The credential format, revocation scheme, etc., are typically also specified and implemented in this phase.

The digital wallet is provided with a witness generation program and a proof generation program, which implements the proofs for arithmetic circuits. Typically, the zk-SNARK circuits are integrated with the digital wallet by using a circuit compiler.

The verifier's backend is provided with the server-side circuits of the zk-SNARK scheme, which allows the verifier to validate the ZKPs generated by the digital wallet. The verifier in this scenario is equivalent to a relying party in the eIDAS2 context.

6.5.1.3 Issuance phase

During the issuance phase the digital wallet generates a key-pair and submits the public key in a credential request to the issuer. The issuer creates and signs the credential, for example an X.509 certificate, and returns it to the digital wallet where it is installed. The issuance phase can for example be performed as described in the ETSI EN 319 411-1 [i.75] standard for trust service providers issuing certificates..

6.5.1.4 Proof phase

The proof phase is initiated by the verifier, who submits a proof request (including a nonce) to the digital wallet. The user selects the credentials to be used for verification, and the digital wallet runs the verification algorithm using the locally stored credentials. The verification algorithm depends on the credentials framework, which could for example be a PKIX CA, ICAO PKD, or SSI type issuer of W3C VCs. The digital wallet also creates a ZKP that this verification algorithm was run correctly, without providing any further information than the statement provided by the verifier.

EXAMPLE: If a PKIX CA is used for issuance of X.509 certificates, the validation process should check that the user possesses the private key associated with the X.509 certificate, and that the X.509 certificate is valid (properly signed). The X.509 certificate status can be checked with respect to CA signature, expiry date, and revocation checks using OCSP.

The digital wallet executes the programmable ZKP scheme with the selected credential and its validity as private inputs. The digital wallet generates the witness, proof and public outputs and sends the ZKP result to the verifier. Hence, the digital wallet can use the ZKP scheme to submit the credential's verification result and selected attributes or predicates that need to be disclosed to the verifier. In order for the verifier to trust the verification result, the digital wallet also creates a ZKP that certifies the correct execution of the verification program, yet without sharing any details about the inputs or the results of the credential verification algorithm. Hence, the ZKP scheme can prove that the verification algorithm that was locally executed by the digital wallet resulted in the shared statement. The verifier can use the ZKP to check that the digital wallet has a credential that was indeed issued by a particular CA, and that the user possesses the private key associated with the public holder binding key referenced in the credential.

6.5.2 Cinderella: zk-SNARKs to verify the validity of X.509 certificates

The Cinderella project is described in the paper "Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation" [i.65] by Delignat-Lavaud et al. As indicated by the title, the project is an implementation of how to validate X.509 certificates locally at the digital wallet, and share the results with a verifier by using a ZKP scheme.

More specifically, the Cinderella project implemented a new format for application policies by composing X.509 templates, and provided a template compiler that translates C code for validating X.509 certificates within a given policy into an arithmetic circuit that allows for the generation of proving and verification programs. In order to produce a zero-knowledge verifiable computation scheme based on the Pinocchio [i.174] zk-SNARK, the Geppetto [i.60] cryptographic compiler was used.

The Cinderella project was evaluated by two real-world applications: a plug-in replacement for certificates within TLS [i.126], and access control for the Helios [i.1] voting protocol. Fine-grained validation policies were implemented for TLS with revocation checking and selective disclosure of certificate contents, which turn X.509 certificates into anonymous credentials. For Helios, additional privacy and verifiability guarantees for voters equipped with X.509 certificates were obtained, such as those currently available from certain national ID cards.

Rather than modifying the TLS standard and implementations, the X.509 certificate chains communicated during the TLS handshake were replaced with a single X.509 pseudo-certificate that carries a short-lived ECDSA public key and a proof that this key is properly signed with a valid RSA certificate whose subject matches the peer's identity. Also OCSP stapling can be communicated via the Cinderella version of TLS. National eID smartcards with X.509 certificates issued in Belgium, Estonia, and Spain have been evaluated with the Cinderella version of TLS.

One immediate issue is proving performance. Since the resulting Cinderella pseudo-certificates can take up to 9 minutes to generate for complex policies on a computer, it is recommended that they are generated offline and refreshed typically on a daily basis. Once the setup is configured or refreshed, online verification of the Cinderella pseudo-certificates and their embedded proof takes less than 10 ms. Yet, progress in zk-SNARK proving performance - e.g. lookup table with PLONKish arithmetization, assembly provers for mobile platforms, and tolerance of "bigger" proofs (hundreds of kilobytes) would arguably make a re-implementation of Cinderella practical on mobile phones, with proving times in the low double-digit seconds range.

NOTE: A vulnerability [i.103] in the Geppetto compiler that was found later would also require another toolchain to compile C-code to a ZKP (e.g. zk-SNARK) proving and verification algorithm.

6.5.3 zk-creds: zk-SNARKs used with ICAO passports

The zk-creds protocol was introduced in the paper "zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure" [i.183] by Rosenberg et al. The zk-creds protocol uses programmable ZKPs in the form of zk-SNARKs to:

- Remove the need for credential issuers to hold persistent signing keys. Instead, credentials can be issued to a bulletin board instantiated as a transparency log, a Byzantine system, or a blockchain.
- Convert existing identity documents into anonymous credentials without modifying documents or coordinating with their issuing authority.
- Allow for flexible, composable, and complex identity statements over multiple credentials.

The second use case has been implemented by generating ZKPs of ICAO compliant eMRTDs (passports) to create anonymous credentials for accessing age-restricted videos. More specifically, the eMRTDs were NFC-enabled and issued by the US State Department, which signs a hash tree of the eMRTD data with a raw RSA signature. The ZKP is essentially generated based on the eMRTD's Data Group 1 (DG1), which contains the textual information available on the eMRTD's data page and the Machine Readable Zone: name, issuing state, date of birth, and passport expiry.

6.5.4 Analysis of systems based on programmable ZKPs

The protocols that combine general-purpose ZKP schemes and digital identities provide some valuable characteristics:

- The existing digital identity infrastructures can be re-used as is, more specifically the eIDAS2 framework of X.509 certificates. This covers secure hardware for issuers' signing keys, secure hardware in mobile phones as commonly used with FIDO2. In particular, the issuance process would not need to be changed at all if the hardware attestation chain for the holder binding keypair is checked by the issuer in this step (which should usually be the case).
- The existing validation algorithm and revocation checking schemes can be executed in the digital wallet.
- Only the relevant information about the credential's validity and selected attributes or predicates need to be shared with the verifier because the holder also shares a zk-SNARK of correct local verification with the verifier.
- Both the credentials and zk-SNARK protocol can be designed with cryptographic algorithms that are plausible quantum-safe.
- Features such as very general predicates (e.g. proof of location within a certain region based on coordinates) and designated verifier proofs that can improve both security and privacy guarantees are easy to implement.
- Designated verifier properties that are challenging to achieve concurrently with unlinkability and non-interactiveness can be easily implemented. Designated verifier proofs allow the holder to make sure that only the designated recipient is convinced of the correctness of the verifiable presentation, mitigating risks of monetization of sensitive, attested (Q)EAA and of man-in-the-middle attacks.

However, the anonymous credential schemes described in this clause are still under research and development, and have not been deployed at scale. Hence, the maturity can be considered as low, although they provide a promising option for zero-knowledge proofs for the future of eIDAS2 and the EUDI Wallet. Moreover, yet, arithmetic circuits for commonly used cryptographic primitives, such as SHA256, RSA, and ECDSA are very complex and involve higher proving times than common digital signature schemes such as ECDSA. Proving time may be even worse for lattice-based post quantum secure digital signatures. The programmable ZKP systems that are most mature (zk-SNARKs) add some pronounced tradeoffs, e.g. the generality of preprocessing versus performance aspects.

6.6 Anonymous attribute based credentials systems

6.6.1 Idemix (Identity Mixer)

The Idemix (Identity Mixer) technology [i.109] was invented by IBM® Research in 2008. The Idemix system caters for strong authentication that is privacy preserving based on ABC (Attribute Based Credentials).

In summary, the Idemix scheme contains two protocols: Issuing the credential to a user and presenting it when accessing a relying party. An overview of the Idemix ABC scheme is illustrated in Figure 17.

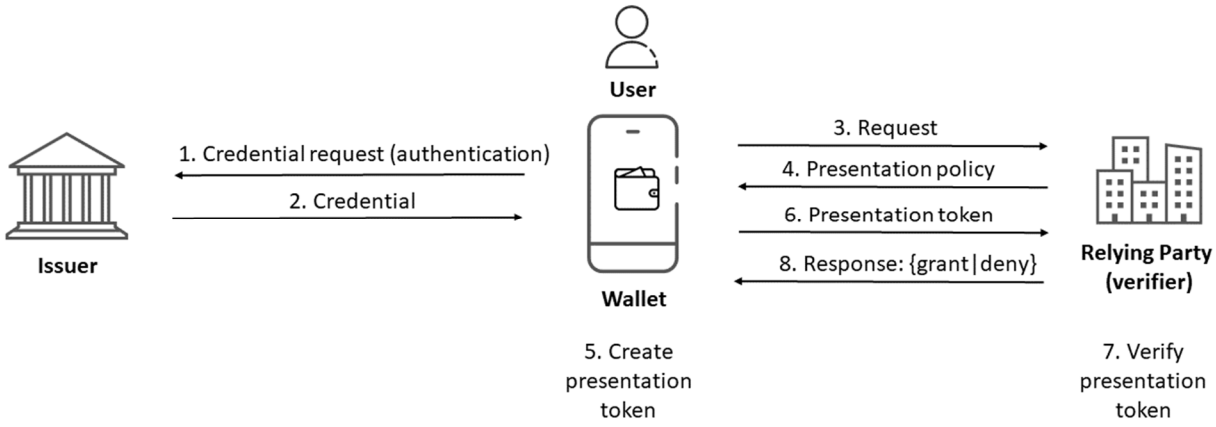

Figure 17: Overview of the Idemix ABC scheme

The Idemix system supports selective disclosure based on unlinkable Zero-Knowledge Proofs, such that users can prove that they are over 18 years old without revealing their name or birthdate. Idemix uses the pairing-based CL-signature scheme (clause 4.3.2) to prove knowledge of a signature in a Zero-Knowledge Proof.

NOTE 1: CL-signatures are not SOG-IS approved and not plausible quantum-safe.

The Idemix solution has been implemented by IBM® Identity Mixer [i.109], Hyperledger Fabric [i.106], Radboud University Nijmegen's IRMA project [i.180], and the EU-project PrimeLife [i.177]. The Idemix system was also selected as an ABC solution by the EC-funded project Attribute based Credentials for Trust (ABC4Trust) [i.110].

NOTE 2: Idemix is similar to the U-Prove (see clause 6.6.2) in the sense that both protocols are based on privacy-preserving ABC technology, although the iterations in the issuance phase and the underlying cryptographic algorithms differ.

NOTE 3: Idemix caters for multi-show unlinkability, whilst U-Prove does not [i.179].

The Idemix ABC system has been formalized by Camenisch et al in the paper "A Formal Model of Identity Mixer" [i.44] and the Idemix revocation mechanisms are discussed by Lapon et al in the paper "Analysis of Revocation Strategies for Anonymous Idemix Credentials" [i.155].

6.6.2 U-Prove

The U-Prove scheme is based on Attribute Based Credentials (ABC), which in turn relies upon Stefan Brand's cryptographic research on selective disclosure and blinded signature schemes in the book "Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy" from 2000 [i.31]. Brands founded a company to implement the U-Prove ABC scheme, and this company was later acquired by Microsoft®. In 2013, Microsoft® Research released the Identity Metasystem with support for U-Prove ABC to cater for anonymous credentials [i.160]. The U-Prove ABC system was also selected by the EC-funded project Attribute based Credentials for Trust (ABC4Trust) [i.110].

In summary, the U-Prove scheme contains two protocols: Issuing the credential to a user and presenting it when accessing a relying party. The U-Prove scheme is illustrated in Figure 18.

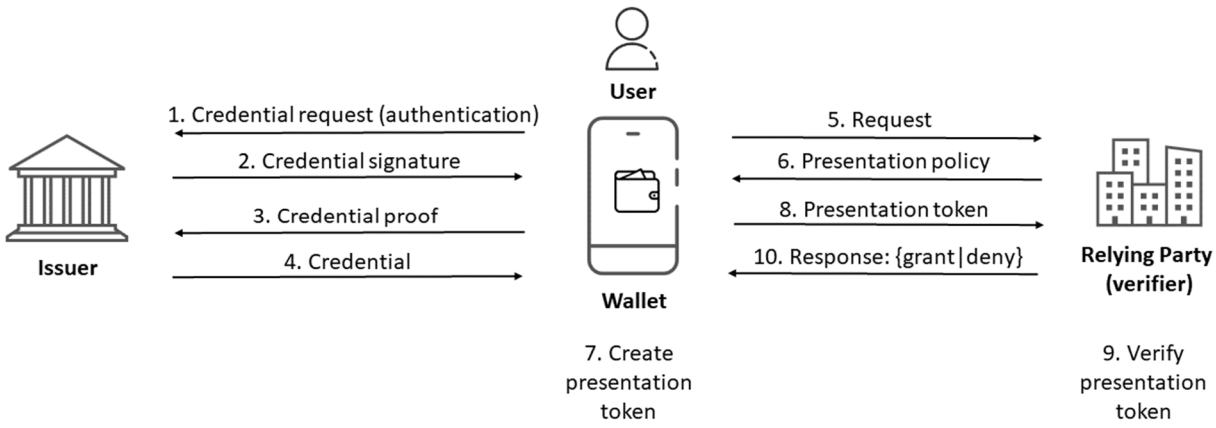

Figure 18: Overview of the U-Prove ABC scheme

The U-Prove issuing protocol is performed between the issuer and the user. The objective of this protocol is for the user to receive a credential, such that it can later present a selected set of attributes to access a relying party. The issuer basically applies a blind signature to the credential with attributes. In other words, the issuer verifies the validity of the attributes and applies a signature without seeing the resulting signature. Since the issuer does not store the result of the issuing protocol, the user cannot be tracked when using the credential, i.e. the processes of issuing and presenting are unlinkable.

The U-Prove presentation phase is based on a selective disclosure protocol between the user and the relying party. Based on the relying party's presentation policy, the user selects those attributes that it is willing to present from the issued credential. All the other attributes can be proved by the user to be unchanged in the credential. By the end of the interaction the relying party receives a presentation token with all the revealed attributes and the intact issuer's signature on the whole set of attribute values.

NOTE 1: U-Prove is similar to the Idemix (see clause 6.6.1) in the sense that both protocols are based on privacy-preserving ABC technology, although the iterations in the issuance phase and the underlying cryptographic algorithms differ.

The U-Prove scheme is based on the DLP and the credentials are issued as DLREP-based certificates as well as for RSAREP-certificates.

NOTE 2: Since U-Prove is based on algorithms using the DLP, the scheme cannot be considered as quantum-safe.

NOTE 3: Idemix caters for multi-show unlinkability, whilst U-Prove does not [i.179].

6.6.3 ISO/IEC 18370 (blind digital signatures)

The ISO/IEC 18370 [i.142] series standardize blind digital signature protocols. Whereas, ISO/IEC 18370-1:2016 describes an overview of blind digital signature solutions, ISO/IEC 18370-2:2016 specifies discrete logarithm based mechanisms.

More specifically, section 8 of ISO/IEC 18370-2:2016 specifies a DLP-based blind signature protocol with selective disclosure capabilities. Actually, mechanism 4 described in section 8 of ISO/IEC 18370-2:2016 is a standardization of Microsoft® U-Prove anonymous credential system (see clause 6.6.2).

Since ISO/IEC 18370 [i.142] is an international standard, which has the potential status to be referenced by EU regulations. This begs the question if ISO/IEC 18370 [i.142] could serve as a standardized selective disclosure protocol for the EUDI Wallet. There are however two critical issues associated with ISO/IEC 18370 [i.142].

The first critical issue with mechanism 4 described in section 8 of ISO 18370-2:2016 (i.e. U-Prove) is that it does not provide multi-show unlinkability. In other words, it is only possible to present a U-Prove credential once, thereafter additional presentations of the U-Prove credential are linkable.

The second issue is that the U-Prove scheme is broken under certain conditions, as described in the article "On the (in)Security of ROS" [i.20]. Provided that the U-Prove issuance protocol is executed concurrently, it is possible to forge a U-Prove credential. However, U-Prove will remain secure if the issuance protocol is only executed sequentially, but this would not be practical nor user-friendly.

Since ISO/IEC 18370-2:2016 is based on algorithms using the DLP, the scheme cannot be considered as quantum-safe.

Hence, the ISO/IEC 18370 standard [i.142] on blind signatures is not recommended to be considered as a selective disclosure protocol for the EUDI Wallet.

6.6.4 Keyed-Verification Anonymous Credentials (KVAC)

The anonymous credentials systems Idemix (clause 6.6.1) and U-Prove (clause 6.6.2) are based on public key primitives. A different approach, that is based on algebraic Message Authentication Codes (MACs) in prime-order groups, was proposed by Chase et al in the paper "Algebraic MACs and keyed-verification anonymous credentials" [i.52]. The paper describes two anonymous credentials systems called "Keyed-Verification Anonymous Credentials (KVAC)" as they require the verifier to know the issuer secret key. The KVAC system is based on two algebraic MACs in prime-order groups, along with protocols for issuing credentials, asserting possession of a credential, and proving statements about hidden attributes (e.g. the age of the user). The performance of the KVAC schemes is comparable to U-Prove and faster than Idemix. However, the presentation proof, for n unrevealed attributes, is of complexity O(n) in the number of group elements.

In order to address the complexity issue, a new KVAC system has been designed that provides multi-show unlinkability of credentials and is of complexity O(1) in the number of group elements. This enhanced KVAC scheme was described by Barki et al in the paper "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials" [i.13]. A new algebraic MAC_BBS+ scheme based on a pairing-free variant [i.46] of BBS [i.25] is also described in the paper.

This KVAC system is suitable for resource constrained environments like SIM-cards, and MAC_BBS+ has been implemented as a prototype on standard SIM-cards. Only the verification process differs between the MAC_BBS+ and BBS+ versions but all other operations remain the same (such as credentials issuance and generation of verifiable presentations). The MAC_BBS+ signatures are therefore equivalent to BBS+ signatures for the KVAC system as a whole. Hence, the verification of a MAC_BBS+ verifiable presentation can be done more efficiently and without pairings, provided that the verifier and the issuer are the same entity and therefore share the issuance private key. This could for example be the case for instance in e-voting or public transportation use cases, where the voting authority respectively public transportation authority manages the virtual ballot box server respectively turnstiles/validators. The BBS+ variant of the KVAC system, which can be seen as the public-key variant of MAC_BBS+, is described in clause 4.3 in the paper "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials" [i.13].

6.7 ISO mobile driving license (ISO mDL)

6.7.1 Introduction to ISO/IEC 18013-5 (ISO mDL)

The ISO mobile driving license (ISO mDL) is specified in the ISO/IEC 18013-5 [i.140] standard, which on a high level can be divided in the device retrieval flow (see clause 6.7.2) and the server retrieval flows (see clause 6.7.3) for selective disclosure of the user's mDL mdoc.

ISO/IEC CD 18013-7 [i.141] is a draft specification that extends the ISO/IEC 18013-5 [i.140] standard with unattended flows (see clause 6.7.3), which are online protocols for selective disclosure of the user's mDL mdoc to a web hosted ISO mDL reader.

6.7.2 ISO/IEC 18013-5 (device retrieval flow)

The ISO mDL device retrieval flow is described in ISO/IEC 18013-5 [i.140], clauses 6.3.2, 6.3.2.1 (as flow 1) and 6.3.2.4.

The credential format is the ISO mDL mdoc, which contains the attributes about the user, in conjunction with the Mobile Security Object (MSO). The MSO is a signed object that contains a list of salted attribute hashes of the user's mDL attributes. The MSO caters for selective disclosure based on the salted attribute hashes as described in clause 5.4.2.

The selected attributes of the ISO mDL mdoc and the MSO are presented by the user's ISO mDL app to an ISO mDL reader by using BLE, NFC or WiFi. The ISO mDL reader verifies the MSO and the selectively disclosed attributes (see clause 7.2.4 for more information on the ISO mDL device retrieval flow).

ISO/IEC 18013-5 [i.140] is considered mature, and several ISO mDL device retrieval solutions have been deployed in production, for example in a number of states in the US.

The ISO mDL MSO and DeviceSignedItems can be signed with cryptographic algorithms that are currently approved by SOG-IS [i.188]. Since the MSO and DeviceSignedItems are signed with a COSE-formatted signature, this caters for MSOs to be signed in the future with QSC algorithms as discussed in the IETF report "JOSE and COSE Encoding for Post-Quantum Signatures" [i.119].

NOTE: Although DeviceSignedItems can be signed with candidate quantum-safe signatures, the issue of having a quantum-safe key agreement mechanism to secure the communication channel remains. The ephemeral session keys between the ISO mDL device and the reader are currently exchanged using the ECKA-DH key agreement, which is vulnerable to quantum computing attacks. Furthermore, MAC signatures are mentioned in ISO/IEC 18013-5 [i.140] as offering better privacy guarantee, but the MAC secret is derived from an ECKA-DH key agreement, which is exposed to the quantum computing vulnerability. An extensive analysis of the ISO mDL session key exchange goes beyond the scope of the present document, however, but this quantum computing vulnerability should be observed.

The ISO mDL device retrieval flow has been selected as a PID protocol for the EUDI Wallet as specified in the ARF [i.59].

An extensive analysis of the ISO mDL device retrieval flow, and how it can be applied for eIDAS2 QTSPs and EUDI Wallet PID/(Q)AEE, is available in clause 7.2.3.

6.7.3 ISO/IEC 18013-5 (server retrieval flows)

The ISO mDL server retrieval flows are described in ISO/IEC 18013-5 [i.140], clause 9.2.

The ISO mDL server retrieval flow can be initialized as a hybrid device/server process (see clause 7.2.4.2) or as a server process (see clause 7.2.4.3). Once the ISO mDL server retrieval flow has been initialized, it continues with either the WebAPI flow or the OpenID Connect (OIDC) flow.

In the WebAPI flow the mDL Reader submits a server retrieval WebAPI Request with a list of requested DataElements to the Issuing Authority. Upon the user's consent, the Issuing Authority will reply with the mDL Response with the selected and disclosed DataElements (see clause 7.2.4.4 for more information).

In the OIDC flow the mDL Reader (OIDC client) submits a server retrieval OIDC Request with the requested data elements (JWT claims) to the Issuing Authority, which operates an OIDC Authorization Server. This activates the OIDC authorization code flow [i.170]. Based on the user's consent, the Issuing Authority (OIDC Authorization Server) will reply to the mDL Reader (OIDC client) with the OIDC Token with the selected and disclosed JWT claims about the user (see clause 7.2.4.5 for more information).

ISO/IEC 18013-5 [i.140] and OIDC standards are considered mature, and several ISO mDL server retrieval solutions have been deployed in production, for example in a number of states in the US.

The WebAPI and OIDC tokens are JWTs that can be signed with cryptographic algorithms that are currently approved by SOG-IS [i.188]. Since the WebAPI and OIDC tokens are signed with a JOSE-formatted signature, this caters for those JWTs to be signed in the future with QSC algorithms as discussed in the IETF report "JOSE and COSE Encoding for Post-Quantum Signatures" [i.119].

An extensive analysis of the ISO mDL server retrieval flow, and how it can be applied for eIDAS2 QTSPs and EUDI Wallet PID/(Q)AEE, is available in clause 7.2.4.

6.7.4 ISO/IEC 18013-7 (unattended flow)

ISO/IEC CD 18013-7 [i.141] draft standard extends ISO/IEC 18013-5 [i.140] with the unattended flow, i.e. the online flow whereby an ISO mDL app connects directly to an mDL reader that is hosted as a web server application. ISO/IEC 18013-7 [i.141] is backward compatible with the protocols specified in ISO/IEC 18013-5 [i.140].

ISO/IEC CD 18013-7 [i.141] unattended flow is based on the following protocols:

- Device Retrieval from an ISO mDL app to a web server application by using REST APIs over HTTPS POST; this flow is described in clause 7.2.5.1.
- OpenID for Verifiable Presentations (OID4VP) [i.171] in conjunction with Self-issued OpenID Provider v2 (SIOP2) [i.173]; this flow is described in clause 7.2.5.2.

Both protocols for the unattended flow transmit the selectively disclosed ISO mDL attributes in conjunction with the MSO from the ISO mDL app to the ISO mDL reader. The ISO mDL attributes and the MSO are verified according to the same principles as for the ISO mDL device retrieval flow (see clause 7.2.3).

As described in clause 6.7.1, the MSO can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use.

ISO/IEC CD 18013-7 [i.141] is still a draft, so there are no real deployments in production. NIST NCCoE will carry out interoperability tests [i.165] with an ISO/IEC CD 18013-7 [i.141] compatible reader during the course of 2023 and 2024.

The ISO mDL proximity unattended flow has been selected as a PID protocol for the EUDI Wallet as specified in the ARF [i.59].

An extensive analysis of the ISO mDL unattended flow, and how it can be applied for eIDAS2 QTSPs and EUDI Wallet PID/(Q)AEE, is available in clause 7.2.5.

6.7.5 ISO/IEC 23220-4 (operational protocols)

ISO/IEC CD 23220-4 [i.146] is a draft specification describing operational (presentation) protocols for a digital wallet. The specification expands on ISO/IEC 18013-5 [i.140] with reader engagement, internet online connections to a reader, and bridges to additional standards for user authorization such as OID4VP [i.171] and credential formats such as W3C Verifiable Credentials [i.209].

ISO/IEC CD 23220-4 [i.146] presentation protocols are based on the following protocols:

- Device Retrieval from a digital wallet to a web server application by using REST APIs over HTTPS POST.
- OpenID for Verifiable Presentations (OID4VP) [i.171] in conjunction with Self-issued OpenID Provider v2 (SIOP2) [i.173].

More specifically, Annex B in ISO/IEC CD 18013-7 [i.141] draft specification refers to ISO/IEC CD 23220-4 [i.146] for the OID4VP/SIOP2 profile to be used for presentation of the ISO mDL and an MSO in an ISO/IEC CD 18013-7 [i.141] unattended flow. As described in clause 6.7.1, the MSO can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use.

Furthermore, Annex B in ISO/IEC CD 23220-4 [i.146] WD9 describes how to present W3C Verifiable Credentials [i.209] in conjunction with IETF SD-JWT [i.123] for selective disclosure. The SD-JWT can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use (see clause 7.3).

In order to secure the HTTPS connection to an online reader (relying party), ISO/IEC 23220-4 [i.146] recommends the use of QWACs.

ISO/IEC CD 23220-4 [i.146] is still a draft, so there are no real deployments in production. However, the ARF [i.59] refers to ISO/IEC CD 23220-4 [i.146] as an alternative attestation exchange REST API protocol.

7 Implications of selective disclosure on standards for (Q)EAA/PID

7.1 General implications

The purpose of clause 7 is to analyse the implications of selective disclosure and unlinkability on ETSI standards for (Q)EAAs and PIDs.

More specifically, the (Q)EAA/PID credentials discussed in the following clauses 7.2 and 7.3 are scoped to ISO/IEC 18013-5 [i.140] mDL and SD-JWT, because these formats are explicitly specified as selective disclosure formats for PIDs in the ARF [i.59]. The main reason why ISO mDL and SD-JWT were selected in the ARF [i.59] as (Q)EAA/PID credentials is that they can be signed with cryptographic algorithms that are currently approved by SOG-IS [i.188], and that the credentials also allow for being signed with Quantum-Safe Cryptography (QSC) algorithms for future use. More technical details on how the issuer may apply such signatures on ISO mDL and SD-JWT are discussed in clauses 7.2.1 and 7.3.1 respectively.

Furthermore, clause 7.4 analyses the possibilities of using BBS+ credentials as (Q)EAA/PID. The reason for analysing BBS+ is due to the emerging ISO standardization of BBS+, which may be used with W3C VCDM in conjunction with W3C VCDI. Since BBS+ with blinded signatures is fully unlinkable, it would be a viable alternative from a privacy preserving perspective. This in turn may cater for BBS+ to be referenced in a future version of the ARF and/or the ETSI TS 119 472-1 [i.81] standard on (Q)EAAs profiles.

Also, clause 7.5 analyses solutions that utilize programmable ZKPs such as zk-SNARKs in conjunction with existing digital infrastructures. The reason for analysing such solutions is that they can provide fully unlinkable presentations that provide selectively disclosed attributes and revocation information, based on existing eIDAS X.509 QCs and the forthcoming eIDAS2 (Q)EAAs/PIDs. This in turn may cater for zk-SNARK based solutions to be referenced in a future version of the ARF and/or the ETSI TS 119 462 [i.79] standard on EUDI Wallet interfaces.

The analysis in clause 7 is primarily focused on selective disclosure and unlinkability since those characteristics are defined in eIDAS2 [i.86] and the ARF outline [i.58]. Predicates are described on a high level, with proposals on how to implement them for the selected PID credentials ISO mDL and SD-JWT.

The selected (Q)EAA/PID credentials are analysed with respect to the issuance by a QTSP/PIDP, how the credentials are stored in the EUDI Wallet, and how selected attributes are presented to a relying party.

Firstly, it is analysed how the QTSP or PID provider may issue (Q)EAAs/PIDs with capabilities for selective disclosure. This analysis also describes the PKI trust models for the issuance process and whether EU Trusted Lists (EU TLs) can be applied. Furthermore, it is described how the (Q)EAAs/PIDs should be issued to cater for unlinkability. The recommended policies and practices for such QTSP/PIDP issuance processes are discussed for ISO mDL in clause 7.2 and SD-JWT in clause 7.3.

Secondly, it is analysed how the (Q)EAAs/PIDs with capabilities for selective disclosure and unlinkability are stored in the EUDI Wallet. This analysis also describes the associated cryptographic keys used for proving the user's ownership of the (Q)EAAs/PIDs. The implications for storing the (Q)EAAs/PIDs with selective disclosure in an EUDI Wallet are discussed for ISO mDL in clause 7.2 and SD-JWT in clause 7.3.

Thirdly, it is analysed how the selected attributes can be presented to a relying party, yet sustaining unlinkability. The recommended policies and practices for presenting the (Q)EAAs/PIDs with an EUDI Wallet are discussed for ISO mDL in clause 7.2 and SD-JWT in clause 7.3.

7.2 Implications for ISO mDL with selective disclosure

7.2.1 QTSP/PIDP issuing ISO mDL

7.2.1.1 General

The ISO mDL, as specified in ISO/IEC 18013-5 [i.140], is composed by the ISO mDL mdoc with the user's elements, the ISO mDL authentication key, and the Mobile Security Object (MSO) with a signed list of salted hash values of these elements. The MSO is a CDDL-encoded [i.136] object, which is signed by the issuer with a COSE-formatted signature [i.133].

ISO/IEC 18013-5 [i.140] describes the Issuing Authority Certification Authority (IACA) that is the root CA that used for issuing subordinated certificates, which in turn are used for signing the user's ISO mDL MSOs, signing revocation data (OCSP-responses and CRLs), and securing online services (JWS and TLS).

The clauses below compare and map the requirements on ISO mDL compliant IACAs into considerations for eIDAS2 compliant QTSPs/PIDPs when issuing ISO mDL with capabilities for selective disclosure and (predetermined) predicates. The clauses below also provide a summary of the ISO mDL and its Issuing Authorities, but it is recommended to have studied the ISO/IEC 18013-5 [i.140] before to have an understanding of the ISO mDL ecosystem.

7.2.1.2 Certificate profiles

The IACA's trust anchor is a DER-encoded X.509 certificate that should be issued according to the certificate profile in ISO/IEC 18013-5 [i.140], Annex B.1. ISO/IEC 18013-5 [i.140], Annex B.1.1 declares that all X.509 certificates are DER-encoded and specifies the generic certificate requirements on certificate extensions and subjects. The IACA certificate profile also defines the cryptographic algorithms that are approved by ISO/IEC 18013-5 [i.140].

In the context of eIDAS2, the cryptographic algorithms used in the QTSP/PIDP CA certificates are required to comply with the SOG-IS list of EU approved cryptographic algorithms [i.188]. Hence, the QTSP/PIDP CA certificates used for issuing ISO mDLs are required to comply with the intersection of IACA's and SOG-IS' requirements on cryptographic algorithms.

EXAMPLE 1: SOG-IS [i.188], section 4.3 "Discrete Logarithm in Elliptic Curves" lists the following approved ECC curves: BrainpoolP256r1, BrainpoolP384r1, and BrainpoolP512r1.

EXAMPLE 2: ISO/IEC 18013-5 [i.140], Table B.3 "Document signer certificate" lists the following approved ECC curves: BrainpoolP256r1, BrainpoolP320r1, BrainpoolP384r1, BrainpoolP512r1, Curve P-256, Curve P-384, and Curve P-521.

The IACA trust anchor is used for issuing the following subordinated certificates in an IACA PKI:

- mDL MSO signer certificate (ISO/IEC 18013-5 [i.140], Annex B1.2).
- JWS signing certificate (ISO/IEC 18013-5 [i.140], Annex B.1.3.1).
- TLS server certificate issuing authority (ISO/IEC 18013-5 [i.140], Annex B1.6).
- TLS client authentication certificate (ISO/IEC 18013-5 [i.140], Annex B.1.8).
- OCSP signer certificate (ISO/IEC 18013-5 [i.140], Annex B.1.9).

Furthermore, the ISO mDL IACA CRL profile is specified in Annex B.2 in ISO/IEC 18013-5 [i.140].

An eIDAS2 QTSP/PIDP that issues ISO mDLs should adhere to the IACA PKI and the certificate and CRL profiles described above.

One more alternative could be for ETSI to assign a specific QC extension to be used for trust anchor certificates that are used by accredited QTSPs to issue ISO mDLs.

7.2.1.3 Trusted Lists

According to article 22(1) of eIDAS [i.87], each EU Member State is required to publish a Trusted List (TL) with all QTSPs in that EU Member State. All information referred to in eIDAS article 22(3), including the location and signing certificates of the TLs, is compiled in the EU LOTL (List Of Trusted Lists). Furthermore, the Commission Implementing Directive (CID) 2015/1505 [i.84] mandates the use of ETSI TS 119 612 [i.78] for the implementation of the trusted lists. ETSI TS 119 612 [i.78] specifies the format and mechanisms for establishing, locating, accessing and authenticating trusted lists. The EU TLs and EU TOTL are XML-encoded according to specific XML schemas and signed with XAdES-signatures as specified in ETSI TS 119 612 [i.78].

ISO/IEC 18013-5 [i.140] has introduced a similar concept called Verified Issuer Certificate Authority List (VICAL), which contains the trustworthy IACA's that issue certificates for creating and operating ISO mDLs. An ISO mDL VICAL can be formatted and signed either in CDDL [i.136] or CMS [i.124] format. The ISO mDL VICAL Providers publishes the VICALs. ISO/IEC 18013-5 [i.140], Annex C specifies the policy and security requirements and technical and procedural controls for a VICAL Provider.

NOTE: ISO/IEC 18013-5 [i.140], Annex C refers to ETSI EN 319 411-1 [i.75] and FPKIPA X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework [i.91] for the operations of an ISO mDL VICAL Provider.

Hence, there are synergies between the EU TLs and the ISO mDL VICALs, in the sense that both trusted lists contain trust anchors. The main differences are the encodings and signature formats (EU TL XML/XAdES versus ISO mDL VICAL CDDL/CMS). In order to bridge this gap, ETSI TS 119 612 [i.78] may specify a CDDL/CMS profile of the EU TL that is compatible with the ISO mDL VICAL, or ISO/IEC 18013-5 [i.140] may be extended to specify an XML profile of the VICAL that is compatible with the ETSI EU TLs. In such a scenario, an eIDAS2 accredited QTSP/PIDP could issue CA certificates that are included in an EU TL, which in turn could be trusted as a VICAL in the ISO mDL ecosystem.

In summary, transposing ISO/IEC 18013-5 [i.140], Annex C to an eIDAS2 context results in the following recommendations:

- The ISO mDL Issuing Authority corresponds to the eIDAS2 QTSP/PIDP.
- The IACA trust anchor should be issued as a trust anchor by the eIDAS2 QTSP/PIDP that issues ISO mDL as (Q)EAA/PID.
- The eIDAS2 QTSP/PIDP should ensure that its IACA trust anchor is published in the EU TL, which is issued by the supervisory body in the applicable EU Member State.
- ETSI TS 119 612 [i.78] may specify an additional CDDL/CMS profile of the EU TL that is compatible with the ISO mDL VICAL, or ISO/IEC 18013-5 [i.140] may be extended to specify an XML profile of the VICAL that is compatible with the ETSI EU TLs.
- The EU TLs may include a specific extension for the QTSPs that are authorized to issue QEAAs that also are compliant with ISO mDL; the EU TL extension can reference the ISO mDL VICAL where the QTSP is also listed.

7.2.1.4 Issuance of ISO mDLs

An ISO mDL, which has been issued to the user's EUDI Wallet on a device, is essentially composed of the mDL mdoc and the MSO, which are associated with the mDL authentication key (see clause 7.2.2).

The ISO mDL mdoc is an unsigned list of the user's elements belonging to the nameSpace "org.iso.18013.5.1", as defined in ISO/IEC 18013-5 [i.140].

The MSO (mobile security object) is defined in ISO/IEC 18013-5 [i.140], section 9.1.2.4 as a signed object, which contains the mDL authentication public key and a list of salted attribute hashes of the user's elements. The MSO is signed with a COSE-formatted signature, by the IACA's MSO signer certificate.

NOTE 1: In the context of eIDAS2, a QTSP/PIDP will issue an MSO signer certificate with cryptographic algorithms that are approved by both SOG-IS [i.188] and the ISO/IEC 18013-5 [i.140].

NOTE 2: Since the MSO's signature is COSE-formatted, QSC algorithms can also be considered for the future according to the IETF IESG report [i.119].

According to section E.8.4 of ISO/IEC 18013-5 [i.140] and section E.8.4 and section E.5 of ISO/IEC CD 23220-3 [i.145] it is recommended that the mDL authentication keys and related MSOs are updated frequently to achieve unlinkability when presenting the ISO mDL elements multiple times . Hence, the QTSP/PIDP should establish processes for issuing multiple MSOs to the user's EUDI Wallet, typically in batches prior to the device retrieval use of the MSOs. The EUDI Wallet may also signal to the QTSP/PIDP when it is necessary to refresh the MSOs. When issuing a new MSO, the random salts in IssuerSignedItems for the hash calculations should be unique such that the random salted hash values differ for each MSO, even if the user's ISO mDL mdoc elements remain the same.

EXAMPLE 1: Assume that the user's GivenName in the ISO mDL mdoc is "Smith". If the GivenName is combined with random salt S1 and hashed, the resulting hash value becomes H1 in the first MSO. If the same GivenName name is combined with another random salt S2 and hashed, the resulting hash value becomes H2 in the second MSO.

ISO mDL does not support predicates in the sense that Zero-Knowledge Proofs or range proofs can be dynamically derived based on the elements in the ISO mDL mdoc. However, ISO/IEC 18013-5 [i.140], clause 7.2.5 specifies the possibility to insert predetermined Boolean elements as "age_over_NN" in the ISO mDL.

EXAMPLE 2: The Boolean statement "age_over_18" could be an element in the ISO mDL mdoc.

NOTE 3: It is possible to include signed computational inputs and parameters to enable dynamic predicates (see clause 4.4.7.4).

In order to achieve (predetermined) predicates, the issuing QTSP/PIDP should establish processes to identify the relevant Boolean statements and insert them as elements in the ISO mDL.

7.2.1.5 Comparison with ETSI certificate profiles for Open Banking (PSD2)

ETSI TC ESI has specified certificate profiles and TSP policy requirements for Open Banking in the sector specific ETSI TS 119 495 [i.77]. The scope of ETSI TS 119 495 [i.77] is:

- Specifies requirements for qualified certificates for electronic seals and website authentication, to be used by payment service providers in order to meet needs of Open Banking including the EU PSD2 [i.85] Regulatory Technical Standards (RTS) [i.82].
- Specifies additional TSP policy requirements for the management (including verification and revocation) of additional certificate attributes as required by the above profiles.

In summary, a QTSP can issue PSD2 compliant certificates (QWACs or QCert for eSeal), using the certificate profile specified in ETSI TS 119 495 [i.77] as follows. The PSD2 specific attributes are checked by the (Q)TSP as part of the identity proofing, as specified in the ETSI TS 119 495 [i.77], REG-6.2.2-1, which states: *"The TSP shall verify the Open Banking Attributes (see clauses 5.1 and 5.2) provided by the subject using authentic information from the Competent Authority (e.g. a national public register, EBA PSD2 Register, EBA Credit Institution Register, authenticated letter)."* The EBA (European Banking Association) maintains a register of payment institutions [i.71], which can be used for that purpose. As a result, a QCStatement extension with Open Banking attributes is included in the PSD2 certificate, which proves its compliance with the PSD2 RTS.

A relying party intending to validate a PSD2 certificate usually performs a two step validation approach:

1) The relying party validates the qualified status of the certificate using the EU TLs.
2) The relying party confirms the correctness of the PSD2 attributes included in the certificate QCStatement using either the national public registers, or the EBA register. The relying parties need to have out-of-band knowledge of where to retrieve the EBA register.

The ETSI TS 119 495 [i.77] requirements for (Q)TSPs issuing PSD2 certificates may partially be re-used also for the issuance of ISO mDLs, but with the following differences:

- The format will be (Q)EAA for ISO mDL instead of X.509 certificates.
- The relying party will confirm that the QTSP having issued the (Q)EAA is authorized to issue this specific type of (Q)EAA by looking into a domain-specific list, i.e. the ISO mDL VICAL.

- To facilitate the validation of (Q)EAAs being used ISO mDLs, EU TLs could be used to point towards the domain-specific VICAL list where a QTSP is listed as being authorized for a specific scope. Alternatively, an URI for accessing this domain-specific VICAL list could be included in the ISO mDL (Q)EAA itself, although this may be too static as this URI may change over time.

7.2.1.6 Mapping of ISO mDL and eIDAS2 terms

As discussed in the clauses above, there are several equivalences between the terms in ISO/IEC 18013-5 [i.140] and the terms in eIDAS2 [i.86] and the ARF [i.59].

Table 1 provides a mapping of eIDAS2 and ARF terms with the syntax used in ISO/IEC 18013-5 [i.140].

Table 1: Mapping of eIDAS2/ARF and ISO/IEC 18013-5 terms

Terms in eIDAS2 and the ARF	**Terms in ISO/IEC 18013-5** [i.140] **(mDL)**
End users of EUDI Wallets	mDL Holder
EUDI Wallet issuers	Technology Providers
Person Identification Data Providers	Issuing Authorities
Providers of registries of trusted sources (e.g. EU TL)	Verified Issuer Certificate Authority List (VICAL) Providers
Qualified and non-qualified electronic attestation of attributes (qEAA) providers	Issuing Authorities
QTSPs for issuing qualified and non-qualified certificate for electronic signature/seal providers	Issuing Authority Certification Authority (IACA)
Providers of other trust services	Not defined
Authentic sources	Governmental authoritative source
Relying parties	mDL Reader, operated by a mDL verifier
Conformity Assessment Bodies (CAB)	Auditing Bodies following ISO/IEC 27001 [i.148] and ISO/IEC 27002 [i.149]
Supervisory bodies	Auditing Bodies following ISO/IEC 27001 [i.148] and ISO/IEC 27002 [i.149]
Device manufacturers and related subsystems providers	Technology Providers
Catalogue of attributes and schemes for the attestations of attribute providers	ISO mDL namespace

7.2.2 EUDI Wallet mDL authentication key

The mDL authentication key is used to prevent cloning of the ISO mDL and to mitigate man in the middle attacks. The mDL authentication key pair consists of a public and a private key denoted as (SDeviceKey.Priv, SDeviceKey.Pub). The mDL authentication public key is stored as the DeviceKey element in the MSO, and the corresponding mDL authentication private key is used for signing the response data contained in the DeviceSignedItems structure (see ISO/IEC 18013-5 [i.140], clauses 9.1.3, 9.1.2.4 and 9.1.3.3 for more information).

Hence, the mDL authentication key is used by the EUDI Wallet for authentication of selectively disclosed mDL mdoc elements that are presented to a relying party (see clause 7.2.3).

More information on how to store the ISO mDL mdoc, MSO, and the mDL authentication key is available in clause 7.6.

See also clause 4.4.4.2 on the possibility to use Hierarchical Deterministic Key derivation functions where the MSO issuer can issue a batch of MSOs, each with a unique and unlinkable DeviceKey element derived from a single DeviceKey element.

7.2.3 EUDI Wallet used with ISO mDL device retrieval flow

7.2.3.1 Overview of the ISO mDL device retrieval flow

The scope of the present clause is to describe how the EUDI Wallet can present ISO mDL selectively disclosed elements over the ISO mDL device retrieval flow, and how eIDAS2 trust services can be used to support this process.

NOTE: The ISO mDL device retrieval flow is mandatory for the EUDI Wallet according to the ARF [i.59].

The ISO mDL device retrieval flow is described in ISO/IEC 18013-5 [i.140], clauses 6.3.2, 6.3.2.1 (as flow 1) and 6.3.2.4. This clause will not repeat the entire ISO mDL device retrieval process, although a brief summary is provided below for readability with references to the ISO/IEC 18013-5 [i.140].

The ISO mDL device retrieval flow is illustrated in Figure 19.

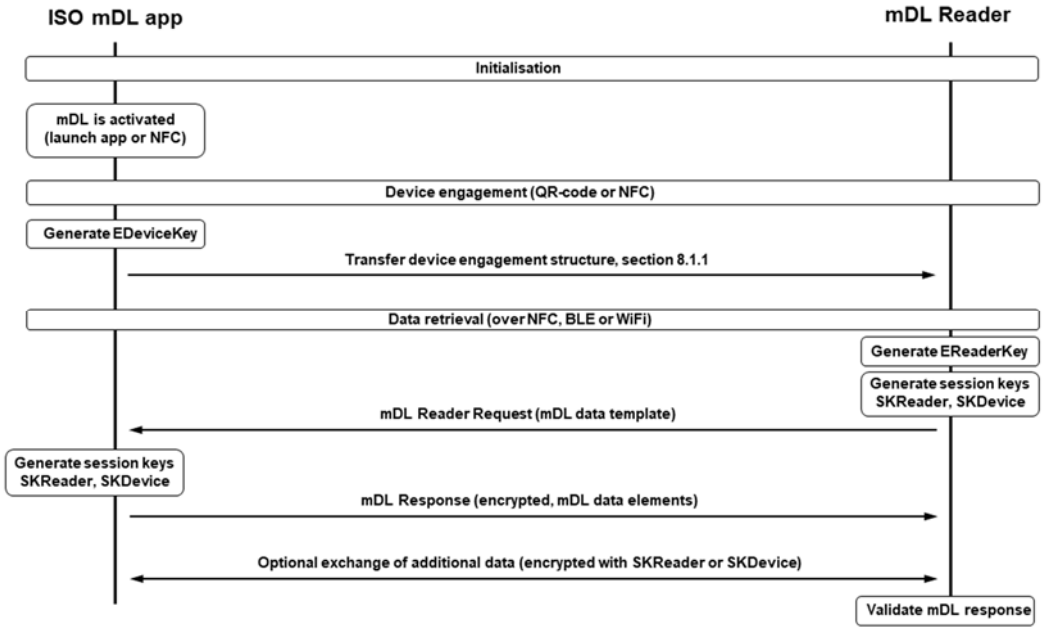

Figure 19: Overview of the ISO mDL device retrieval flow

On a high level, the ISO mDL device retrieval flow can be divided in the following phases, where the ISO mDL reader is equivalent to an attended eIDAS2 relying party:

- Initialization phase, whereby the ISO mDL app is activated either by the user or triggered by NFC contact with the ISO mDL reader (see ISO/IEC 18013-5 [i.140], clause 6.3.2.2 for more information).
- Device engagement phase, whereby the ephemeral device key EDeviceKey is generated, and the device engagement structure is transferred over NFC or as QR-code. The device engagement structure contains parameters for device retrieval transfer options TransferMethod and TransferOptions (see ISO/IEC 18013-5 [i.140], clauses 6.3.2.3, 9.1.1, 8.2.1, 8.2.2 and 8.2.1.1 for more information).
- Data retrieval phase, whereby the EReaderKey, SKReader and SKDevice keys are generated to establish an encryption session. The ISO mDL reader then transmits the mDL Reader Request and the ISO mDL replies with the mDL Response (see ISO/IEC 18013-5 [i.140], clauses 9.1, 9.1.1, 8.3.2.1.2 and 8.3.2.2.2 for more information).

As regards to selective disclosure, the mDL Reader Request contains a list of the DataElements the mDL Reader requests from the mDL app. Upon the user's consent, the mDL app will reply with the mDL Response with the selected DataElements in the DeviceSignedItems. The DeviceSignedItems object is signed by the mDL Authentication Key, to which the user is authenticated with a PIN-code or biometrics (see ISO/IEC 18013-5 [i.140], clauses 8.3.2.1.2 and 8.3.2.2.2 for more information).

The selected DataElements will be hashed at the mDL reader, and be compared with the corresponding hash values in the MSO. ISO/IEC 18013-5 [i.140], clause 9.1.2.3 describes how the relying party validates the MSO signature and how to check that the hashed mDL mdoc elements match the hash values in the MSO.

More specifically, ISO/IEC 18013-5 [i.140], clause 9.1.2.3 specifies in detail how the mDL reader validates the certificate chain of the IACA trust anchor and the Issuing Authority's MSO signer certificate. ISO/IEC 18013-5 [i.140], Annex C describes the ISO mDL VICAL, which points to the IACA trust anchor and revocation information.

7.2.3.2 Analysis of the ISO mDL device retrieval flow applied to eIDAS2

An analysis of the ISO mDL device retrieval flow applied to an eIDAS2 context results in the following observations and recommendations:

- The ISO mDL app should be part of an EUDI Wallet.
- The ISO mDL Issuing Authority corresponds to a QTSP, PIDP and/or an EUDI Wallet provider.
- The mDL Reader corresponds to an device retrieval eIDAS2 relying party (that will validate the ISO mDL as an (Q)EAA/PID).
- The recommendations should be observed in clause 7.2.1 on how a QTSP/PIDP supervised under eIDAS2 can operate as an ISO mDL IACA.
- The recommendations should be observed in clause 7.2.1 on how an eIDAS2 EU TL should be formatted to be compatible as an ISO mDL VICAL or vice versa.
- The eIDAS2 relying party should use the eIDAS2 EU TL (which is equivalent to an ISO mDL VICAL) to retrieve the QTSP/PIDP trust anchor (which is equivalent to the IACA trust anchor).
- The eIDAS2 relying party should validate the MSO (submitted by the ISO mDL app in the mDL Response) according to the principles in ISO/IEC 18013-5 [i.140], clause 9.1.2.3, by using the QTSP/PIDP trust anchor.
- The MSOs in the EUDI Wallet ISO mDL app should be unique as described in clause 7.2.1 to cater for verifier unlinkability when validated by the relying party.

NOTE 1: ISO mDL MSO does not enable unlinkability; it only enables selective disclosure.

NOTE 2: While issuer unlinkability is impossible to achieve, verifier unlinkability can be achieved by having the QTSP/PIDP issue batches of MSOs, each with unique salts, signatures, and DeviceKey elements. This will require an operational procedure of issuing multiple MSOs to each device on a regular basis, which may result in an additional operational cost for the QTSP/PIDP. Operational costs may be lessened by relying on a HDK function as described in clause 4.4.4.2 whereby the issuer only needs to keep track of a single DeviceKey element and use it to derive unique per MSO DeviceKey elements that the user can derive the corresponding private key for.

- The MSO is signed by the QTSP/PIDP with a COSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.188] and for QSC for future use [i.119].

These observations and recommendations should be considered with respect to selective disclosure for the ETSI work items ETSI TS 119 462 [i.79], ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81].

7.2.4 EUDI Wallet used with ISO mDL server retrieval flow

7.2.4.1 Overview of the ISO mDL server retrieval flows

The scope of the present clause is to describe how the EUDI Wallet can present ISO mDL selectively disclosed elements over the ISO mDL server retrieval flow, and how eIDAS2 trust services can be used to support this process.

NOTE: This ISO mDL server retrieval flow is NOT mentioned by the ARF, but may need to be used by national or specific implementations that need to be interoperable with ISO mDL.

The ISO mDL server retrieval flow can be initialized as a hybrid device/server process (see clause 7.2.4.2) or as a server process (see clause 7.2.4.3). Once the ISO mDL server retrieval flow has been initialized, it continues with either the WebAPI (see clause 7.2.4.5) or the OpenID Connect (OIDC) flow (see clause 7.2.4.7). Clause 7.2.4 will not repeat the entire ISO mDL server retrieval process, although a brief summary is provided below for readability with references to the ISO/IEC 18013-5 [i.140] standard.

7.2.4.2 ISO mDL flow initialization

The initialization of the ISO mDL device and server retrieval flows are described in ISO/IEC 18013-5 [i.140], clauses 6.3.2, 6.3.2.1 (as flow 2) and 6.3.2.4.

The ISO mDL device/server data retrieval flow is illustrated in Figure 20.

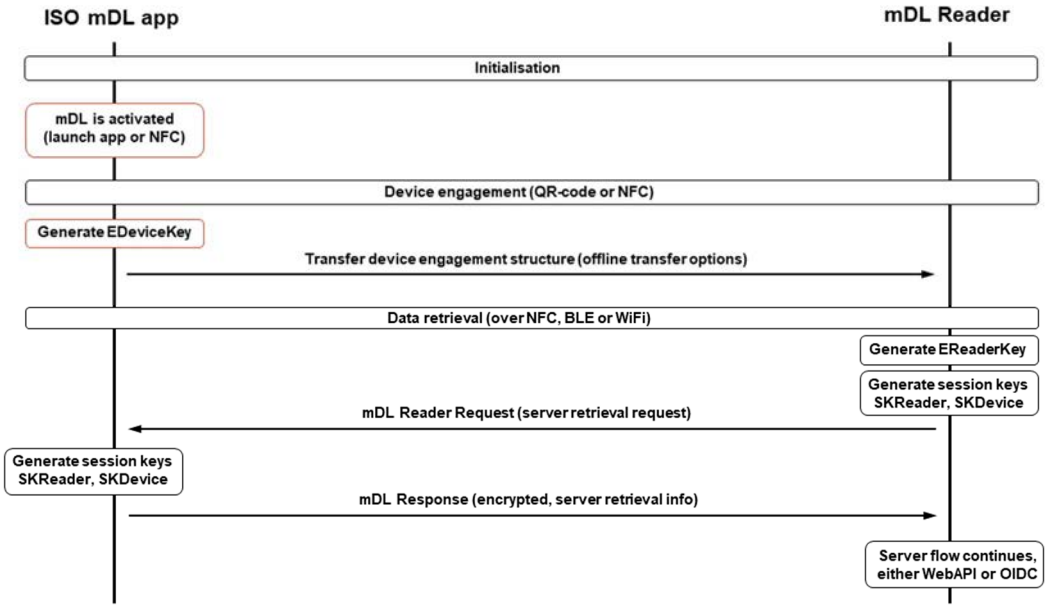

Figure 20: ISO mDL flow initialization

On a high level, the ISO mDL device/server retrieval flow can be divided in the following phases (where the ISO mDL reader is equivalent to an eIDAS2 relying party):

- Initialization phase, whereby the ISO mDL app is activated either by the user or triggered by NFC contact with the ISO mDL reader (see ISO/IEC 18013-5 [i.140], clause 6.3.2.2 for more information).
- Device engagement phase, whereby the ephemeral device key EDeviceKey is generated, and the device engagement structure is transferred over NFC or as QR-code (see ISO/IEC 18013-5 [i.140], clauses 6.3.2.3, 9.1.1, 8.2.1 and 8.2.2 for more information).
- Data retrieval phase, whereby the EReaderKey, SKReader and SKDevice keys are generated to establish an encryption session. The ISO mDL reader then transmits the mDL Reader Request including the server retrieval request and the ISO mDL replies with the mDL Response including the server retrieval information (see ISO/IEC 18013-5 [i.140], clauses 9.1, 9.1.1, 8.3.2.1.2.1 and 8.3.2.1.2.2 for more information).

The ISO mDL online data retrieval flow continues with either the WebAPI (see clause 7.2.4.5) or OIDC (see clause 7.2.4.7).

7.2.4.3 ISO mDL server retrieval flow initialization

The ISO mDL server retrieval flow initialization is described in ISO/IEC 18013-5 [i.140], clauses 6.3.2 and 6.3.2.1 (as flow 3) and 6.3.2.4.

The ISO mDL server retrieval flow initialization is illustrated in Figure 21.

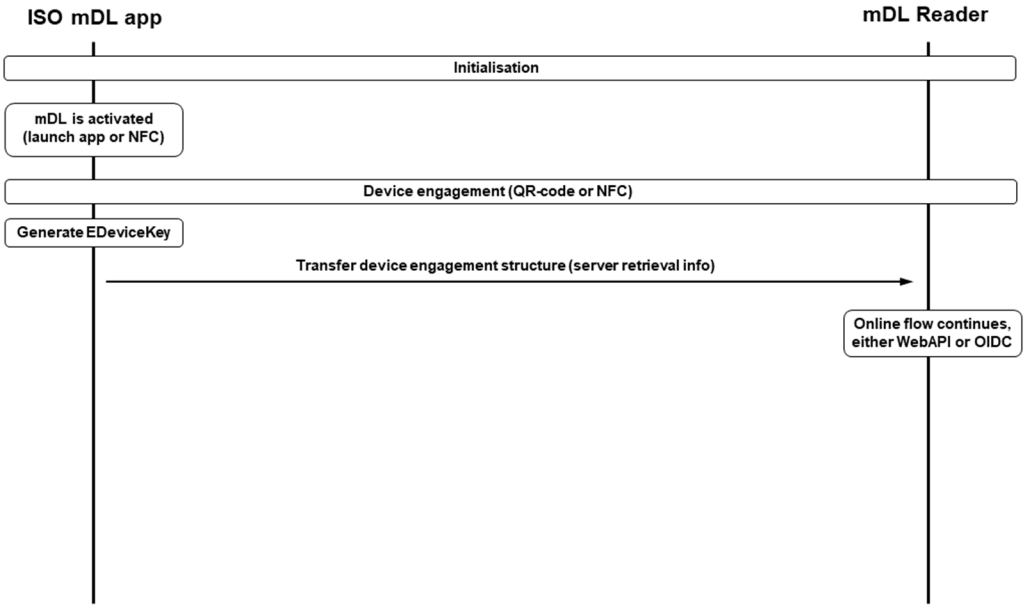

Figure 21: ISO mDL server retrieval flow initialization

On a high level, the ISO mDL server retrieval flow can be divided in the following phases (where the ISO mDL reader is equivalent to an eIDAS2 relying party):

- Initialization phase, whereby the ISO mDL app is activated either by the user or triggered by NFC contact with the ISO mDL reader (see ISO/IEC 18013-5 [i.140], clause 6.3.2.2 for more information).
- Device engagement phase, whereby the ephemeral device key EDeviceKey is generated, and the device engagement structure is transferred over NFC or as QR-code. The device engagement structure contains parameters for online transfer options WebAPI or OIDC (see ISO/IEC 18013-5 [i.140], clauses 6.3.2.3, 9.1.1, 8.2.1, 8.2.2 and 8.2.1.1 for more information).

The ISO mDL server retrieval flow continues with either the WebAPI (see clause 7.2.4.5) or OIDC (see clause 7.2.4.7).

7.2.4.4 ISO mDL server retrieval WebAPI flow

The ISO mDL server retrieval flow is described in ISO/IEC 18013-5 [i.140], clause 8.3.2.2 and the WebAPI calls are specified in ISO/IEC 18013-5 [i.140], clause 8.3.2.2.2.

The ISO mDL WebAPI server retrieval flow is illustrated in Figure 22.

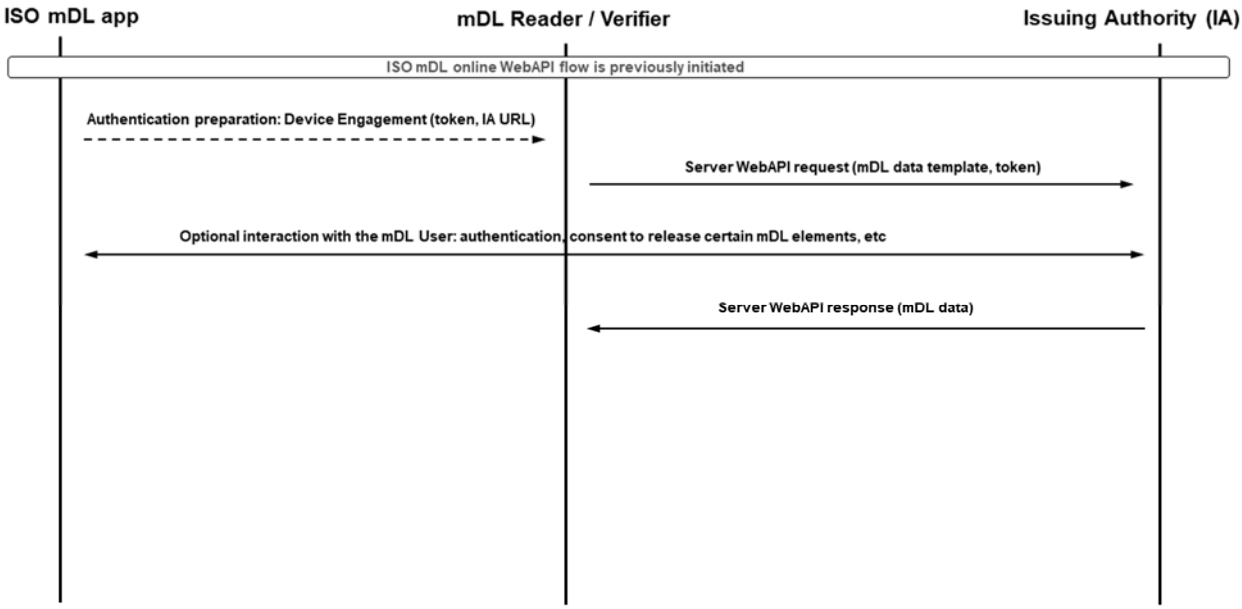

Figure 22: ISO mDL server retrieval WebAPI flow

As regards to selective disclosure, the mDL Reader submits a server retrieval WebAPI Request with a list of requested DataElements to the Issuing Authority. Upon the user's consent, the Issuing Authority will reply with the mDL Response with the selected and disclosed DataElements (see ISO/IEC 18013-5 [i.140], clause 8.3.2.2.2 for more information).

7.2.4.5 Analysis of the ISO mDL server retrieval WebAPI flow applied to eIDAS2

An analysis of the ISO mDL WebAPI server retrieval flow applied to an eIDAS2 context results in the following observations and recommendations:

- The ISO mDL app should be part of an EUDI Wallet.
- The ISO mDL Issuing Authority corresponds to a QTSP, PIDP and/or an EUDI Wallet provider.
- The mDL Reader corresponds to an eIDAS2 relying party, which will connect to the ISO mDL Issuing Authority over the WebAPI to request information about the user.

NOTE 1: eIDAS2 [i.86] Article 5a.14 states: "The provider of the European Digital Identity Wallet shall neither collect information about the use of the European Digital Identity Wallet which is not necessary for the provision of European Digital Identity Wallet services, nor combine person identification data or any other personal data stored or relating to the use of the European Digital Identity Wallet with personal data from any other services offered by that provider or from third-party services which are not necessary for the provision of European Digital Identity Wallet services, unless the user has expressly requested otherwise." If the ISO mDL Issuing Authority also has the role as an eIDAS2 European Digital Identity Wallet provider, the statement in eIDAS2 article 5a.14 may require additional privacy considerations when the server retrieval is used.

NOTE 2: eIDAS2 [i.86] Article 5a.16 states: "The technical framework of the European Digital Identity Wallet shall: (a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorized by the user". If the ISO mDL Issuing Authority also has the role as an eIDAS2 QTSP/PIDP, the statement in eIDAS2 article 5a.16(a) may imply that server retrieval is not possible unless explicitly approved by the user.

- The ISO mDL Issuing Authority may deploy QWACs in order to prove its authenticity over TLS to the connecting relying parties.
- The WebAPI token is a JWT that is signed by the ISO mDL Issuing Authority OIDC Authorization Server. The JWT signer certificate should be issued by an IACA, which in the eIDAS2 context is also a QTSP.

- The ISO mDL Reader, which is an eIDAS2 relying party, should use the ISO mDL VICAL (EU TL) to retrieve the IACA trust anchor (QTSP trust anchor).
- The WebAPI JWT is signed by the QTSP/PIDP with a JOSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.188] and for QSC for future use [i.119].

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.79], ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81].

7.2.4.6 ISO mDL server retrieval OIDC flow

The ISO mDL server retrieval flow is described in ISO/IEC 18013-5 [i.140], clause 8.3.2.2 and the OIDC calls are specified in ISO/IEC 18013-5 [i.140], clause 8.3.3.2.2.

The ISO mDL OIDC server retrieval flow is illustrated in Figure 23.

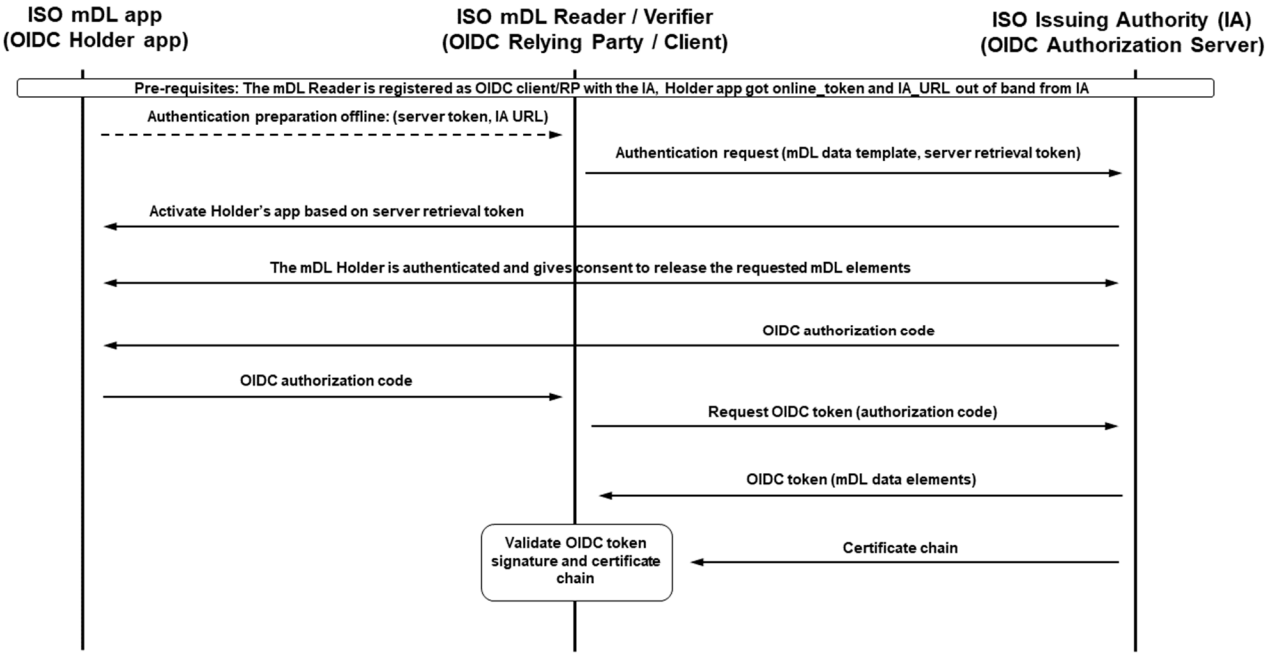

Figure 23: ISO mDL server retrieval OIDC flow

As regards to selective disclosure, the mDL Reader (OIDC client) submits an server retrieval OIDC Request with the requested data elements (JWT claims) to the Issuing Authority, which operates an OIDC Authorization Server. This activates the OIDC authorization code flow [i.170]. Based on the user's consent, the Issuing Authority (OIDC Authorization Server) will reply to the mDL Reader (OIDC client) with the OIDC Token with the selected and disclosed JWT claims about the user (see ISO/IEC 18013-5 [i.140], clause 8.3.3.2.2 and Annex D.4.2.2 for more information about the OIDC workflow).

7.2.4.7 Analysis of the ISO mDL OIDC server retrieval flow applied to eIDAS2

An analysis of the ISO mDL OIDC server retrieval flow applied to an eIDAS2 context results in the following observations and recommendations:

- The ISO mDL app should be part of an EUDI Wallet.
- The ISO mDL Issuing Authority corresponds to a QTSP, PIDP and/or an EUDI Wallet provider.
- The ISO mDL Issuing Authority operates an OIDC Authorization Server, which supports the OIDC authorization code flow.
- The mDL Reader corresponds to an eIDAS2 relying party, which is registered as an OIDC client to the ISO mDL Issuing Authority OIDC Authorization Server. The mDL Reader will connect to the ISO mDL Issuing Authority over OIDC to request information about the user.

NOTE 1: eIDAS2 [i.86] Article 5a.14 states: "The provider of the European Digital Identity Wallet shall neither collect information about the use of the European Digital Identity Wallet which is not necessary for the provision of European Digital Identity Wallet services, nor combine person identification data or any other personal data stored or relating to the use of the European Digital Identity Wallet with personal data from any other services offered by that provider or from third-party services which are not necessary for the provision of European Digital Identity Wallet services, unless the user has expressly requested otherwise." If the ISO mDL Issuing Authority also has the role as an eIDAS2 European Digital Identity Wallet provider, the statement in eIDAS2 article 5a.14 may require additional privacy considerations when the server retrieval is used.

NOTE 2: eIDAS2 [i.86] Article 5a.16 states: "The technical framework of the European Digital Identity Wallet shall: (a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorized by the user". If the ISO mDL Issuing Authority also has the role as an eIDAS2 QTSP/PIDP, the statement in eIDAS2 article 5a.16(a) may imply that server retrieval is not possible unless explicitly approved by the user.

- The ISO mDL Issuing Authority may deploy QWACs in order to prove its authenticity over TLS to the connecting relying parties.
- The OIDC Token is a JWT that is signed by the ISO mDL Issuing Authority OIDC Authorization Server. The JWT signer certificate should be issued by an IACA, which in the eIDAS2 context is also a QTSP.
- The ISO mDL Reader, which is an eIDAS2 relying party, should use the ISO mDL VICAL (EU TL) to retrieve the IACA trust anchor (QTSP trust anchor).
- The OIDC token JWT is signed by the QTSP/PIDP with a JOSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.188] and for QSC for future use [i.119].

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.79], ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81].

7.2.5 EUDI Wallets used with ISO/IEC 18013-7 for unattended flow

7.2.5.1 Overview of the ISO/IEC 18013-7 flows

ISO/IEC CD 18013-7 [i.141] draft standard extends ISO/IEC 18013-5 [i.140] with the unattended flow, i.e. the server retrieval flow whereby an ISO mDL app connects directly to an mDL reader that is hosted as a web server application. ISO/IEC CD 18013-7 [i.141] is backward compatible with the protocols in ISO/IEC 18013-5 [i.140].

NOTE: Since the ISO mDL app connects directly to the web hosted mDL reader without involving any issuer, this flow preserves the user's privacy as required in eIDAS2 [i.86], Article 5a.16.

ISO/IEC CD 18013-7 [i.141] unattended flow is designed based on the following protocols:

- Device Retrieval from an ISO mDL app to a web server application over HTTPS POST; this flow is described in clause 7.2.5.2.
- OpenID for Verifiable Presentations (OID4VP) [i.171] in conjunction with Self-issued OpenID Provider v2 (SIOP2) [i.173]; this flow is described in clause 7.2.5.3.

7.2.5.2 ISO/IEC 18013-7 Device Retrieval flow

The general data retrieval architecture is described in ISO/IEC 18013-5 [i.140], clause 6.3.2.4. ISO/IEC CD 18013-7 [i.141] draft standard describes device retrieval of data for unattended (i.e. online web application) use cases. The ISO mDL app and the ISO mDL reader support device retrieval using the mDL request and response as specified in ISO/IEC 18013-5 [i.140], clause 8.3.2.1.

ISO/IEC CD 18013-7 [i.141] adds Annex A that specifies the Reader Engagement phase, which takes place before the Device Engagement phase in ISO/IEC 18013-5 [i.140]. The Reader Engagement struct contains the parameter RetrievalOptions, which in turn includes the RestApiOptions that defines the URI and REST API parameters for the HTTPS connection to the web hosted mDL Reader.

ISO/IEC CD 18013-7 [i.141] unattended online retrieval flow is illustrated in Figure 24.

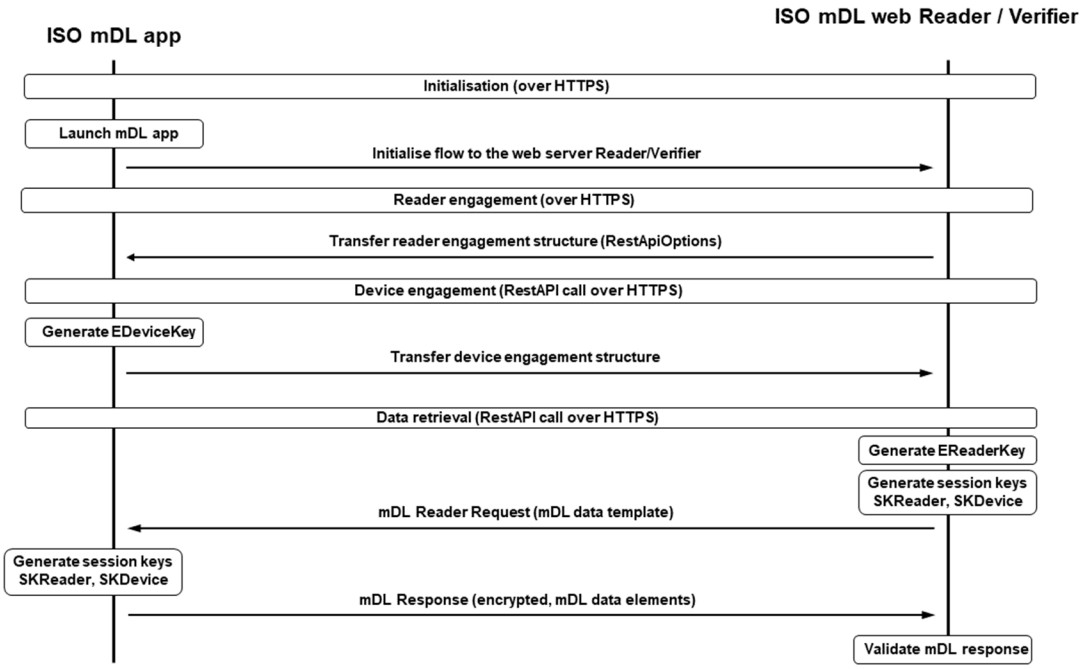

Figure 24: ISO mDL unattended Device Retrieval flow

When the mDL Response has been retrieved and parsed by the ISO mDL reader/verifier, the mDL selected attributes and MSO are verified according to the same process as the ISO mDL device retrieval flow (clause 7.2.3).

As regards to selective disclosure for the ISO mDL unattended Device Retrieval flow, the same principles and recommendations apply as for the ISO mDL device retrieval flow (clause 7.2.3). However, the ISO/IEC 18013-7 specification [i.141] is not referred to by the ARF [i.59], although the associated specification ISO/IEC 23220-4 [i.146] is mentioned in the ARF.

7.2.5.3 ISO/IEC 18013-7 OID4VP/SIOP2 flow

As an alternative to the unattended Device Retrieval flow, ISO/IEC CD 18013-7 [i.141] specifies an unattended (online) flow based on OID4VP [i.171] with SIOP2 [i.173]. The OID4VP/SIOP2 flow is defined in Annex B of ISO/IEC CD 18013-7 [i.141]. Furthermore, the OID4VP/SIOP2 protocol is based on the ISO/IEC CD 23220-4 [i.146] profile for presentations of ISO mDL.

ISO/IEC CD 18013-7 [i.141] unattended OID4VP/SIOP2 flow is illustrated in Figure 25.

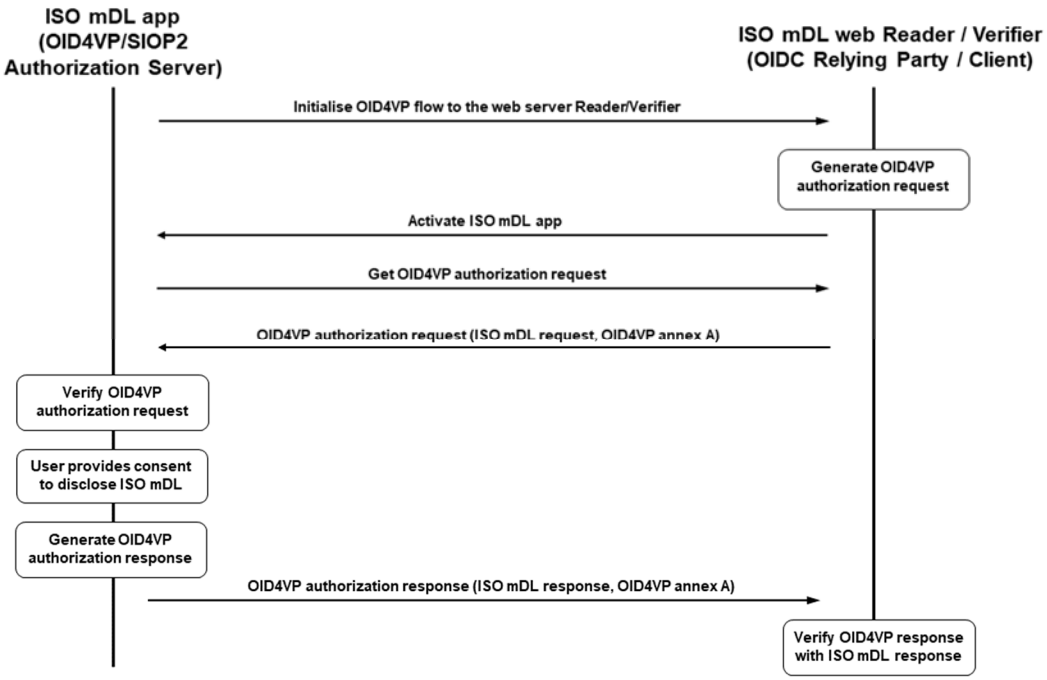

Figure 25: ISO mDL unattended OID4VP/SIOP2 flow

When the OID4VP Response, which contains the mDL Response, has been retrieved and parsed by the ISO mDL reader/verifier, the mDL selected attributes and MSO are verified according to the same process as the ISO mDL device retrieval flow (clause 7.2.3).

As regards to selective disclosure for the ISO mDL unattended OID4VP/SIOP2 flow, the same principles and recommendations apply as for the ISO mDL device retrieval flow (clause 7.2.3). However, the ISO/IEC CD 18013-7 [i.141] specification is not referred to by the ARF [i.59], although the associated specification ISO/IEC CD 23220-4 [i.146] is mentioned in the ARF.

NOTE: ISO/IEC CD 23220-4 [i.146] is mentioned as a target in the ARF [i.59], but not mandatory since not yet published. If ISO/IEC CD 23220-4 [i.146] will include ISO/IEC 18013-5 [i.140] proximity as well as OID4VCI and OID4VP then 23220-4 is likely to be mandatory in a future version of the ARF.

7.3 Implications for SD-JWT selective disclosure

7.3.1 Background to W3C VCDM and SD-JWT

The ARF 1.0 text mandates the joint utilization of W3C VCDM v1.1 and SD-JWT. The former is used to express the data model and provide the overall structure of the attestation, whereas the latter is proposed as a selective disclosure capable proof mechanism.

NOTE 1: The SD-JWT specification works as a standalone attestation format too as it was not designed to provide selective disclosure capability for W3C VCs specifically but for JWTs in general. The IETF SD-JWT-based Verifiable Credentials (SD-JWT VC) draft specification [i.112] details data formats as well as validation and processing rules for expressing attestations as JSON payloads with selective disclosure capability as detailed in the SD-JWT specification.

NOTE 2: At the time of writing, the status of W3C VCDM v1.1 in the ARF is being revised. Proposals exist that would rely on SD-JWT VC for the PID and (Q)EAAs and on a mapping algorithm to ensure VCDM 1.1 compliance. However, discussing proposals is outside the scope of the present document, which focuses only on selective disclosure.

To understand the implications of SD-JWT for selective disclosure, especially in relation to the W3C VCDM, it is important to first understand what W3C VCDM is and how different proof mechanisms relate to it. After providing such a primer, the reader will be better able to understand the motivation behind the recommendations and the specific ways SD-JWT is presented herein.

7.3.2 A primer on W3C VCDM

7.3.2.1 Overview of W3C Verifiable Credential Data Model (VCDM)

The W3C Verifiable Credential Data Model (VCDM) is a way to express verifiable electronic attestation of attributes on the Web. At its core, a W3C Verifiable Credentials (VC) is a standardized digital format for presenting and exchanging verifiable claims (in essence statements expressed using subject-property-value relationships) about individuals, organizations, or things. These claims can be expressed as attributes in an electronic attestation of attributes. Specifically designed for the Web, the W3C VCDM aims to enable users to present attribute assertions from potentially different issuers and about potentially different identity subjects. These assertions can be organized into information graphs expressing subject-property-value relationships (e.g. Credential-type-DrivingLicense).

The W3C Verifiable Credentials Data Model (VCDM) is an open standard and is designed to be interoperable across different systems and platforms and to support a wide range of applications. The W3C VCDM v1.1 [i.209] describes a issuer-holder-verifier based model for digital "verifiable credentials" (defined as a "set of one or more claims made by an issuer" that are also "tamper-evident [with] authorship that can be cryptographically verified"). Specifically, the VCDM v1.1 aims to improve the ease of expressing digital credentials while also ensuring a high degree of privacy.

EXAMPLE: A trusted authority, such as a PID Provider, could construct a W3C VCDM compliant attestation containing the PID attributes and sign these with their private key. The user (assumed herein to be the identity subject of the VC) can then create a Verifiable Presentation (VP) using one or more VCs and present attributes to a verifier. The resulting W3C VC is verifiable to any verifier who has access to the required cryptographic keys. The proof mechanism could then support privacy features such as selective disclosure and/or unlinkable verifiable presentations.

The VCDM 1.1 text mandates that claims about a subject can be made tamper evident, that these claims are expressed in the form of subject-property-value relationships, and that it is possible to organize these claims into an information graph. However, it is not required that the claims or the proof is expressed as a graph in the attestation. To date, the VCDM 1.1 text has principally focused on JSON-LD type attestations. W3C VCDM Support for JSON only has been limited. The lack of JSON only support is problematic since the ARF prohibits the use of linked data proofs for the PID and only optionally supports JSON-LD. The ARF 1.3text mandates that the PID is issued as a JWT and that it is secured using SD-JWT.

After the publication of VCDM v1.1, the W3C VC WG has been working on VCDM 2.0 to make the standard more flexible and able to support multiple formats and signature algorithms. Work was ongoing to support the representation of verifiable claims in multiple ways including JSON, JSON-LD, or using any other data representation syntax capable of expressing the data model such as XML, YAML, or CBOR, as long as there is a mapping defined back to the base data model defined in the VCDM document (which relies on JSON-LD). This work was ongoing as several outstanding issues remained unsolved.

However, recently the W3C VC WG has argued strongly in favour of removing securing JSON and non linked data formats from the specification (see W3C VC WG issue #88 [i.205]). This means that the W3C VCDM is likely to evolve in a direction that will not address outstanding issues with the underspecified JSON sections, which includes key details such as how to do the required transformations or mappings. By extension, it is likely also that the proposed W3C work on how to secure a (W3C) VC using JSON [i.135] will be postponed until further notice. It is worth noting that the W3C VC WG charter does not specify specific media types, but that there does not exist a consensus with the WG to pursue JSON.

Regardless of the debate outcome, each VC and VP includes fields for specifying the signature schemes used to sign the claim or the presentation of a claim respectively (i.e. whether the verification of the proof is calculated against the data transmitted or against a transformation such as another data model or an information graph). Since the debate outcome is presently unknown, the text herein describes the solutions presently mentioned by VCDM v1.1, which are JSON Web Token and Data Integrity Proofs. Each will be described, with illustrations for possible solutions to still outstanding issues for the JWT based approach. The data integrity proofs will only be briefly explained to help readers understand why some of the ideological differences may make it difficult to secure a W3C VC using SD-JWT without a proper specification on how to secure a W3C VC using JSON.

Finally, the potential of relying only on SD-JWT VC for the attestation and use case specific mapping to VCDM 1.1 will be discussed as it represents the most suitable selective disclosure alternative considering the ongoing debates.

7.3.2.2 W3C VC, JSON-LD, data integrity proofs, and linked data signatures

There are many concepts surrounding the W3C VCDM v1.1, including JSON-LD, data integrity proofs, and linked signatures. The first, JSON-LD, will be explained in detail below, but it is helpful to explain how the other two relate to JSON-LD.

Data integrity proofs are defined by the W3C as "a set of attributes that represent a digital proof and the parameters required to verify it." Put differently, a data integrity proof provides information about the proof mechanism, parameters required to verify that proof, and the proof value itself. This information is provided using Linked Data vocabularies in a JSON-LD formatted attestation.

Linked data signatures are a proposed way to sign data expressed in linked data formats such as a JSON-LD. Linked data signatures sign the underlying information graph as opposed to the payload itself. More specifically, the graph is normalized into a byte stream that is signed. The corresponding verification can be of the graph of information, and not necessarily the syntax specific content itself meaning that the same digital signature would validate information expressed in multiple compatible syntaxes without necessitating syntax specific proofs (see W3C VC Data Integrity v1.0 where this idea is explored in detail).

To understand how a W3C VCDM v1.1 compliant attestation would look like, it is necessary to understand its core format, JSON-LD. Being similar to JSON, a key difference is that JSON-LD uses a property called `"@context"` to link attributes to descriptions that provide semantic clarity on how to unambiguously interpret each attribute. Each attribute is expressed in the form of subject-predicate-object triples that essentially describe an information graph.

Consider the following example of a JSON-LD document describing a person. The attributes `name` and `jobTitle` are mapped to concepts in the schema.org vocabulary as detailed in the `"@context"`.

```
{
  "@context": "http://schema.org/",
  "@id": "https://me.example.com",
  "@type": "Person",
  "name": "John Doe",
  "jobTitle": "ETSI TR editor"
}
```

The @context allows the JSON-LD to be mapped to an Resource Description Framework (RDF) model and thus an information graph. The information graph for the above looks as follows:

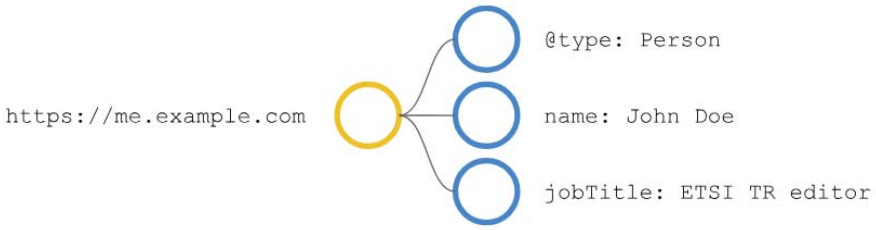

Figure 26: Example of W3C VCDM v1.1 graph

And the W3C VCDM v1.1 graph triples are as follows:

Table 2: Example of W3C VCDM v1.1 graph triples

Subject	Predicate	Object
`https://me.example.com`	`http://www.w3.org/1999/02/22-rdf-syntax-ns#type`	`http://schema.org/Person`
`https://me.example.com`	`http://schema.org/jobTitle`	`ETSI TR editor`
`https://me.example.com`	`http://schema.org/name`	`John/Jane Doe`

And the associated N-Quads (a syntax for RDF datasets) are:

1) <https://me.example.com> <http://schema.org/jobTitle> "ETSI TR editor" .

2) <https://me.example.com> <http://schema.org/name> "John/Jane Doe".

3) <https://me.example.com> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/Person>.

The benefit with the above is that it does not matter what syntax is used to describe the underlying information graph as they would all describe the same model and thus enable a mapping to the exact same N-Quads.

NOTE: Since data integrity proofs sign the N-Quads containing triples as opposed to only the object, they do not fully support predicates that rely on the algebraic manipulations of the object. For instance, while it is possible to check for message equality, it is not possible to check whether one value is larger than another. Consequently, the signature scheme used to sign the N-Quads may support additional predicates than the N-Quads allow (e.g. a range proof may be supported by the signature scheme but the N-Quad may limit the predicate to an equality test).

To enable selective disclosure of a W3C VCDM v1.1 using data integrity proofs and linked data proofs, an issuer would need a proof mechanism that can logically order the N-Quads in such a way that the verifier knows that the presented attributes are properly paired. One way is to use the N-Quad message digests as leaf nodes to a Merkle tree and include the Merkle root in the attestation. Another, assuming that the issuer is comfortable with using JSON-LD and linked data proofs only, is to include N-Quad messages as selectively disclosable values in a SD-JWT `"_sd"` array (see clause 7.3.1.2 for a detailed description of how to generate a disclosure in [i.123] (IETF OAUTH: "Selective Disclosure for JWTs (SD-JWT)") and let the user present only the parts of the information graph that the verifier needs.

To date, the most well developed solution relies on the `bbs-2022` cryptosuite, which supports JSON-LD + data integrity proofs + linked data proof. Including triples in SD-JWT is not entirely straight forward and would require additional specification.

To conclude, JSON-LD is a way to express linked data and JSON-LD based attestations may include data integrity proofs that also rely on linked data for their verification. When also using linked data proofs, issuers can issue (Q)EAAs that are highly optimized for semantic interoperability. However, it is not entirely clear how selective disclosure and predicates would work in the context of PID/(Q)EAAs. Supporting crypto suites like `bbs-2022` are based on primitives that the public sector is unlikely to use since they are not considered as being plausible quantum safe. Solutions like SD-JWT can support linked data proofs but it is not entirely clear how they could be combined with data integrity proofs (and what the benefits would be) as SD-JWT was designed with JWT based attestations in mind.

Having described how W3C VCDM v1.1 compliant attestations can be secured using SD-JWT also for JSON-LD and linked data signatures, attention now turns to JWT based W3C VCs and SD-JWT.

7.3.2.3 JWT based W3C VC

One popular proof format that is actively used in several implementations is the JSON Web Token (IETF RFC 7519 [i.132]). A JWT encodes claims as a JSON object contained in a JSON Web Signature (JWS) (IETF RFC 7515 [i.130]) or JWE (IETF RFC 7516 [i.131]). A user could present a VP with the VC claims using JWT as described in example 32 of the W3C VC Data Model [i.209]. The decoded JWT contains the presentation as exemplified next.

```
{
   ...,
   "verifiableCredential":[
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRpZDpleGFtcGxlOmFiZmUxM2...QGbg"
   ]
}
```

```
The VC contained within (highlighted above in yellow) contains the following information about the
identity subject.
{
   ...,
   "credentialSubject":{
      "degree":{
         "type":"BachelorDegree",
         "name":"<span lang='fr-CA'>Baccalauréat en musiques numériques</span>"
      }
   }
}
```

The VC contains the attribute in cleartext. Typically, a signed JWT containing identity data cannot support use cases where the JWT is issued once and then presented multiple times by the user who seeks to disclose only the attributes necessary for the service. In and of itself, the W3C VC standard only supports, but does not enforce, selective disclosure by design. The standard is flexible and supports multiple selective disclosure techniques. However, until recently these selective disclosure techniques have relied on multi-message signature schemes like `bbs-2022` suite.

NOTE: The text below assumes that there is a way to secure JSON for W3C VCDM v1.1 and ignores the ongoing debate on the topic within the W3C VC WG.

7.3.2.4 SD-JWT based attestations

To support selective disclosure in JWTs, Fett, Yasuda, and Campbell (2023) specify Selective Disclosure JSON Web Token (SD-JWT) in the Internet Engineering Task Force (IETF) draft document [i.123] entitled "Selective Disclosure for JWTs (SD-JWT)". At its core, an SD-JWT is a digitally signed JSON document that can contain salted attribute hashes that the user can selectively disclose using disclosures that are outside the SD-JWT document. This allows the user to share only those PID attributes that are strictly necessary for a particular service.

NOTE 1: SD-JWT is generally applicable to selective disclosure of JWTs that are not bound to the W3C VCDM v1.1. A W3C VCDM v1.1 contains sections that describe how a VC can be JSON encoded in a JWT and then protected using JWS/JWE. Correspondingly, the SD-JWT specifies how any JWT can support selective disclosure. But the joint utilization of the two is not straightforward.

NOTE 2: An SD-JWT supports selective disclosure solutions that require a clear logical ordering of data. It does not support algebraic manipulations of data.

Each SD-JWT contains a header, payload, and signature. The header contains metadata about the token including the type and the signing algorithm used. The signature is generated using the PID Provider's private key. The payload includes the proof object that enables the selective disclosure of attributes. Each disclosure contains a salt, a cleartext claim name, and a cleartext claim value. The issuer then computes the hash digest of each disclosure and includes each digest in the attestation it signs and issues.

Using the proof object and the user shared disclosures, the verifier can verify that the disclosed claims were part of the original attestation. To do so, the verifier first verifies the issuer's signature over the entire SD-JWT. The verifier then calculates the digest over the shared disclosures and checks that the digest is included in the signed SD-JWT. Since the SD-JWT includes only digests of disclosable attributes, the verifier can only learn about claim names and claim values that are disclosed by the user or that are included as clear-text claims. The verifier cannot learn about any other claim names or values as these are included in the SD-JWT as salted attribute digests.

The IETF SD-JWT draft specification 07 [i.123] of 2023-12-11 details the exact process of creating a disclosure in section 5.2. In essence, for each disclosable claim, the issuer generates and associates a random salt with each key value pair, and encodes the byte representation of these as base64url. An example of a disclosure is shown in Figure 27.

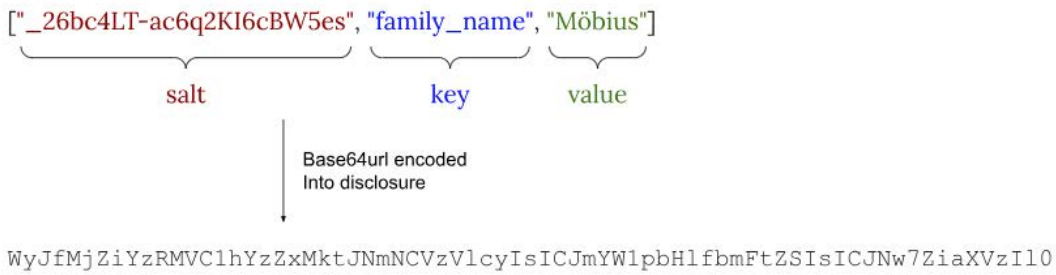

Figure 27: Example of SD-JWT disclosure

Figure 27 illustrates an example with the byte representation of the JSON-encoded array containing the salt, key, and value, is base64url-encoded into the disclosure.

NOTE: A linked data signature could be included in the `_sd` array but it is not entirely clear how to handle triples in the disclosure. One option could be to set the subject to the sub property in the attestation and to only include predicates in the disclosures as: `[<salt>, <predicate>, <object>]`

To embed a disclosure in the SD-JWT, the issuer hashes each disclosure using a specified hash algorithm. The base64url encoded bytes of the digest, and not the disclosure, is then included in the SD-JWT as an array in the claim `_sd`, which includes only an array of strings, each being the digest of a disclosure or a random number (used to hide the original number of disclosures). This array is randomized so that the order of attribute disclosures is not always the same.

The SD-JWT specification supports selectively disclosable claims in both flat and more complex nested data structures. The issuer can therefore decide for each key individually, on each level of the JSON, whether or not the key should be selectively disclosable. The `_sd` claim is included in the SD-JWT at the same level as the original claim. Selectively disclosable claims can in turn include other objects with selectively disclosable claims.

Below, this text only exemplifies the flat and the nested data structure examples, but others are possible too.

Table 3: Example of SD-JWT using a flat data structure

Contents	`["imQfGj1_M0El76kdvf7Daw", "address", {"street_address": "Schulstr. 12", "locality": "Schulpforta", "region": "Sachsen-Anhalt", "country": "DE"}]`
Disclosure	`WyJpbVFmR2oxX00wRWw3NmtkdmY3RGF3IiwgImFkZHJlc3MiLCB7InN0cmVldF9hZGRyZXNzIjogIlNjaHVsc3RyLiAxMiIsICJsb2NhbGl0eSI6ICJTY2h1bHBmb3J0YSIsICJyZWdpb24iOiAiU2FjaHNlbi1BbmhhbHQiLCAiY291bnRyeSI6ICJERSJ9XQ`
Digest	`FphFFpj1vtr0rpYK-14fickGKMg3zf1fIpJXxTK8PAE`
_sd value	`{ "_sd": [ "FphFFpj1vtr0rpYK-14fickGKMg3zf1fIpJXxTK8PAE" ], ..., "_sd_alg": "sha-256" }`

Table 4: Example of nested SD-JWT with the sub-claim country in cleartext

Contents	`["QSNIhu_n6a1rI8_2eNARCQ", "street_address", "Schulstr. 12"], ["QPkblxTnbSLL94I2fZIbHA", "locality", "Schulpforta"], ["jR-Yed08AEo4gcogpT5_UA", "region", "Sachsen-Anhalt"]`
Disclosures	`WyJRU05JaHVfbjZhMXJJOF8yZU5BUkNRIiwgInN0cmVldF9hZGRyZXNzIiwgIlNjaHVsc3RyLiAxMiJd, WyJRUGtibHhUbmJTTEw5NEkyZlpJYkhBIiwgImxvY2FsaXR5IiwgIlNjaHVscGZvcnRhIl0, WyJqUi1ZZWQwOEFFbzRnY29ncFQ1X1VBIiwgInJlZ2lvbiIsICJTYWNoc2VuLUFuaGFsdCJd`
Digests	`"G_FeM1D-U3tDJcHB7pwTNEElLal9FE9PUs0klHgeM1c", "KlG6HEM6XWbymEJDfyDY4klJkQQ9iTuNG0LQXnE9mQ0", "ffPGyxFBnNA1r60g2f796Hqq3dBGtaOogpnIBgRGdyY"`
_sd value	`{ "address": { "_sd": [ "G_FeM1D-U3tDJcHB7pwTNEElLal9FE9PUs0klHgeM1c", "KlG6HEM6XWbymEJDfyDY4klJkQQ9iTuNG0LQXnE9mQ0", "ffPGyxFBnNA1r60g2f796Hqq3dBGtaOogpnIBgRGdyY" ], "country": "DE" }, ..., "_sd_alg": "sha-256" }`

The QTSP/PIDP will have to send the raw claim values contained in the SD-JWT, together with the salts, to the EUDI Wallet user. The SD-JWT standard requires that data format for sending the SD-JWT and the disclosures to the EUDI Wallet user is a series of base64url-encoded values in what is called the Combined Format for Issuance, which looks like follows: `<SD-JWT>~<Disclosure 1>~<Disclosure 2>~...~<Disclosure n>~<optional Holder Binding JWT>`. Note the separation of between the values using ~. The specific ways the ~ character should be used is defined under section 5 in the SD-JWT v.07 specification.

When the EUDI Wallet user receives the attestation from the QTSP/PIDP, the SD-JWT standard requires that the user verifies the disclosures. The user does so by extracting the disclosures and the SD-JWT from the Combined Format for Issuance, hashing each disclosure, and accepts the SD-JWT only if each resulting digest exists in the `_sd` array.

Relatedly, during presentation, the user sends the SD-JWT and the *n* disclosures to the verifier as a series of base64url encoded values in what is called the Combined Format for Presentation, which looks as follows: `<SD-JWT>~<Disclosure 1>~<Disclosure 2>~...~<Disclosure n>~<optional Holder Binding JWT>`

The verifier checks that the issuer's signature is valid over the SD-JWT, that the disclosure digests are part of the SD-JWT, and if applicable that the Holder binding is valid (for specific steps see section 8 in the SD-JWT 07 specification [i.123]).

Having described JSON secured W3C VCs and how SD-JWT can ensure selective disclosure of JWT based attestations, the text next discusses the potential joint utilization of both W3C VCs and SD-JWT, and why it is not as straightforward as it may appear.

7.3.2.5 Securing the W3C VC payload using SD-JWT

It is very difficult to clearly communicate options on how to secure a W3C VC using SD-JWT. There are two main ways a W3C VC can be secured and there is no agreement on whether or not to secure JSON within the W3C VC WG. Also, SD-JWT was designed with JWT based attestations in mind and not to cater specifically to the needs of W3C VCDM. As such, the text herein is speculative.

The focus of SD-JWT v.07 [i.123] is to specify how claims in a JWT can be selectively disclosed. This applies to any type of attestation where attribute assertions are JSON encoded in a JWT, including potential JWT versions of any W3C VCDM v1.1 compliant attestation (assuming that future work in the W3C VC WG will also secure JSON).

The April 11 specification of SD-JWT (v.04) includes an appendix that exemplifies how to use the SD-JWT specification to secure a payload represented as a W3C VC data model. Relatedly, the W3C VCDM recommendation contains examples of W3C VCs encoded as JWT. However, the two examples build on different assumptions. One way to jointly utilize W3C VCDM v1.1 and SD-JWT is to include the entire W3C VC as a claim value in the SD-JWT. Another way is to rely on a transformation algorithm that would allow a verifier to recreate the W3C VC from an SD-JWT that uses JSON only. Both have their associated challenges.

The SD-JWT specification does support selective disclosure of a W3C VCDM v1.1 compliant attestation either as an embedded value, e.g. as `"vc": {<W3C VC>}`, or using a transformation algorithm (for an example using VCDM 2.0 see clause 9.1 of [i.209]). Similarly, it is possible to rely on proposals similar to the W3C Securing Verifiable Credentials using JSON Web Tokens [i.206] and use SD-JWT to secure it.

Relatedly, the VCDM v1.1 introduces one way to design VCs that could be jointly utilized with SD-JWT. The VCDM v1.1. uses a JWT to secure a VC payload that needs to follow the rules for a JSON-LD payload. Consequently, the JWT is an envelope, which means that it is not compatible with more recent drafts of SD-JWT and SD-JWT VC. There is also confusion on how to include JWT claims in the credential payload. Furthermore, the presentation is another JWT, where the VC is embedded. Such a design is not without problems.

NOTE 1: Until recently, the VCDM 2.0 included proposals that would address limitations in the VCDM v1.1. These proposals in the VCDM 2.0. would require only that the VC can be mapped into a JSON-LD representation (can be one directional). Consequently, a VC can be just a JWT secured using SD-JWT that relies on a pure JSON payload. The way presentations are created is also up to the respective presentation. However, the ongoing disagreement around the continued support for this work (see W3C VC WG issue #88 [i.205]) means that it is no longer clear that the W3C VCDM 2.0 will support JSON. And since W3C VCDM v1.1 requires additional work to fully work with SD-JWT, the way to secure an W3C VCDM v1.1 compliant attestation using SD-JWT is unclear.

NOTE 2: The SD-JWT specification published on April 11 2023 is developed around the assumption that the VCDM 2.0. would secure JSON too. Relatedly, the ARF 1.3 text mandates VCDM v1.1. compliance with the assumption that there would be a way to rely on pure JSON payloads (the ARF until version 1.3 has always mentioned JSON-LD as optional and only intended for (Q)EAAs).

NOTE 3: There are security concerns associated with polyglot parsing (i.e. allowing some verifiers to parse an attestation formatted as JSON-LD as JSON only and others to parse the same attestation as JSON-LD) so the ARF text mandates JSON only without assuming linked data for the PID. Due to these concerns, this document recommends relying on a mapping algorithm to achieve VCDM 1.1 compliance and to rely on SD-JWT VC for the attestation format itself. This mapping is facilitated further with the recent introduction of SD-JWT VC DM [i.92] that adds schemas and vocabularies as well as JAdES support on top of SD-JWT VC.

To exemplify possible joint utilizations, the following VCDM v1.1. compliant attestation will first be populated with some of the mandatory PID attributes. The example will utilize an external proof since data integrity proofs are of questionable use in the PID context (which means that it does not require a `proof` property). The content is shortened for brevity and only includes values relevant for selective disclosure.

```
{
  "@context":[
  <>
  ],
  "id":"http://example.com/credentials/4643",
  "type":[
    "VerifiableCredential",
    "IdentityCredential"
  ],
  "issuer":"https://example.com/issuers/14",
  "issuanceDate":"2010-01-01T19:23:24Z",
  "credentialSubject":{
    "givenName":"Jane",
    "familyName":"Doe",
    "birthDate":"2000-01-01"
  }
}
```

Figure 28: Example of W3C VC with some ARF 1.3 mandatory PID attributes

The example in Figure 28 shows a W3C VC Data Model 1.1 compliant attestation with some of the mandatory PID attributes as mentioned in ARF 1.3. The identity data are highlighted in blue.

To secure the above attestation with a JWT and enable selective disclosure, it is necessary to create a disclosure of each mandatory attribute claim in the `credentialSubject` property, and then to create a valid JWT. This may appear to be straightforward, but the issuer needs to decide:

1) whether or not to use linked data proofs (which the ARF 1.3 text prohibits for PID attestations); and

2) whether or not to use the SD-JWT as a container (which using Figure 28 would suggest that the attestation can be parsed as JSON-LD, which the ARF does not mandate and can introduce security concerns); or

3) rely on a transformation algorithm.

To discuss every possible option is outside the scope of this text; only a single option is shown for illustrative purposes.

One possible way is to put the credential payload `"vc"` claim to differentiate it from the claims in the JWT that is used as the security envelope. Furthermore, because JWT uses different property names, some implementations duplicate the `iss`, `jti`, and `iat` claim names while others rely on the mapping proposed in the JSON encoding section in the W3C VCDM v1.1 recommendation. Below, the example uses the duplicate claim names because this is how the examples are provided in the W3C VCDM v1.1 recommendation (duplicate claim names are optional in IETF RFC 7519 [i.132]). Note the omission of the `sub` claim due to it being selectively disclosable. Finally, the proof is omitted in Figure 29.

```
{
  "vc":{
    "@context":[
      "..."
    ],
    "id":"http://example.com/credentials/4643",
    "type":[
      "VerifiableCredential",
      "IdentityCredential"
    ],
    "issuer":"https://example.com/issuers/14",
    "issuanceDate":"2010-01-01T00:00:00Z",
    "credentialSubject":{
      "_sd":[
        "2cj...szs",
        "H03...iVY",
        "S7e...uDc"
      ]
    },
    "_sd_alg":"sha-256"
  },
  "iss":"https://example.com/issuers/14",
  "jti":"http://example.com/credentials/4643",
  "iat":"1262304000"
}
```

Figure 29: Example of how SD-JWT could secure a W3C VC

The example in Figure 29 shows a possible way an SD-JWT could secure an W3C VC Data Model 1.1 compliant attestation containing the mandatory PID attributes as disclosure digests (highlighted in blue). Conflicts that exist between the W3C VC Data Model 1.1 and SD-JWT were resolved by adhering to the W3C standard. Appendix A.4. from the SD-JWT specification draft 4 was used as the basis for this example. Note that this approach is not compliant with SD-JWT VC. Note also that it is not entirely clear what selective disclosure actually means in this approach (in linked data selective disclosure means revealing a subtree of the information graph, which is not what SD-JWT does).

A Verifiable Presentation for the above VC looks as follows:

```
{
  "iss": "some key identifier",
  "aud": "did:example:4a57546973436f6f6c4a4a57573",
  "nbf": 1541493724,
  "iat": 1541493724,
  "exp": 1573029723,
  "nonce": "343s$FSFDa-",
  "vp":{
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "type": [
      "VerifiablePresentation"
    ],
    "verifiableCredential": [
      "..."
    ]
  }
}
```

Figure 30: Example of a VP for an SD-JWT secured W3C VC

The example in Figure 30 shows a Verifiable Presentation for an SD-JWT secured W3C Verifiable Credentials Data Model v1.1 [i.209] compliant attestation. The SD-JWT is a base64url encoded string.

There are some difficulties with using an SD-JWT (the IETF SD-JWT draft specification v0.4 [i.123]) and the W3C Verifiable Credentials Data Model v1.1 [i.209] as illustrated in the examples above. Some important difficulties are:

- A lack of a syntax definition catering for the selective disclosure capability in SD-JWT. Put differently, it is possible to include the digests of the disclosures and decoys in the VC but it is not entirely clear how this would harmonise with linked data principles. It is also unclear how the presentation of selectively disclosable attributes will look like. The W3C VCDM was designed with presentation capabilities in mind where attributes from multiple VCs, about potentially different subjects, could be combined into a single presentation. In contrast, the SD-JWT combined presentation format is focused on ease of use and ease of deployment.

- Selectively disclosable claims are base64url encoded twice (once in the SD-JWT and once again in the VC. This double encoding adds inefficiencies.

- There exists confusion in how to use preexisting JWT claims, e.g. `sub`, in the credential payload. Some follow the duplicate claims approach (which is prevalent in the examples in the W3C VCDM v1.1 text). Others rely on the provided JSON encoding rules and the W3C VCDM v1.1 implementer's guidelines recommendations.

- JSON-LD was not designed to extend into the SD-JWT. The interaction between the JSON-LD context and the disclosures protected in the SD-JWT is undefined until after the claims have been decoded from the SD-JWT (assuming the context defines the disclosable attributes and not the selective disclosure array).

One may try different versions of their joint utilization to circumvent some of the four above mentioned problems. But perhaps most importantly, the above example may trigger questions as to the benefits derived from combining JSON-LD with SD-JWT. The former was developed to ensure semantic interoperability in an open data world. And SD-JWT was designed to provide selective disclosure capabilities to a JSON based attestation in a JWT. Using JSON-LD for W3C VC together with data integrity proofs provide benefits in a usage scenario where the actors face semantic interoperability challenges that can be resolved by accessing other related information about a particular thing. Seemingly, jointly utilizing JSON-LD based W3C VCs with SD-JWT does not result in their combined benefits, but rather that their respective benefits are not utilized.

One alternative is to simply use SD-JWT VC also for W3C VCDM v1.1 attestations, and rely on transformation algorithms to re-create the W3C VC.

7.3.2.6 Using SD-JWT VC only

The IETF SD-JWT VC draft specification [i.112] provides a format that is optimized for the transport of the credential including the disclosures without further encoding. It is not designed to be embedded into any envelopes. It is arguably better to simply rely on JSON only claims for SD-JWT VC and recreate the W3C VCDM using a mapping algorithm. This option does not require the issuer to use linked data proofs (the ARF 1.3 text does not allow the use of linked data proofs for the PID attestation), includes identity subject claims in an SD-JWT VC, and where a transformation is used to map the SD-JWT VC claims to a W3C VCDM 1.1 compliant information graph. Relying on SD-JWT VC and mapping would circumvent the aforementioned four difficulties and also adhere strictly to the design logic of a particular solution approach.

An example is provided next.

```
{
  "alg": "ES256",
  "typ": vc+sd-jwt,
  <other header info>
}
.
{
  "iss":"https://example.com/issuers/14",
  "nbf": 1262304000,
  "iat": 1262304000,
  "vct": "eu.europa.ec.eudiw.pid.se.1",
  "_sd":[
    "2cj...szs",
    "H03...iVY",
    "RKE...omY",
    "S7e...uDc"
   ],
   "_sd_alg":"sha-256"
}
```

Figure 31: Example of a SD-JWT VC where W3C VCDM compliance relies on mapping

The example in Figure 31 shows an SD-JWT VC secured attestation (not using JSON-LD) with the mandatory and disclosable PID attributes highlighted in blue. The `"_sd"` is here included as a root claim. This SD-JWT VC can be consumed, without prior processing, by any compliant SD-JWT VC library. Further evaluation can be done using standard JWT payload processing algorithms. In the example in Figure 31.

- The JOSE header indicates the type.
- The claims in the credential are standard JWT claims. Applications can use predefined and established JWT claims from the "JWT Claims Registry", like `"sub"` for user identifiers. They can also use more complex claim structures such as those defined by OpenID Connect for Identity Assurance for providing information about provenance and level of assurance. This means existing JWT-based implementations can consume such VC payloads directly.
- The vct communicates to the verifier how to interpret any disclosed claim and there is no need for a separate `@context`.

A presentation is constructed using the combined format for presentation as defined in the SD-JWT specification.

NOTE 1: The present document recommends using the IETF October 23 2023 version of SD-JWT without Appendix A4 and A5 to understand the selective disclosure mechanism. Relatedly, to understand how to use SD-JWT VC as an attestation format, see the 2023-10-23 version of "SD-JWT-based Verifiable Credentials (SD-JWT VC)" [i.122].

NOTE 2: It should also be observed that SD-JWT VC is referenced by the OpenID4VC High Assurance Interoperability Profile (HAIP) [i.172], which is a profile of OpenID for Verifiable Credentials.

7.3.2.7 SD-JWT and multi-show unlinkable disclosures

Because every SD-JWT disclosure contains a unique salt, this unique salt acts as an identifier for the entire SD-JWT. Put differently, it is enough for a malicious issuer to receive a single disclosure from a colluding verifier for the issuer to uniquely identify the identity subject. Similarly, colluding verifiers could compare salt values to link together presentations from the same user (see clause 9.4 in the SD-JWT [i.123] specification for additional details).

While it is impossible to prevent issuers from identifying the user based on the unique salt in the salted attribute hashes approach, it is possible to enable multi-show verifier unlinkable disclosures even if verifiers collude or if a single curious verifier attempts to learn more about the user than what is disclosed in each presentation. To achieve complete multi-show unlinkability it is required that:

1) each SD-JWT VC contains only unique salts (even for the same claim); and

2) each SD-JWT VC is associated with a unique cryptographic key material used for device binding and/or holder binding (denoted as "holder binding key" in the context of SD-JWT).

Consequently, issuers are required to rely on batch issuance of SD-JWT to the EUDI Wallet if device retrieval functionality is desired (in an online scenario, the user can request a new SD-JWT on demand).

NOTE: To reduce the burden on issuers, it is possible to introduce a limit on the number of uses of each SD-JWT. The user's SD-JWTs would then be linkable in a portion of their presentations.

EXAMPLE: A user is given 10 PID attestations as SD-JWT VCs. The user presents the first 9 SD-JWT VCs once and the 10^{th} twice. Out of the 11 presentations, two are linkable.

7.3.2.8 Predicates in SD-JWT

Similar to MSO, an SD-JWT was not designed to support predicates that can be dynamically computed (e.g. to compute an age over proof from the birth date). Here too, the recommendation is to use static claims with Boolean values such as `"age_over_NN": "True"`. However, as presented above in clause 4.4.7, it is possible to rely on issuer signed computational inputs and parameters to enable dynamic predicate support in SD-JWT.

7.3.3 Analysis of using SD-JWT as (Q)EAA format applied to eIDAS2

An analysis of the W3C VC and IETF SD-JWT formats applied to an eIDAS2 context results in the following observations and recommendations:

- The W3C VC Data Model v1.1 in conjunction with IETF SD-JWT should be supported by an EUDI Wallet according to the ARF [i.59]. However, this is problematic given the difficulties detailed in clause 7.3.2. Consequently, the recommendation of the present document is to use SD-JWT VC and to rely on transformation algorithms if issuers want to achieve W3 VCDM v1.1 compliance. Note that these mappings are use case specific.

- It is possible (but currently not specified how) to jointly utilize JSON-LD and linked data proofs with SD-JWT, but data integrity proofs remain an open question.

- The present document recommends using SD-JWT VC as a standalone attestation format where selective disclosure is required. When verifier unlinkability is required, it is possible to rely on a batch issuance approach where each SD-JWT VC contains unique salts. Each attestation in a batch should also contain a unique public key that the user needs for the holder binding JWT. Clause 4.4.4.2 describes the possibility to use Hierarchical Deterministic Key derivation functions where the SD-JWT VC issuer can issue a batch of SD-JWT VCs, each with a unique and unlinkable public key value derived from a single user controlled public key.

- Another option to achieve unlinkability afforded by HAIP is for the user to request specific claims they need to present to a verifier and for the issuer to issue only these claims in the attestation; an approach that fits particularly well with the logic of short lived attestations.

- The SD-JWT VC issuer corresponds to a QTSP and/or a PIDP.

- The SD-JWT VC verifier corresponds to an eIDAS2 relying party (that will validate the SD-JWT as a (Q)EAA/PID).

- The eIDAS2 relying party should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor.

- The eIDAS2 relying party should validate the attestation (submitted by the EUDI Wallet) according to the principles described in clause 7.3.2; the issuer's signature should be validated by using the QTSP/PIDP trust anchor.

- The SD-JWT VCs in the EUDI Wallet should all use unique salts as described in clause 7.3.2 to cater for verifier unlinkability when validated by the relying party.

NOTE 1: Hence, the QTSP/PIDP would need to issue batchwise SD-JWT VCs in order to cater for multi-show verifier unlinkability. Batch issuance will require an operational procedure of issuing multiple SD-JWT VCs to each device on a regular basis, which may result in an additional operational cost for the QTSP/PIDP. Clause 4.4.4.2 describes an approach where the issuer can derive multiple unique user controlled public keys on the basis of a single user controlled public key.

NOTE 2: SD-JWT does not satisfy the requirements of full unlinkability.

- The SD-JWT VC is signed by the QTSP/PIDP with a JOSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.188] and for QSC for future use [i.119].

- The SD-JWT VC may be signed with an ETSI JAdES signature if supported by the relying party. Thus, the JAdES signature format may contain additional information about revocation information, CA-chains and time-stamps.

These observations and recommendations should be considered with respect to selective disclosure for the ETSI work items ETSI TS 119 462 [i.79], ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81], where also a mapping algorithm for the PID could be proposed.

7.4 Feasibility of BBS+ applied to eIDAS2

7.4.1 General

This clause provides an analysis of the feasibility of BBS+ applied to eIDAS2. The BBS+ scheme is of interest since it caters for issuer and verifier unlinkability, which could support privacy for a user's EUDI Wallet that shares selectively disclosed attributes. The following aspects are in scope of the analysis:

- The standardization status of BBS+, and if the schemes can be considered for the eIDAS2 regulation.

- Whether or not a standardized version of BBS+ can be applied to the W3C Verifiable Credentials Data Model (VCDM).

- Post-quantum aspects of BBS+.

- Conclusions of how BBS+ may applied to QTSPs/PIDPs and EUDI Wallets operating under eIDAS2.

7.4.2 Standardization of BBS+

In order for BBS+ to be considered for the EUDI Wallet, it would have to be standardized by CEN, ETSI or ISO as declared in the EU regulation 1025/2012 [i.88].

As described in clause 4.3.5.1, a set of anonymous digital signatures schemes are specified in the ISO/IEC 20008 series [i.143]. More specifically, ISO/IEC 20008-2 [i.143] mechanism 3 specifies the cryptographic primitives of a qSDH scheme, which corresponds to BBS04 with single messages [i.25]. BBS04 with single messages is however not practically sufficient for most attestation formats, including the W3C Verifiable Credentials Data Model and SD-JWT VC, which require BBS+ with multi messages.

BBS+, which supports multi messages, is however not yet fully standardized. IETF CFRG is currently in the process of specifying BBS+ in the IETF CFRG BBS standard [i.116], whilst DIF is drafting a specification for blind signatures extension of BBS+ [i.66], but even when the IETF and DIF standards are finalized they will not have the status such that they can be referenced by the eIDAS2 regulation.

In order to bridge this gap, ISO/IEC has initiated the Preliminary Work Item (PWI) 24843 [i.144] on privacy-preserving attribute-based credentials. One objective of ISO/IEC PWI 24843 is to formally standardize the multi-message signature scheme version of ISO/IEC 20008-2 [i.143], i.e. BBS+.

ISO/IEC are also working on the common draft ISO/IEC CD 27565 "Guidelines on privacy preservation based on zero knowledge proofs" [i.150]. More specifically, Annex C of ISO/IEC CD 27565 includes an example of selective disclosure by using BBS+, with a reference to the IETF CFRG BBS draft specification.

Hence, the ISO/IEC PWI 24843 proposal, possibly in conjunction with ISO/IEC CD 27565, has the potential to result in an ISO standardized version of BBS+ as well as other multi-message signature schemes. If these ISO standards on BBS+ will materialize, they may be referred by the eIDAS2 regulation and its implementing acts. When such standards become available, the various attestation formats can also detail how BBS+ can be used as a proof mechanisms.

7.4.3 Feasibility of using BBS+ with W3C VCDM

The analysis in clause 5.3.2.2 concludes that if ISO/IEC PWI 24843 and/or ISO/IEC CD 27565 will standardize BBS+ according to IETF CFRG BBS, then W3C BBS Cryptosuite v2023 [i.212] can be enhanced to reference such an ISO standard. In such a scenario, the W3C Verifiable Credential Data Integrity 1.0 specification [i.208] would refer to an ISO compliant version of W3C BBS Cryptosuite v2023. That would in turn mean that the W3C Verifiable Credentials Data Model v2.0, in conjunction with W3C Verifiable Credential Data Integrity 1.0, would be underpinned with an ISO standardized version BBS+.

It should however be observed that the ARF [i.59] requires the JSON PID to be compliant with the W3C Verifiable Credentials Data Model v1.1 with JWT encoding. Since an ISO standardized version of BBS+ would require W3C Verifiable Credentials Data Model v2.0 [i.210] with JSON-LD encoding, it will not be compatible with the ARF.

NOTE: It is not entirely clear what the ARF text requires in terms of W3C VCDM compliance. Section 6.2.2, Table 3 in the ARF text requires that the *presentation* of an attestation is compliant with W3C VCDM 1.1, which means that the presentation includes verifiable statements about subject-predicate-value triplets that can be modelled as a graph. Section 7.5.3 requires that the *issuance* is compliant with the W3C VCDM 1.1. However, section 7.5.3 also requires that attestations are JWT based (optional support only for JSON-LD) and secured using SD-JWT. It is not clear how this compliance is to be achieved, i.e. whether enveloping and/or mapping is intended, and how enveloping would work with selective disclosure. The present report recommends using SD-JWT VC and relying on a mapping approach to ensure VCDM 1.1 compliance. If SD-JWT VCs are used, it is not clear how BBS+ can secure such attestations.

Hence, in order to support an ISO standardized version of BBS+, it is recommended to update the ARF to allow for W3C Verifiable Credentials Data Model v2.0 or preferably specify such format in the forthcoming ETSI TS 119 472-1 [i.81] standard on (Q)EAAs profiles.

7.4.4 Post-quantum considerations for BBS+

As discussed in clause 4.3.1.5, and as further elaborated on in clause 9, BBS+ multi-message signatures and disclosures that are generated in a pre-quantum world will remain confidential in a post-quantum world. Put differently, a computationally unbounded attacker will not be able to reveal neither undisclosed messages nor the hidden signature value.

In a post-quantum world, however, BBS+ cannot maintain data integrity and authenticity. An attacker with a quantum computer can reveal the signer's private key from the public key and forge new signatures and proofs. Clause 9 discusses the prerequisites of this attack, its potential impact, and how to protect against it in greater detail.

7.4.5 Conclusions of using BBS+ applied to eIDAS2

An analysis of the BBS+ scheme applied to an eIDAS2 context results in the following observations and recommendations:

- The BBS+ algorithm would need to be standardized according to ISO/IEC PWI 24843 in order to comply with the EU regulation 1025/2012 on standardization.
- A standardized profile of W3C BBS Cryptosuite v2023 would need to reference the ISO standardized version of BBS+. It is recommended that ETSI TC ESI standardize such a profile.
- A standardized (Q)EAA/PID profile of W3C Verifiable Credentials Data Model (VCDM) v2.0 in conjunction with W3C Verifiable Credential Data Integrity (VCDI) 1.0 would need to be specified, and reference the standardized W3C BBS Cryptosuite v2023. It is recommended that ETSI TC ESI standardizes profiles if attestation formats are to be W3C VCDM compliant and secured using BBS+.
- The issuing QTSPs/PIDPs would need to implement such ETSI standards in order to issue (Q)EAAs/PIDs compliant to W3C VCDM v2.0 and signed with the BBS+ algorithm.

- The BBS+ signature verifier corresponds to an eIDAS2 relying party (that will validate the BBS+ multi message signatures generated by the (Q)EAA/PID).
- The eIDAS2 relying party should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor.
- The eIDAS2 relying party should validate the BBS+ multi message signature (finalized by the EUDI Wallet) according to the principles described in the IETF CFRG BBS specification (or the future ISO standard on BBS+); the issuer's signature should be validated by using the QTSP/PIDP trust anchor.

NOTE 1: The BBS+ algorithm would cater for full unlinkability.

- The EUDI Wallets need to support the BBS+ algorithm in cryptographic keys management systems as specified in clause 6.5.3 of the ARF [i.59]. As described in clause 7.6, such cryptographic keys management systems with support for BBS+ could preferably be remote HSMs (with BBS+ support) or SIM-cards with support for BBS_MAC/BBS+ (see clause 6.6.4).
- A long term (Q)EAA/PID based on BBS+ should be used in a pre-quantum world only. The QTSP/PIDP should plan for migrating to quantum-safe cryptograhic algorithms in a post-quantum world.

NOTE 2: As an option, the QTSP/PIDP may issue batches of one-time (Q)EAAs/PIDs based on BBS+. If such one-time (Q)EAA/PID is used for creating a BBS+ signature just once, this will cater for a blinded signature that is fully unlinkable, supports predicates and selective disclosure, and is post-quantum safe.

These observations and recommendations should be considered with respect to selective disclosure for the ETSI work items ETSI TS 119 462 [i.79], ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81].

7.5 Feasibility of programmable ZKPs applied to eIDAS2 (Q)EAAs

7.5.1 Background and existing solutions

As discussed in clause 6.5, there exist two implementations of ZKP schemes (zk-SNARKs) that are utilized for sharing selectively disclosed attributes and revocation status information.

The Cinderella project (see clause 6.5.2) has integrated support for zk-SNARKs in TLS software libraries, which allows for Cinderella pseudo-certificates with selected attributes and optional OCSP stapled responses to be communicated over the TLS handshake. More specifically, the Belgian, Estonian, and Spanish national eID smartcards with X.509 QCs have been successfully tested with the Cinderella TLS implementation. Hence, the existing eIDAS PKI infrastructure without modifications is re-used. Configuring or refreshing the Cinderella pseudo-certificates can take up to nine minutes, and should therefore be performed offline, but the online verification takes only 10 ms.

The zk-creds project (see clause 6.5.3) has implemented anonymous credentials by using ZKP of ICAO compliant eMRTDs (passports). The ZKP is essentially generated based on the eMRTD's Data Group 1, which contains the textual information available on the eMRTD's data page and the Machine Readable Zone: name, issuing state, date of birth, and passport expiry.

Hence, the Cinderella and zk-creds projects have demonstrated with their prototypes that ZKP schemes can be used with existing digital identity infrastructures to share selected attributes of X.509 certificates and ICAO eMRTDs.

7.5.2 Extensions to EUDI Wallets, relying parties and protocols

In order for an EUDI Wallet to use zk-SNARKs with existing credentials (such as X.509 certificates), a circuit compiler (such as the Geppetto compiler) is needed to integrate the zk-SNARK client circuits into the EUDI Wallet. Furthermore, the authentication protocol (such as TLS) needs to be enhanced in order to generate pseudo-certificates that can be validated by the relying party (TLS server). The EUDI Wallet would need to download the trusted roots based on the EU Trusted List (TL) in order to validate the status of the X.509 certificate and the optional OCSP-response.

The relying party needs to be extended in order to validate the pseudo-certificates and the proof of the OCSP response. The Cinderella project has demonstrated that this is feasible with TLS and X.509 certificates. In a similar fashion, the zk-creds project has demonstrated that it is possible to share selected attributes of an ICAO eMRTD by using ZKP schemes.

Since the ARF specifies ISO mDL mdoc and mandates W3C VCDM compliance for the PID formats, it would be of interest to investigate if the EUDI Wallet could be extended with zk-SNARK client circuits policy templates that can generate selected attributes of pseudo-versions of ISO mDL mdocs and/or W3C VCDM compliant VCs (e.g. SD-JWT VC with mapping) and optional stapled revocation information.

Furthermore, the ARF [i.59] specifies OID4VP [i.171] as the presentation protocol for the EUDI Wallet. Hence, it would be of interest to specify a profile of OID4VP with a DIF Presentation Definition (OID4VP request) [i.67] and DIF Presentation Submission (OID4VP response) [i.67] that could use programmable ZKP schemes to present selected attributes of pseudo-versions of ISO mDL mdocs and/or W3C VCDM compliant VCs and optional stapled revocation information.

Since zk-SNARKs can cater for full unlinkability, this feature would be inherited for the EUDI Wallets as well. Also, it is recommended to select zk-SNARKs that are plausible quantum computing safe (see Table A.4).

7.5.3 Conclusions of programmable ZKPs applied to eIDAS2 (Q)EAAs

An analysis of the ZKP scheme applied to (Q)EAAs, QCs or PIDs in an eIDAS2 context results in the following observations and recommendations:

- The EUDI Wallets would need to be extended with programmable ZKP circuits and policy templates in order to generate pseudo-credentials with selected attributes of (Q)EAAs, QCs or PIDs and optional stapled revocation information. The EUDI Wallet should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor. The zk-SNARK trusted roots would need to be configured as well.

- The issuing QTSPs/PIDPs can re-use the existing eIDAS framework and related ETSI standards in order to issue QCs. The eIDAS2 framework and planned ETSI standards for issuance of (Q)EAAs/PIDs can also be used without modifications. The QTSP/PIDP trust anchor can be published at an eIDAS2 EU TL.

- The verifier corresponds to an eIDAS2 relying party (that will validate zk-SNARK proofs and pseudo-credentials generated out of the (Q)EAA/QC/PID). The eIDAS2 relying parties would need to be extended with zk-SNARK circuits and policy templates in order to validate the pseudo-credentials and stapled revocation information.

NOTE: The zk-SNARK scheme would cater for full unlinkability.

- The zk-SNARKs that are plausible quantum computing safe (see Table A.4) should be used.

- OID4VP would need to be extended in order for an EUDI Wallet to present the pseudo-credentials with selected attributes and stapled revocation information to a relying party.

These observations and recommendations should be considered with respect to selective disclosure for the ETSI work items ETSI TS 119 462 [i.79], ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81]. Implementations of the programmable ZKP schemes in the EUDI Wallets and relying parties may be implemented and evaluated as part of the eIDAS2 large scale pilots.

7.6 Secure storage of PID/(Q)EAA keys in EUDI Wallet

The ISO mDL authentication key and SD-JWT holder binding keys should be protected in the device's Trusted Execution Environment (TEE) or a Secure Element (SE). The user should be able to access the ISO mDL authentication key and SD-JWT holder binding key by authentication with a PIN-code or the use of biometrics. There exist implementations and large scale deployments of ISO mDL for Apple iOS® and Google Android®, which utilize Secure Elements that protect the ISO mDL authentication key. Several ISO mDL and SD-JWT data elements are PII and should therefore be stored securely. Encryption at rest of the SD-JWT is recommended, and if possible the SE/TEE should be used to perform the encryption, with keys protected by the SE/TEE, or else the ISO mDL and SD-JWT should be stored in the SE/TEE. Alternatively, the ISO MSO or SD-JWT keys could be protected in a remote HSM or external device, which are the other cryptographic keys management systems as specified in clause 6.5.3 of the ARF [i.59]. The ARF [i.59], clause 6.5.3 and table 5 also specify how to store and access the PID/(Q)EAA cryptographic keys in a device used by the EUDI Wallet.

Since BBS+ is not (yet) selected to be used for any PID format, there is no specification in the ARF about storage or access to BBS+ credentials and keys. However, the research paper "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials" [i.13] describes how to efficiently implement a BBS_MAC/BBS+ variant on a SIM-card, which can be considered as an external cryptographic device that can be accessed by a mobile device. It is also plausible that HSMs in a near future will be equipped with the BBS+ algorithm, which would then cater for the EUDI Wallets to access BBS+ credentials and keys in a remote HSM. It is however unlikely that BBS+ will be implemented in embedded Secure Elements in the near future.

From a regulatory perspective, the eIDAS2 [i.86] article 5c specifies the legal requirements on an EUDI Wallet certification, which will be defined in a CIR (Commission Implementing Regulation). This CIR will in turn refer to ENISA's EUCC (EU Cybersecurity Certification scheme), which may regulate the certification requirements on protection of the PID/(Q)EAA as ISO mDL and SD-JWT.

Furthermore, CEN TC/224 WG17 may specify Common Criteria Protection Profiles (CC PP) on how to protect the PID/(Q)EAA and associate cryptographic keys related to the ENISA EU-CC; such EUDI Wallet CC PP may be based on TC/224 WG17 [i.48]. Also, TC/224 WG20 [i.49] are specifying how to onboard the PID to an EUDI Wallet, which involves the associated cryptographic key protection as well.

Other certification standards that may underpin the ENISA EU-CC scheme are Global Platform TEE Protection Profile [i.95] and Eurosmart PP-0117 Protection Profile for Secure Sub-System in System-on-Chip (3S in SoC) [i.89].

Additional recommendations on how to store and protect credentials and the associated cryptographic keys in a digital wallet are available in the DIF Wallet Security [i.68], ISO/IEC CD 23220-6 [i.147] and W3C Universal Wallet [i.207] specifications.

NOTE: Complete descriptions about storage of PID/(Q)EAA, protection of cryptographic keys and EUDI Wallet certifications go beyond the scope of the present document, but an overview is provided in the present clause since the cryptographic keys are of relevance to selective disclosure of PID/(Q)EAA in the formats of ISO mDL and SD-JWT.

8 Privacy aspects of revocation and validity checks

8.1 Introduction to revocation and validity checks

Given that eIDAS2 article 5a.16(a) as well as recitals 14, 15, and 59 require that selective disclosures and unlinkability are done in ways that prevent data linkability, then the data unlinkability requirement have to be extended to validity status checks. Herein, the focus includes only options that fall under "state of the art" (solutions that have been deployed on a market) as stipulated in GDPR articles 25, 26, and 32 [i.216], and those approaches that are "experimental" (solutions where technical feasibility has been demonstrated but where market deployments are still lacking). In addition to this, eIDAS2 article 5a.16 should be considered, where it is stated:

"*The technical framework of the European Digital Identity Wallet shall: (a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user;*"

Hence, revocation services and validity status check services should avoid collecting revocation information about the EUDI Wallet and its (Q)EAAs.

Furthermore, a validity status check (e.g. due to revocation) can be conceptualized as a set (non-)membership proof, and alternatives that limit correlation handles and uncertainty reduction are discussed. For completeness, the text also mentions well known options that may not be suitable as a validity status check approach.

NOTE 1: Both (Q)EAAs or PIDs may be considered with respect to revocation and validity status checks; only the term (Q)EAA is used for readability throughout clause 8.

NOTE 2: (Q)EAAs or PIDs may contain unique identifiers or serial numbers; only the term identifier is used for readability throughout clause 8.

8.2 Online certificate status protocol (OCSP)

The online certificate status protocol (OCSP) is an internet protocol specified in IETF RFC 6960 [i.127] that is designed to obtain and check the current validity status of a digital X.509 PKIX certificate.

However, OCSP was not designed with privacy in mind and therefore it lacks certain privacy aspects. The OCSP protocol submits the unique identifier of a (Q)EAA to an OCSP responder, which checks revocation status of the X.509 PKIX certificate against a revocation database and returns an OCSP response with status 'good', 'revoked', or 'unknown'. So, from a privacy perspective, OCSP risks revealing more information with the OCSP responder than the user intended.

With minor changes, however, OCSP could work for (Q)EAAs containing an identifier or serial number, specifically with respect to:

- Single-show attestations, whereby each (Q)EAA has a unique identifier or serial number. This concept is equivalent to atomic (Q)EAAs that are described in clause 4.2. Hence, the EUDI Wallet will submit OCSP requests with different identifiers each time to the OCSP responder.

- OCSP Must-Staple. In an OCSP stapling scenario, the EUDI Wallet itself would query the OCSP responder at regular intervals in order to obtain a signed and time-stamped OCSP response for the user's (Q)EAA. Then the EUDI Wallet would need to append the OCSP response when presenting the (Q)EAA to the verifier. OCSP stapling is supported by TLS in the Certificate Status Request extension (see section 8 in IETF RFC 6066 [i.126]).

8.3 Revocation lists

A Revocation List (RL) is a mature and widely utilized validity status check mechanism. For detailed examples see IETF RFC 5280 [i.217] that specifies the Certificate Revocation List (CRL) profile for PKIX X.509 certificates and IETF RFC 6818 [i.128] that updates IETF RFC 5280 [i.217]. Commonly, a RL is a signed list of identifiers or serial numbers associated with the (Q)EAAs that have been revoked before they expired. Since the identifiers are unique and thus perfectly correlates with the associated (Q)EAAs, any solution that relies on a RL need to consider the following privacy aspects:

- Single-show attestations, whereby each (Q)EAA has a unique identifier or serial number. This concept is equivalent to atomic (Q)EAAs that are described in clause 4.2. Hence, the RL will contain different identifiers for the user's set of atomic (Q)EAAs.

- Range requests, which depends on the size of the RL. The privacy provided by a RL is proportionate to the size of the RL. In the extreme case with one revoked identifier in a RL, the RL provider will be able to identify what (Q)EAA the verifier or user needs to check. The larger the RL is, the more difficult it is for a RL provider to correlate the user's (Q)EAA with the requests to the RL provider.

Additionally, a RL needs to also consider the event where a batch of (Q)EAAs change status at once. In such a scenario, verifiers can collude and compare the (Q)EAA identifiers with the simultaneous validity status changes to learn more about which (Q)EAAs describe the same subject. Cryptographic techniques such as Private Set Intersection (PSI) or Private Information Retrieval (PIR) may prove helpful as solutions:

- Private Set Intersection [i.161] is a secure multiparty cryptographic technique that allows two parties holding sets to compare encrypted versions of these sets in order to compute the intersection. In this scenario, neither party reveals anything to the counterparty except for the elements in the intersection.
- Private Information Retrieval [i.24] is a protocol that allows a client to retrieve an element of a database without the owner of that database being able to determine which element was selected.

8.4 Validity status lists

A validity Status List (SL) is a bit vector that is issued and signed by an issuer (QTSP in eIDAS2 terms). The validity status of a (Q)EAA is represented using either a single bit or multiple bits in the SL bit vector. The (Q)EAA identifier is mapped to an index in the status list. The validity status check of the (Q)EAA is performed by checking the binary value of the bit(s) that is indexed in the status list bit vector. If the binary value of the indexed position in the status list is 1 (one), the (Q)EAA is revoked, else if it is 0 (zero) it is not revoked.

EXAMPLE: The (Q)EAA with the identifier 49361 is mapped to the status list index 136547. In the status list bit vector, the indexed position 136547 is a binary value of 0 (zero). Hence, the (Q)EAA is not revoked in this example.

The W3C Verifiable Credentials working group has specified "Bitstring Status List v1.0 - Privacy-preserving status information for Verifiable Credentials" [i.200] with details on how to issue status lists and check the validity status of Verifiable Credentials. IETF has specified "OAuth Status List" [i.121] that defines status list data structures for representing the status of JSON Web Tokens (JWTs) and CBOR Web Tokens (CWTs).

Status lists have the following features:

- The validity status list bit vector per se does not reveal any information about the (Q)EAA's identifier, which is a privacy preserving feature. (PKIX CRLs contain the serial numbers of the revoked PKIX X.509 certificates.).
- The size of a status list is relatively small. A status list can be constructed for 100 000 (Q)EAAs that is roughly 12,5 kB in size. This is beneficial for performance and bandwidth reasons when a verifier downloads the status list. (PKIX CRLs contain more metadata about the revoked PKIX X.509 certificates and are therefore considerably larger.).
- A verifier can retrieve the entire status list without revealing what index it will check, which is a privacy preserving feature. (An OCSP request contains the PKIX X.509 certificate serial number, which reveals what certificate a verifier needs to check.).

As with RLs, the identifier is a unique correlation handle. Consequently, any solution that relies on a SL need to also consider the following privacy preserving aspects:

- Single-Show attestations, range requests, and/or PSI cardinality as described for RLs.
- Randomized index assignment. The index associated with each (Q)EAA is randomly assigned over the entire set of possible (Q)EAAs. Consequently, chunks of the status list cannot be derived based on e.g. issuance or expiration time.
- Hiding of still valid (Q)EAAs. Status list sizes that equal the number of issued (Q)EAAs allows an attacker to learn information about still valid (Q)EAAs.

As with RL, a SL does also consider events where a batch of (Q)EAAs change status at once. Private Set Intersection and Private Information Retrieval techniques are therefore recommended to be considered.

8.5 Cryptographic accumulators

A cryptographic accumulator allows the aggregation of many values into a fixed-length digest called the accumulator value. Furthermore, and in contrast to cryptographic hash functions, it is possible to verify whether an element is accumulated or not. Asymmetric accumulators rely on a so-called (non-)membership witness. Symmetric accumulators do not require a witness for membership testing. Negative accumulators support non-membership witnesses: positive ones support membership witnesses, and universal ones support both.

A Bloom filter is an append-only data structure that can be used for a set of (non-)membership tests without any witness. These tests allow for false positives but not for false negatives. Put differently, a Bloom filter test will either yield that the tested element is possibly in the set, or that it is definitely not in the set. Multiple Bloom filters can be chained so that the false positives are included in a second Bloom filter that tests for the opposite value (e.g. the first Bloom tests for revocation; the second is a non-revocation test). This process can be repeated indefinitely to create a Bloom filter cascade with a sufficiently low false-positive rate.

In contrast to RL and SL, a Bloom filter does directly reveal information about the set elements. Any validity status change is probabilistic, which means that colluding entities cannot know if the changes reflect a simultaneous validity status change (e.g. a revocation of a batch issued (Q)EAA) or a false positive. However, the probabilities depend on the Bloom filter and it has to be designed with care as colluding verifiers can use any Bloom filter based approach that has a sufficiently low false-positive rate to link together an attestation batch in the event of a validity status change.

Many other cryptographic accumulators exist beside Bloom filters. This text mentions Bloom filters specifically due to the focus on market deployed techniques. However, other examples of market deployed solutions exist, e.g. the accumulator scheme used in Hyperledger AnonCreds [i.104] and by the IRMA [i.138] project, which is an implementation of the Idemix [i.109] attribute-based credential scheme. It is also worth mentioning more recent work that demonstrates how the witness updates can be done in a privacy friendly batch update, meaning that the witness update is the same for all users.

Camenisch and Lysyanskaya introduced the concept of dynamic accumulators in their paper "Dynamic accumulators and application to efficient revocation of anonymous credentials" [i.42] in 2002. A dynamic accumulator allows for dynamically adding or deleting a value, such that the cost of adding or deleting is independent of the number of accumulated values. The paper also provides a construction of a dynamic accumulator and an efficient zero-knowledge proof scheme, which can be proven secure under the strong RSA assumption. Such construction of dynamic accumulators enables efficient revocation of anonymous credentials and membership revocation for group signature and identity escrow schemes.

Furthermore, the first dynamic universal accumulator was introduced in 2009 in a paper by Au, Tsang, Susilo and Mu that describes how dynamic universal accumulators for DDH groups can be applied to attribute-based anonymous credential systems [i.11].

Moreover, Nguyen described accumulators from bilinear pairings and applications in a paper published in 2005 [i.163], which was extended in 2008 by Damgård and Triandopoulos in their paper "Supporting Non-membership Proofs with Bilinear-map Accumulators" [i.64]. Recently, in 2022, the research in this field was extended by Vitto and Biryukov in their paper "Dynamic Universal Accumulator with Batch Update over Bilinear Groups" [i.195].

Hence, cryptographic accumulators, and dynamic accumulators and universal dynamic accumulators are worth considering for revocation schemes when privacy requirements are high.

8.6 Using programmable ZKP schemes for revocation checks

As described in clause 6.5.1, it is possible to design anonymous credentials from programmable ZKPs (typically zk-SNARKs) and existing digital identities (such as X.509 certificates). Furthermore, the revocation and validity status can be performed at the digital wallet, whilst the validation results, selected attributes and predicates are shared with the verifier. Hence, any type of revocation verification protocol, even OCSP, can be implemented at the digital wallet, yet providing privacy for the user.

8.7 Conclusions on validity status checks

This appendix introduces the topic of revocation and validity status checks in the context of selective disclosure capable and unlinkable (Q)EAAs. It is recommended that the validity status check employed does not introduce a correlation handle in cases where selective disclosure and unlinkability are required. Concretely put, long lived (Q)EAAs that support selective disclosure and unlinkability using the mechanisms described in the present document:

- Are recommended to use OCSP in Must-Staple mode where OCSP is supported.
- May use validity Status List bit vectors rather than CRLs, since validity Status Lists do not reveal any information about the (Q)EAA's identifiers, whilst CRLs contain the revoked certificates' serial numbers.
- Cannot rely on Revocation Lists or validity Status Lists without additional privacy considerations as detailed above. Seemingly, the use of Revocation Lists or Status Lists requires Private Information Retrieval or Private Set Intersection techniques not to undermine selective disclosure and unlinkability.
- Can use cryptographic accumulators where possible given the associated complexity. Bloom filters represent an easy first step, whereas universal dynamic accumulators with public batch witness updates represent an interesting possibility for the future development of validity status checks of anonymized credentials and zero knowledge proofs.
- May be combined with ZKP schemes (such as zk-SNARK) such that the status validity checks are performed at the digital wallet, and only the relevant information is disclosed with the verifier.

Ultimately, there is no suitable validity status mechanism that is both simple, mature in terms of standards, and that matches unlinkability requirements of (Q)EAAs capable of selective disclosure and data unlinkability.

Where selective disclosure and unlinkability is required, it is presently advisable to rely on short lived (Q)EAAs. Where users are identified, and/or when using formats based on salted attribute hashes where full unlinkability guarantees cannot be made, standard solutions like RL and SL are suitable.

9 Post-quantum considerations - general remarks

A quantum computer capable of cryptanalysis remains a speculative prospect for a remote future despite the current level of trepidation. While a remote risk, the emergence of one with the computational power to execute algorithms like Shor [i.187] or Grover [i.100] could significantly affect the proposed solutions. To fully realize the impact of quantum computers, it is important to understand three things:

1) when they become a threat;
2) how quickly an attack is performed; and consequently
3) what they threaten.

One way to assess when a quantum computer can be a threat is to look at the requirements for launching a particular attack. These requirements can be expressed as logical qubits (a collection of physical qubits to protect against errors, where each logical qubit acts as the unit of information analogous to a classical bit). Proos and Zalka 2008 [i.178] show that computing the ECDL on an elliptic curve of order n field requires roughly 6*n* qubits without degradation and error rates. However, due to degradation and error rates, it makes more sense to discuss logical qubits and estimate the number of physical qubits for various degradation and error rates. For one reasonable estimate, Roetteler et al. 2017 [i.182] conclude that the ECDL on an elliptic curve defined over an *n*-bit prime field can be computed with at most 9*n*+2*ceil(log2(*n*)) + 10 qubits. This means that 2330 logical qubits are required to perform NIST P-256 point addition and the full Shor algorithm on NIST P-256 would require 1,26*10^{11} universal gates. A final, but important consideration relating to the when, is that once a malicious and extremely well-resourced entity is equipped with a quantum resource it has to choose what to employ this resource on.

Another important consideration is to estimate how quickly the attack, once possible, can be performed. This is important because the time frame for the attack determines both the required size of the quantum computer and what threat it poses. It is thus incorrect to assume that the emergence of a quantum computer capable of cryptanalysis immediately renders all classical cryptography obsolete; an attacker will carefully deploy their quantum computers and each attack takes time. It is difficult to provide an exact size estimation for a given time frame given the many assumptions that need to be made about how a future quantum computer may operate. But with reasonable assumptions, Webber et al. 2022 [i.199] estimate that breaking a 256-bit elliptic curve cryptography within a day would require 13 million physical qubits and a quantum computer capable of running Shor's algorithm [i.187].

After examining the conditions under which a quantum computer could pose a threat and the associated timeframes, the next crucial consideration is to identify the specific targets such a quantum computer would jeopardize within a defined timeframe. This elucidates the threats posed to (Q)EAAs and provides insights into potential countermeasures that prospective (Q)EAA issuers and users can take.

The most significant threat, the Harvest Now, Decrypt Later (HNDL) threat, arises when a quantum computer is utilized on the sensitive ciphertext. In this scenario, an attacker monitors the key agreement between two actors, collects the ciphertext, and employs their quantum computers to find the negotiated symmetric decryption key. The threat here is one against confidentiality, i.e. the extraction of information about the signed message that the signer did not intend to disclose or the signature value itself in ZKP-capable signature schemes. The timeframe for such an attack can span the duration during which the encrypted data retains its sensitivity. Where an (Q)EAA contains information at risk of an HNDL attack, the risk of quantum computers necessitates that the (Q)EAA Provider abstains from using encryption schemes, and/or key sizes, where quantum computers pose a threat. An (Q)EAA Provider has many possible alternatives they could rely on, such as quantum-safe algorithms, zero-knowledge proofs that are quantum resistant (e.g. those based on cryptographic hash functions), increased key sizes, or Oblivious Pseudo-Random Functions, to name a few. However, Providers are recommended to take great care in the mitigating steps they take and be entirely sure that these protect against a HNDL attack.

Another risk is that of signature and proof forging, which is arguably more relevant to the topic of the present report. Here, the risk is relatively much lower due to the time frames involved. Note that an attacker cannot begin the attack without knowledge of some public material (e.g. a public key) derived from the sensitive cryptographic material. The threat here is one against integrity and authenticity, i.e. that the attacker would need to forge signatures, disclosures, and/or proofs. Note also that the attacker does not have the same time frames at their disposal as in the case of an HNDL attack as the attack target is not a decryption key that can be used on pre-collected sensitive ciphertext. Actors may deploy frequent key rotation and rely on short-lived attestations to mitigate the quantum threat. The potential use of one-time signing and proof keys provides excellent protection against an attacker with a quantum computer. Frequent key rotation, or even one-time use of keys, is likely viable for the foreseeable future given existing development trajectories. Once the threat level is sufficiently high, actors can move to alternative signature algorithms (e.g. CRYSTALS Dilithium) and post-quantum safe zero-knowledge solutions.

EXAMPLE: The complexity of forging documents that have been digitally signed in a pre-quantum world can be illustrated by this example. Assume that Alice digitally signs a document in the pre-quantum world. The signed document is also time-stamped by a trusted time-stamping authority. She stores the digitally signed document in an archive, which has an audit log where each log entry is digitally signed and each signed log entry is added to a chain of hashes of previous log entries. In a post-quantum world, the attacker Bob will be able to derive Alice's private key from her public key in the X.509 certificate. Hence, he can create a forged document and sign this with her private key and certificate. However, in order to replace the existing signed document, which is archived, Bob would also need to attack the time-stamping authority to generate a forged time-stamp (with a rewinded clock). He would also need to attack the archive to delete the existing document, replace it with the forged document, and finally forge the signed audit log and hash chain of log entries. Such an attack is utterly complicated to perform, even with the use of quantum computers.

The related concept of everlasting privacy, which is typically applied to e-voting schemes, aims at ensuring the electronic votes will remain secret and secure also in the future. For more information on everlasting privacy the following research papers are recommended: "Practical Everlasting Privacy" [i.6] by Arapinis et al, "Towards everlasting privacy and efficient coercion resistance in remote electronic voting" [i.98] by Grontas et al, "Improvements in Everlasting Privacy: Efficient and Secure Zero Knowledge Proofs" [i.101] by Haines et al, and "SoK: Secure e-voting with everlasting privacy" [i.102] by Haines et al.

10 Conclusions

The eIDAS2 regulation and the Architecture and Reference Framework (ARF) define regulatory requirements on selective disclosure and unlinkability for the EUDI Wallet. The present ETSI technical report provides a comprehensive analysis of signature schemes, credential formats and protocols that cater for selective disclosure, unlinkability, and predicates.

Since the ARF specifies the data model and security mechanism detailed in the ISO mobile driving license (ISO mDL) and compliance with the W3C Verifiable Credentials Data Model 1.1 (VCDM) representation for JWT as Person Identification Data (PID) formats for the EUDI Wallet, the present document analyses 1) ISO mDL 2) W3C VCDM representation for JWT in conjunction with SD-JWT, and 3) SD-JWT VC as a standalone attestation format.

The ISO mDL specified mdoc MSO and the SD-JWT formats and related presentation protocols cater for selective disclosure using a hashed salted attributes approach. Both MSO and SD-JWT support SOG-IS approved cryptographic algorithms and can also be used with quantum-safe cryptography for future use. The conclusion is thus that MSO (as detailed in ISO mDL) as well as the SD-JWT approach meet the eIDAS2 regulatory and technical requirements on selective disclosure when defined as revealing at least one attribute from a single PID or (Q)EAA. Neither format supports selective disclosure of at least two attributes from multiple distinct PID/(Q)EAAs. Neither format supports predicates, although the present document also proposes a new approach to calculate predicates based on hash chains in conjunction with salted attribute hashes, which can be used for dynamically deriving statements about the user without revealing the attribute values.

In addition to limited selective disclosure capabilities, the major drawback with ISO mDL MSO and SD-JWT is the lack of unlinkability. Neither of the formats supports issuer unlinkability or full unlinkability, and verifier unlinkability encumbers the issuer. In order to achieve verifier unlinkability, batches of ISO mDL MSOs or SD-JWTs need to be issued to each EUDI Wallet. When the PID Provider (PIDP) or QTSP supports batch issuance with unique salts, both MSO and SD-JWT can support verifier unlinkability. In order to achieve verifier unlinkability, the random salts in the ISO mDL MSO and SD-JWT should be unique, meaning that refreshed MSOs and SD-JWTs are presented to a relying party.

The present document gives recommendations on how eIDAS2 compliant PIDPs or QTSPs can issue PID/(Q)EAAs in the form of ISO mDL and/or SD-JWT that cater for selective disclosure. For use cases that require W3C VCDM v1.1 compliant representation for JWT, the present document recommends using a transformation algorithm to recreate the original W3C VC. Such a transformation is greatly simplified if SD-JWT VC DM is used to communicate schemas and vocabularies. The present document notes that SD-JWT can provide selective disclosure capability also for attestations that use JSON-LD and linked data proofs but advises against it (support for data integrity proofs is lacking and there exist security concerns with polyglot parsing).

There are many similarities between the ISO mDL issuers and the eIDAS2 QTSPs or PID providers, which could be harmonised in ETSI TS 119 471 [i.80] and ETSI TS 119 472-1 [i.81] that will standardize the issuance policies and profiles of (Q)EAAs. More specifically, the ISO mDL MSO could be issued by an eIDAS2 QTSP certification authority, meaning that the EU trusted lists can be used to retrieve revocation information and trust anchors when validating the ISO mDL MSO signature. ETSI TS 119 495 [i.77], which specifies certificate profiles and TSP policies for Open Banking and PSD2, may partially be re-used for the issuance of ISO mDLs as (Q)EAAs. The same principles could be applied on QTSPs and PID providers that will issue PIDs/(Q)EAAs in conjunction with SD-JWT, although the existing specifications do not specify the issuance policies in detail.

Furthermore, there are recommendations on how to store ISO mDL MSO and W3C VCDM 1.1 compliant representation for JWT in the EUDI Wallet, and how to present selectively disclosed attributes to eIDAS2 relying parties. The presentation protocols for the ISO mDL and OID4VP/SIOP2 are specified in the ARF, and the present report describes how to use these protocols for selective disclosure of attributes in ISO mDL and SD-JWT.

The multi-message signature schemes on the other hand are designed to provide selective disclosure and full unlikability. Such multi-message signature schemes are BBS+, CL-Signatures, PS-MS signatures and Mercurial signatures. However, such signature schemes are based on pairing-based elliptic curve cryptographic algorithms that are not yet fully standardized. So far, ISO/IEC 20008 [i.143] has standardized single-message signature schemes that underpin BBS and PS-MS, but they are not sufficient for PID formats and (Q)EAAs that require multi-message signature schemes. However, ISO/IEC PWI 24843 intends to standardize BBS+ with blinded signatures, which may allow for a future standard that could be used in compliance with the EUDI Wallet requirements on selective disclosure and unlinkability in eIDAS2. Furthermore, there are cryptographic research projects, such as MoniPoly, where undisclosed attributes have no impact on the proof size.

Another interesting approach to achieve solutions for the EUDI Wallet with selective disclosure and full unlinkability are the systems that combine ZKP schemes (such as zk-SNARKs) with existing digital identity infrastructures (such as X.509 certificates or ICAO eMRTD). There are existing research projects, such as Cinderella and zk-creds, that have succeeded to implement prototypes where zk-SNARKs are used to generate pseudo-certificates that share selected attributes from the (Q)EAAs and derived revocation information. These projects are still in the research phase, but may be considered for the EUDI Wallet and eIDAS2 relying parties.

In order to achieve privacy preserving features for revocation and validity status checks it is recommended to use OCSP in Must-Staple mode, implement Revocation Lists or validity Status Lists with additional privacy techniques such as Private Information Retrieval or Private Set Intersection, and use cryptographic accumulators where possible given the associated complexity. If ZKP schemes (such as zk-SNARKs) are combined with existing (Q)EAAs (such as X.509), the status validity checks are performed at the EUDI Wallet, and only the relevant information is disclosed with the verifier.

Annex A:
Comparison of selective disclosure mechanisms

A.1 Selective disclosure signature schemes

Table A.1 provides a comparison of the investigated selective disclosure signature schemes.

Table A.1: Comparison of selective disclosure signature schemes

Signature scheme	Cryptography	Plausible quantum-safe	Unlinkability	Predicates	Reference
Category: Atomic attribute (Q)EAAs					
Atomic attribute (Q)EAAs	Conditional: depends on the signature on the credential	Yes, the (Q)EAAs can be signed with QSC algorithms.	Verifier unlinkable attestations can be achieved. Fully unlinkable (Q)EAAs are not possible.	No dynamic predicates are supported. Workaround: enrol for atomic attributes with Boolean attributes.	See clause 4.2
Category: Multi-message signature schemes					
BBS+ signatures	Multi-message signature scheme based on ECC bilinear pairings	ZKPs generated pre-quantum will remain plausible safe post-quantum. BBS+ is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.3.1
Camenisch- Lysyanskaya (CL) signatures	Multi-message signature scheme based on strong RSA assumption	ZKPs generated pre-quantum will remain plausible safe post-quantum. CL-signatures are plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.3.2
Mercurial Signatures	Multi-message signature scheme based on decisional Diffie-Hellman (DDH)	ZKPs generated pre-quantum will remain plausible safe post-quantum. MS is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.3.3
Pointcheval- Sanders Multi-Signatures (PS-MS)	Multi-message signature scheme based on improved CL-signatures	ZKPs generated pre-quantum will remain plausible safe post-quantum. PS-MS is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.3.4

Signature scheme	Cryptography	Plausible quantum-safe	Unlinkability	Predicates	Reference
Category: Salted attribute hashes					
Salted attribute hashes	**Salted attribute hashes**, signed with RSA, ECC, or QSC	Yes, the (Q)EAAs can be signed with QSC algorithms.	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not protected against issuer linkability	No dynamic predicates are supported. Workaround: set Boolean attributes in the PID/(Q)EAA.	See clause 4.4
ACDC	Salted attribute hashes structured in a Directed Acyclic Graph	Yes	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not fully unlinkable.	No dynamic predicates are supported. Workaround: set Boolean attributes in the PID/(Q)EAA.	See clause 4.4.8
Gordian Envelopes	Salted attribute hashes structured in a Directed Acyclic Graph	Yes	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not fully unlinkable.	No dynamic predicates are supported. Workaround: set Boolean attributes in the PID/(Q)EAA.	See clause 4.4.9
HashWires	Salted attribute hashes structured in a chain of hashes	Yes	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not fully unlinkable.	HashWires supports range proofs that can be combined with selectively disclosed salted hashes of attributes (see clause 4.4.7.4)	See clause 4.4.7
Category: Proofs for arithmetic circuits (programmable ZKPs)					
zk-SNARKs	Proofs for arithmetic circuits based on various mechanisms in Annex A.4	Some zk-SNARK schemes are QSC, see table A.4	Yes	Yes	See clause 4.5.2 and clause A.4

A.2 (Q)EAA formats with selective disclosure

Table A.2 provides a comparison of the investigated credential formats with selective disclosure.

Table A.2: Comparison of credential formats with selective disclosure

(Q)EAA format	Scheme	Encoding	Maturity	Reference
Category: Atomic attribute credentials				
IETF X.509 attribute certificates	Atomic attribute (Q)EAAs	ASN.1/DER	X.509 attribute certificate (IETF RFC 5755 [i.125]) is an IETF PKIX standard	See clause 5.2.2
W3C Verifiable Credentials	Atomic attribute (Q)EAAs	JSON-LD or JWT	W3C VC Data Model [i.209] is a standard	See clause 5.2.3
Category: Multi-message signature schemes				
Hyperledger AnonCreds	CLRSA-signatures	JSON (JWS)	Deployed in Government of British Columbia, IDunion, and the IATA Travel Pass	See clause 5.3.4
W3C VC with CL-signatures	CL-signatures	JSON (LD)	W3C VC Data Model [i.209], implemented in several wallets	See clause 5.3.1
W3C VC Data Integrity with BBS+ signatures	BBS+ signatures	JSON (LD)	W3C VC Data Integrity [i.208]	See clause 5.3.2
W3C VC Data Integrity with ECDSA-SD	ECDSA-SD signatures	JSON (LD)	W3C VC Data Integrity [i.208]	See clause 5.3.3
Category: Salted attribute hashes				
IETF SD-JWT	salted attribute hashes	JSON (JWT)	IETF SD-JWT draft standard [i.123], several reference implementations	See clause 5.4.2
ISO/IEC 18013-5 [i.140] MSO (Mobile Security Object)	salted attribute hashes	CBOR/CDDL (COSE)	ISO/IEC 18013-5 [i.140], implemented in several wallets, deployed in the US	See clause 5.4.3
Category: JSON container formats				
IETF JSON Web Proof	Flexible: CL-signatures, BBS+, etc.	JSON (JWS)	IETF JSON Web Proof draft standard [i.75]	See clause 5.5.1
W3C JSON Web Proofs For Binary Merkle Trees	Merkle trees	JSON Web Proofs	W3C draft specification	See clause 5.5.1

A.3 Selective disclosure systems and protocols

Table A.3 provides a comparison of the investigated selective disclosure protocols.

Table A.3: Comparison of selective disclosure systems and protocols

Protocol	Credentials	Protocol	Maturity	Reference
Category: Atomic attribute (Q)EAAs				
IETF X.509 attribute certificate (protocol)	IETF X.509 attribute certificates	Attribute certificate authorization protocol	X.509 attribute certificate [i.125] is an IETF PKIX standard	See clause 6.2.1
VC-FIDO	W3C Verifiable Credentials	VC-FIDO	Deployed as a prototype at NHS in the UK	See clause 6.2.2
Category: Multi-message signature schemes				
Hyperledger AnonCreds (protocol)	AnonCreds [i.104] based on CLRSA-signatures	Hyperledger Aries protocol [i.105] in conjunction with Hyperledger AnonCreds SDK [i.104]	Deployed in Government of British Columbia, IDunion, and the IATA Travel Pass	See clause 6.3.1
Direct Anonymous Attestation (DAA)	DAA credentials	ISO/IEC 20008-2 [i.143]	Deployed at large scale by TCG in TPM 2.0 and Intel® in EPID 2.0	See clause 6.3.2
Category: Salted attribute hashes protocols				
Singapore's Smart Nation OpenAttestation	Document Integrity credentials	OpenAttestation protocol [i.169]	Deployed at the Singapore's Smart Nation	See clause 6.4.1
Category: Proofs for arithmetic circuits solutions				
Cinderella	X.509 certificates	zk-SNARK	In research phase	See clause 6.5.2
zk-creds	ICAO eMRTDs	zk-SNARK (Pinocchio)	In research phase	See clause 6.5.3
Category: ABC (Attribute Based Credentials)				
Idemix	Idemix ABC credentials [i.109] based on CL-signatures	Idemix ABC protocol [i.109]	Implemented by IBM®, Hyperledger Fabric [i.106], IRMA project [i.180], and the EU-projects PrimeLife [i.177] and ABC4Trust [i.110]	See clause 6.6.1
U-Prove	U-Prove ABC credentials [i.160]	U-Prove ABC protocol [i.160]	Implemented in Microsoft® Identity Metasystem and the EU-project ABC4Trust [i.110]	See clause 6.6.2
ISO/IEC 18370 [i.142]	U-Prove ABC credentials [i.160]	ISO/IEC 18370 [i.142]	Implemented in U-Prove solutions, security flaws detected	See clause 6.6.3
Keyed-Verification Anonymous Credentials (KVAC)	Keyed-Verification Anonymous Credentials	BBS_MAC+ [i.13]	Implemented as a prototype on SIM-cards	See clause 6.6.4
Category: ISO mobile driving license (ISO mDL)				
ISO/IEC 18013-5 [i.140] (device retrieval)	ISO/IEC 18013-5 [i.140] mDL/MSO [i.140]	ISO mDL/MSO over BLE/NFC	ISO standard, implemented in several wallets, deployed in the US	See clause 6.7.2
ISO/IEC 18013-7 [i.141] (unattended)	ISO/IEC 18013-5 [i.140] mDL/MSO [i.140]	SIOP2 [i.173], OID4VP [i.171]	Draft ISO/IEC CD 18013-7 [i.141] standard, correlated with ISO/IEC CD 23220-4 [i.146]	See clause 6.7.4
ISO/IEC 23220-4 [i.146]	ISO mDL [i.140], SD-JWT [i.123], etc.	SIOP2 [i.173], OID4VP [i.171]	Draft standard, correlated with ISO/IEC CD 18013-7 [i.141]	See clause 6.7.5
ISO/IEC 18013-5 [i.140] (server retrieval)	OpenID Connect ID-Token [i.170]	OpenID Connect (OIDC) Core [i.170]	ISO standard, implemented in several wallets, deployed in the US	See clause 6.7.3

A.4 zk-SNARK protocols

Table A.4 provides a comparison of the different zk-SNARK protocols.

The comparison is made based on transparency, universality, and plausible quantum-safety. A transparent protocol is defined as it does not require any trusted setup and uses public randomness. A universal protocol is defined as it does not require a separate trusted setup for each circuit. A plausibly quantum-safe protocol is one that is not considered to be vulnerable to attacks by quantum computing algorithms.

Table A.4: Comparison of zk-SNARK protocols

Protocol	Published	Transparent	Universal	Quantum-safe
Pinocchio [i.174]	2013	No	No	No
Geppetto [i.60]	2015	No	No	No
TinyRAM [i.17]	2013	No	No	No
Buffet [i.196]	2015	No	No	No
ZoKrates [i.70]	2018	No	No	No
xJsnark [i.154]	2018	No	No	No
vnTinyRAM [i.19]	2014	No	Yes	No
MIRAGE [i.153]	2020	No	Yes	No
Sonic [i.157]	2019	No	Yes	No
Marlin [i.57]	2020	No	Yes	No
PLONK [i.93]	2019	No	Yes	No
Spartan [i.159]	2019	No	Yes	Yes
SuperSonic [i.37]	2020	Yes	Yes	No
Hyrax [i.197]	2018	Yes	Yes	No
Halo [i.29]	2019	Yes	Yes	No
Virgo [i.214]	2020	Yes	Yes	Yes
Ligero [i.3]	2017	Yes	Yes	Yes
Aurora [i.18]	2019	Yes	Yes	Yes
zk-STARK [i.12], [i.150]	2018	Yes	Yes	Yes
Bulletproofs [i.36]	2017	Yes	Yes	No

Annex B:
Code examples

B.1 Hash chain code example

This annex contains a Python code example of how to use hash chains to calculate a predicate of a user's age.

```
import secrets
from hashlib import sha256

# Get the user's age
while True:
  try:
    age = int(float(input("Enter your age: ")))
    if age < 0:
      raise ValueError
    break
  except ValueError:
    print("Enter a non negative number.")

# The issuer generates a seed and the commitment the user will need.
seed = secrets.token_bytes()
commitment = sha256(seed)
hash_chain = [commitment.hexdigest().encode('ascii')]

# The issuer then generates the hash chain.
for i in range(age):
  commitment = sha256(commitment.hexdigest().encode('ascii'))
  hash_chain.append(commitment.hexdigest().encode('ascii'))

# The hash chain is reversed so that the index values equal age
hash_chain.reverse()

# The issuer includes the following claim in the signed attestation
age_is_zero = hash_chain[0]

# The verifier wants a proof for age_over_n
n = 10
age_proof = None

# The user has to generate the following age proof
assert isinstance(n, int) and n >= 0, "The value is a non-negative integer."
try:
  age_proof = hash_chain[n] if n != 0 else age_is_zero
  print(f"The proof value is: {age_proof}")
  print(f"Copy this value for the next cell's input prompt: {age_proof.decode('ascii')}")
except IndexError:
  print(f"The user does not have a long enough hash chain for the required age proof of {n}")

# The user sends the age proof to the verifier, who verifies the chain length
age_proof_test = input("Copy paste the provided value from the previous cell: ")
age_proof_test = age_proof_test.encode('ascii')

above_n = False
if n == 0 and age_proof_test == age_is_zero:
  above_n = True
else:
  for i in range(n):
    age_proof_test = sha256(age_proof_test).hexdigest().encode('ascii')
  above_n = True if age_proof_test == age_is_zero else False

print(f"The user provided valid proof for the age is equal to or greater than {n} test: {above_n}")
```

B.2 HashWires for SD-JWT and MSO

Code examples in Python and descriptions on how to use HashWires for inequality tests for SD-JWT and MSO have been provided by Peter Lee Altmann at the repository "Inequality tests in salted attribute digest based attestations" [i.4].

Annex C:
Post-quantum safe zero-knowledge proofs and anonymous credentials

C.1 General

This annex describes research and innovations of new types of ZKP schemes. These types of innovative ZKP schemes are still being researched at an academic level and are not yet standardized, so they cannot be considered for the EUDI Wallet at the time of writing (April 2024). Nevertheless, the research on ZKP schemes is described in this annex since they may be implemented and standardized, which could be of interest for future standardization of the EUDI Wallet.

C.2 Quantum physics applied on ZKP schemes

C.2.1 Background

The advent of quantum computers is typically considered a disruption for classic cryptography. In 1994 Peter Shor published the paper "Algorithms for quantum computation: discrete logarithms and factoring algorithm" [i.187] that described how quantum computers can use certain algorithms for finding discrete logarithms and factoring integers. As a consequence, classic asymmetric cryptographic algorithms such as RSA and ECDSA, which are based on the discrete logarithm problem, are vulnerable against quantum computing attacks in a post-quantum world.

One countermeasure is to invent quantum-safe cryptography (QSC) algorithms, i.e. cryptographic algorithms (typically public-key algorithms) that are expected to be secure against a cryptanalytic attack by quantum computers. NIST conducts a research program [i.168] to identify candidates for QSC algorithms that can be standardized. The signature scheme finalists (December 2023) are FALCON [i.63], FIPS 204 [i.166] (based on CRYSTALS Dilithium [i.63]) and FIPS 205 [i.167] (based on SPHINCS+ [i.189]).

Furthermore, Dutto et al has published the paper "Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity" [i.69], which analyses quantum-safe variants of BBS+ and CL-signatures based on a lattice-based scheme. The paper also identifies the open issues for achieving VCs suitable for selective disclosure, non-interactive renewal mechanisms, and efficient revocation.

NOTE: The countermeasures above describe lattice-based or hash-based algorithms that are executed in classic computers with the intention to protect against quantum computing attacks with Shor's algorithm, but the QSC algorithms per se are not designed for quantum computers.

On the contrary to quantum computing attacks on classic cryptography, quantum physics and quantum computers can be used as an advantage when designing cryptographic protocols for a post-quantum world. There exist Quantum Key Distribution (QKD) protocols and quantum-based ZKP schemes, which are described in the following clauses.

C.2.2 Quantum key distribution (QKD)

The most mature quantum cryptographic application is Quantum Key Distribution (QKD), which utilizes quantum mechanics to share a random secret key with two parties, which then can be used to encrypt and decrypt messages. A unique property of quantum key distribution is the ability to detect if any third party has tried to eavesdrop on the communication channel between the two parties. The first QKD scheme was BB84 [i.22] that was invented by Charles Bennett and Gilles Brassard in 1984. BB84 is based on Heisenberg's uncertainty principle and uses the polarization state of photons to encode key bits, which means that the quantum data encoded as photons cannot be copied or measured without disturbing the key exchange protocol. There exist several commercial products that implement QKD schemes, which can be used for example to share symmetric AES keys. A tutorial on QKD with more information on this subject is published by IEEE [i.215].

C.2.3 Quantum physics applied to the graph 3-colouring ZKP scheme

The graph 3-colouring (G3C) problem is a classic problem that was introduced already in 1856. The graph 3-colouring problem takes as input a graph (G) and decides whether it can be coloured using only three (3) colours, such that no two adjacent vertices (nodes) have the same colour. The graph 3-colour problem is proven to be NP-complete.

The graph 3-colouring problem can be used as a ZKP scheme as described below.

Let G be a graph with n vertices and define the set of vertices as $V = \{v_1, ..., v_n\}$. Also define the set of edges as $E = \{e_{i,j}\}$. where $e_{i,j}$ is the edge between vertices v_i and v_j. The graph G is known to both parties. The prover's private knowledge is the 3-colouring of the graph G. whilst the verifier only knows the graph shape (with black "hidden" colours). The protocol is executed as follows:

1) Prover: Randomly permute the 3-colours of graph G. Commit to the permutation of the colours of all vertices, such that $c_i = P(v_i, \textit{colour of } v_i)$.

2) Prover: Share the graph G (with black "hidden") colours to the verifier.

3) Verifier: Select edge $e_{i,j}$ and send $e_{i,j}$ to the prover.

4) Prover: Open c_i and c_j.

5) Verifier: Accept if $c_i \neq c_j$, else reject.

The protocol is illustrated with the figures below.

In step 1, the prover permutes the colours of a graph G as illustrated in the figure below. Two permutations are shown in figure C.1, and the prover commits to permutation P_2 in this example.

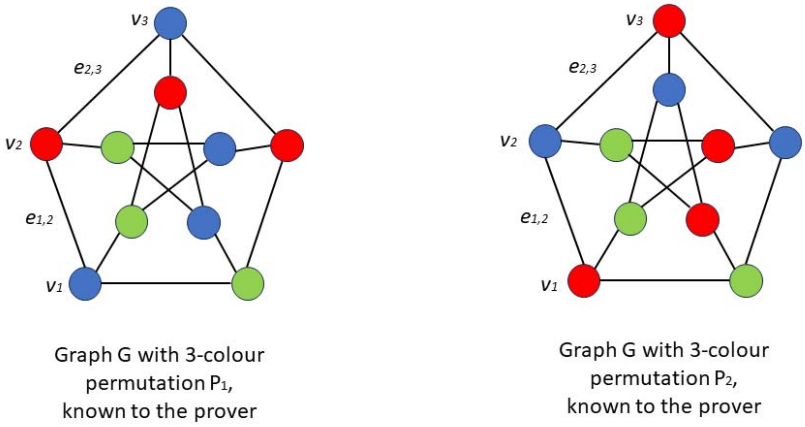

Figure C.1: Examples of 3-coloured graphs

The prover shares the graph G (with hidden colours) with the verifier, as shown to the left in figure C.2. The verifier selects edge $e_{1,2}$ whereupon the prover opens vertices v_1 and v_2. Since v_1 is red and v_2 is blue, i.e. the colours are different, the verifier can accept the proof.

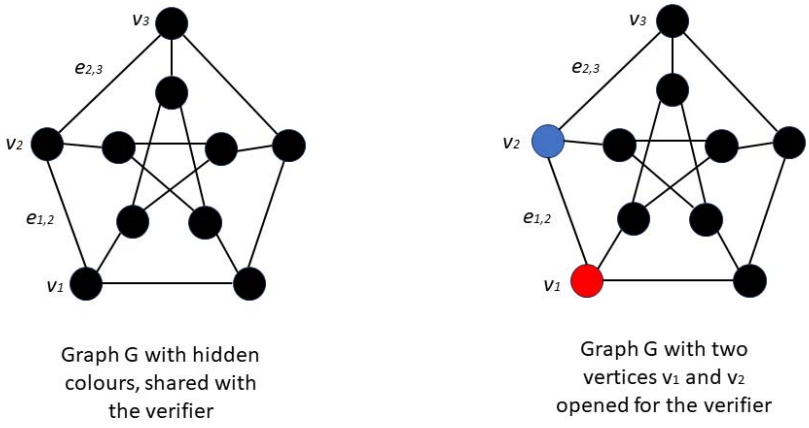

Figure C.2: Example of 3-coloured graph ZKP

Hence, the prover's knowledge is the 3-colouring permutation of the graph, and can prove this for each edge of the graph to the verifier. The prover's zero-knowledge proofs are the vertices that are opened to the verifier.

A formal description of the graph 3-colouring ZKP scheme is described as Zero-Knowledge Protocol for Graph Isomorphism in the paper "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems" [i.96] published in 1991 by Goldreich et al.

The classic graph 3-colouring ZKP scheme can be transposed to the quantum world. Simply put, large entangled quantum states are utilized for a graph in a quantum computer, equivalent to how the colour permutations are computed on a graph in a classic computer. The quantum graphs may also be shared between the prover and verifier by using the quantum key distribution as described in the previous clause. The paper "Experimental relativistic zero-knowledge proofs" [i.3] describes how the graph 3-colouring ZKP can be implemented in a way that is theoretically quantum computing safe:

- The quantum cryptography behind the graph 3-colouring ZKP schemes goes beyond the scope of the present report. For further reading the following research papers are recommended: "Zero-knowledge against quantum attacks" [i.198] by Watrous, "Post-quantum Efficient Proof for Graph 3-Coloring Problem" [i.72] by Ebrahimi, and "Zero-knowledge proof systems for QMA" [i.33] by Broadbent et al.

C.2.4 ZKP using the quantum Internet (based on Schnorr's algorithm)

Another quantum ZKP scheme is based on Schnorr's algorithm on non-interactive zero-knowledge proof [i.134].

Assume that the prover wants to prove that it knows the secret value x such that $Y = g^x \bmod p$, for prime p and generator g, with g, p, and Y public. Schnorr's algorithm can then be performed as follows:

1) The prover chooses the value r and calculates $t = g^r \bmod p$. The prover sends value t to the verifier.

2) The verifier sends the random value c to the prover.

3) The prover calculates $s = r + cx$, and sends the value s to the verifier.

4) The verifier checks that $g^s \equiv t \times Y^c \bmod p$.

Schnorr's algorithm can be proven as follows:

$$t \times Y^c \equiv g^r \times (g^x)^c \bmod p$$

$$\equiv g^{(r+cx)} \bmod p$$

$$\equiv g^s \bmod p$$

Carney has described [i.47] how to replace the use of the generator g in Schnorr's scheme for a quantum mechanical qubit rotation, and how to perform zero-knowledge proofs using quantum algorithms over the quantum Internet. The applied quantum cryptography goes beyond the scope of the present report, but for further reading the paper "On Zero-Knowledge Proofs over the Quantum Internet" [i.47] is recommended.

C.2.5 Conclusions on quantum ZKP schemes

Quantum cryptography takes advantage of quantum computers to design new cryptographic protocols for a post-quantum world.

The Quantum Key Distribution (QKD) schemes are rather mature and are implemented in several commercial products. Hence, the QKD schemes may be used for sharing keys between two parties using classic ZKP schemes.

Several quantum cryptographic algorithms for use with ZKP are also being developed. The classic graph 3-colouring scheme and Schnorr's algorithm have been transposed into quantum cryptographic algorithms. There are also relativistic quantum ZKP protocols [i.3] with promising applications for identification tasks and blockchain applications such as cryptocurrencies or smart contracts.

The quantum ZKP schemes are still being researched at an academic level and are not yet standardized, so they cannot be considered for the EUDI Wallet yet. It is however worthwhile to monitor the research and development of quantum ZKP schemes: if the quantum ZKP schemes get standardized and implemented in commercial products they could be considered for a future revision of the eIDAS regulation.

C.3 Lattice-based anonymous credentials schemes

C.3.1 Background

The transition to post-quantum cryptography is an enormous challenge for cryptographers and the IT-security industry as a whole. There have been significant enhancements such as the future NIST standards on Post-Quantum Safe (PQS) cryptography. However, these NIST standards have so far only been focusing on general cryptographic mechanisms, such as digital signatures or key exchange, whilst there are not yet any similar PQS standardization efforts for blind signatures, group signatures, and anonymous credentials.

Nevertheless, there are cryptographic research initiatives in the field of PQS multi-message signatures and anonymous credentials. In 2016, Libert et al published the research paper "Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions" [i.156]. The result of this research indicated that anonymous credential schemes, which are based on plausibly PQS cryptography using lattices, generate signature and proof sizes in the magnitude of several hundreds of MB. This lattice-based scheme is however outdated, and the research to improve the performance and proof sizes has continued as described in clause C.3.2.

Another option is to apply PQS zk-SNARKs to the Cinderella project (see clause 6.5.2), whereby PQS ZKPs can be derived from X.509 certificates. Potential PQS zk-SNARKs for such a setup are Spartan [i.159], Virgo [i.214] or Ligero [i.18]. Furthermore, the X.509 certificates would need to be signed with PQS cryptographic algorithms, such as CRYSTALS Dilithium [i.63]. There are also programmatic issues to be resolved with such an integration, such as patching the vulnerability in the Gepetto compiler.

Hence, until recently there have essentially been two alternatives to achieve a plausible PQS ZKP system: a system with large signature and proofs that rely upon cryptographic algorithms, or a system based on ad-hoc integrations of PQS zk-SNARKs. The research of how to improve the performance and proof sizes of PQS ZKP systems has however progressed in recent years, which is further described in clause C.3.2.

C.3.2 Research on effective lattice-based anonymous credentials

In order to address the issues with large sized signatures, cryptographic research is currently being performed on PQS anonymous credentials with small signature sizes.

In 2022, Jeudy et al published the cryptographic research paper "Lattice Signature with Efficient Protocols, Application to Anonymous Credentials" [i.151]. The paper introduced a new construction that is based both on standard lattices and structured ones, which resulted in significant performance improvements. In particular, the size of a signature proof was reduced to less than 650 KB.

Based on Jeudy's research, Dutto et al proposed a PQS ZKP scheme in their paper "Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity" [i.69], which describes PQS variants of BBS+ and CL-signatures based on a lattice-based scheme.

The research by Jeudy et al was continued in 2024 by Argo et al who published their research paper "Practical Post-Quantum Signatures for Privacy" [i.7] that proposes privacy-preserving signatures with efficient protocols (SEP). The SEP is lattice-based and generates short-sized signatures that are PQS. Furthermore, the SEP has been integrated with an anonymous credential system, resulting in anonymous credentials of less than 80 KB. The source code of this project is published at the repository "Lattice Anonymous Credentials" [i.8].

Furthermore, Bootle et al published the research paper "A Framework for Practical Anonymous Credentials from Lattices" [i.27] in 2023. Their paper introduces a framework for practical anonymous credential schemes based on a new family of lattices. The security of this lattice scheme is based on the difficulty to generate a pre-image for an element given short pre-images of random elements in a set. Such a framework can be used to implement efficient privacy-preserving cryptographic primitives for blind signatures, anonymous credentials, and group signatures.

Hence, there are several cryptographic research initiatives that aim at inventing anonymous credentials and privacy-preserving signature schemes that are PQS with efficient and small-sized signature proofs.

Annex D: Bibliography

- Ben-or-Goldwasser-Shafi-Kilian-Wigderson: "Multi prover interactive proofs: How to remove intractability assumptions".
- Camenisch-Dubovitskaya-Lehmann: "Concepts and Languages for Privacy-Preserving Attribute-Based Authentication".
- ENISA: "Cybersecurity Certification: Candidate EUCC Scheme V1.1.1".
- ETSI EN 319 102-1: "Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation".
- ETSI EN 319 403-1: "Electronic Signatures and Infrastructures (ESI); Trust Service Provider Conformity Assessment; Part 1: Requirements for conformity assessment bodies assessing Trust Service Providers".
- ETSI EN 319 411-2: "Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 2: Requirements for trust service providers issuing EU qualified certificates".
- ETSI TR 103 619: "CYBER; Migration strategies and recommendations to Quantum Safe schemes".
- ETSI TS 119 182-1: "Electronic Signatures and Infrastructures (ESI); JAdES digital signatures; Part 1: Building blocks and JAdES baseline signatures".
- FIDO Alliance: "Fast Identity Online v2 (FIDO2)".
- Fuentes-González-Olvera-Veseli: "Assessment of attribute-based credentials for privacy-preserving road traffic services in smart cities".
- IETF RFC 6749: "OAuth 2.0 Authorization Framework".
- OASIS: "PKCS #11 Cryptographic Token Interface Base Specification Version 2.40".
- OpenID Foundation: "OpenID for Verifiable Credential Issuance".
- W3C®: "JSON-LD 1.1 - A JSON-based Serialization for Linked Data".
- W3C® Working Draft 21 July 2023: "Securing Verifiable Credentials using JOSE and COSE".
- Zhang-Genkin-Katz-Papadopoulos: "vRAM: Faster Verifiable RAM with Program-Independent Preprocessing".

Annex E:
Change history

Date	Version	Information about changes
August 2023	1.1.1	Publication
January 2024	1.1.2	Stable draft with updates made according to ESI(23)000072 "Comments on ETSI TR 119 476 V1.1.1 for the revision to ETSI TR 119 476 v1.2.1".
February 2024	1.1.3	Stable draft with updates made according to ESI(24)082054 "Resolved collated comments on ETSI TR 119 476 v1.1.2".
March 2024	1.1.4	Editorial edits based on feedback from ETSI's directorate.
April 2024	1.1.5	Final draft with updates made according to ESI(24)82b004 "Resolved collated comments on ETSI TR 119 476 v1.1.4".

History

Document history		
V1.1.1	August 2023	Publication
V1.2.1	July 2024	Publication