Concepts for Secure Wallets in Decentralised Identity Ecosystems

Paul Bastian[1], Micha Kraus[1], Jörg Fischer[1]

[1] Bundesdruckerei GmbH, Berlin, Germany.
paul.bastian@bdr.de

Abstract. Decentralised identity ecosystems offer promising solutions for numerous applications in the private sector and public administration. The applications have very different requirements and regulated environments place high security requirements on the wallet and the credentials it safeguards. At the same time, the smartphone market offers a fragmented range of security solutions. We investigate the security requirements for mobile, native wallet architecture for self-sovereign users and evaluate the existing security solution building blocks for hardware-bound key storage, biometrics and features of Android and iOS operating systems. The regulatory requirements are addressed through measures such as device binding, user retention and authentication of the wallet and the requesting party. We analyse and evaluate the different variants and characteristics and develop from these an interoperable and privacy-oriented procedure for the trustworthy issuance and verification of identity credentials, and we describe the trust model behind it. We discuss the advantages and disadvantages of the system and provide an outlook on future developments in wallet security.

Keywords: *wallet*, *self-sovereign identity*, device binding, holder binding, digital identity, hardware-bound security

1 Introduction

Digital identities are an essential building block to facilitate the digitisation of our society and the economy as a whole, both in the state and private sector . The activities of the last few years show an increasing interest on many levels: More and more states are funding identity-related support projects, numerous new standards are emerging and the European Union (EU) is planning a European wallet for digital identities as part of the eIDAS revision (European Commission 2022b). In this context, the focus is increasingly on decentralised identity solutions, shifting interest from traditional identity ecosystems to user-centric, self-sovereign solutions based on the principle of self-sovereign identity (SSI) (Strüker et al. 2021). In our paper, we focus on the solutions and visions for wallet security within a decentralised identity ecosystem.

2 Background

Decentralised identity ecosystems usually comprise four different roles. The W3C Verifiable Credentials Data Model (Sporny et al. 2022) describes the issuer who issues credentials to the holder and the holder who receives the credentials, stores them in a digital wallet and presents them to the verifier. One verifiable data registry acts as a trust anchor and includes, for example, identifiers, public keys and public actor schemas. Fig. 1 shows the roles of a decentralised identity ecosystem and their trust relationships.

A wallet is a digital software component that performs actions on behalf of and under the control of its holder. In the same way that a physical wallet contains identity documents, the digital wallet can securely store and manage multiple credentials and their associated keys. It implements protocols and their interfaces[1] to receive credentials from issuers and send them as presentations to verifiers. The wallet authenticates the holder and usually only executes transactions with the holder's explicit consent.

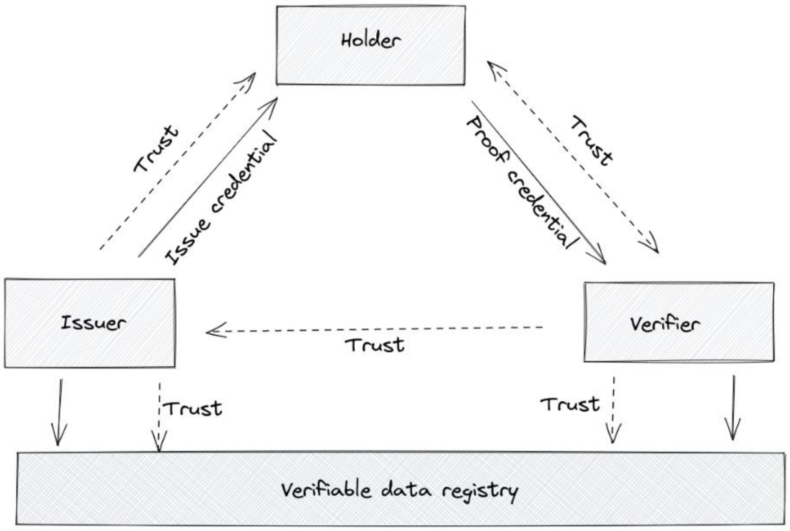

Fig. 1: Roles of decentralised identity ecosystems and trust relationships

Credentials are a collection of one or more attributes about a subject made available by an issuer. Verifiable credentials are tamper-proof credentials that can be cryptographically verified by a verifier. As a rule, this verification identifies the issuer and contains further metadata such as the date of issue and expiry. Currently, various data formats for verifiable credentials are being developed and used, such as W3C Verifiable Credentials, AnonCreds or ISO mdoc (ISO/IEC JTC 1/SC 27 2021; Young 2021; Bastian 2022). In the context of this paper, we use the term credentials to mean verifiable credentials.

The advantages of an open, decentralised identity ecosystem lie in the variety and breadth of possible applications. The ability to combine state and private sector, sensitive and everyday credentials in one wallet can provide the holder with greater convenience, trust and security in everyday use. Many scenarios for such combinations can already be found in everyday life today: to rent a car, an ID card, driving licence and credit card are required; for a job application, ID card, references and

[1] In this paper, we do not make a distinction between wallets and agents, as can be found in the literature and other contributions, because in our opinion it does not add any value to the core concepts of wallet security.

possibly an extract from the judicial record; in the case of stadium attendance, a ticket, proof of age and proof of vaccination. Credentials can be divided into identity credentials and attestation credentials (see Fig. 2). Identity credentials can be used to prove the identity of a holder in a similar way to physical ID documents and passports. Attestation credentials simply verify the person presenting them and allow the authentication of attributes (European Commission 2022c).

The distinction between identity credentials and attestation credentials is relevant not only semantically but also in terms of the security requirements of the wallet. Credentials for identities are usually subject to high requirements for regulatory use cases, attestation credentials often do not require a high level of assurance. Ideally, a wallet for self-sovereign identities must therefore be able to support, communicate and serve different levels of assurance.

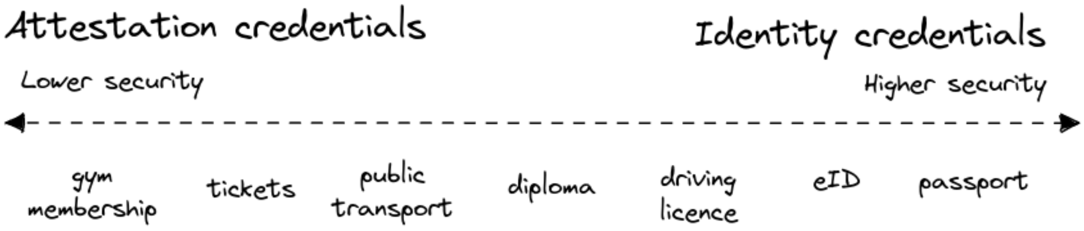

Fig. 2: Differentiation of credentials based on security requirements

3 Objectives

The security of a decentralised ecosystem is determined by its underlying trust relationships. A trust relationship consists of a trust provider and a trust recipient, for example, the verifier grants the issuer the trust to embody a real organisation and to issue only legitimate credentials or to revoke invalid credentials in a timely manner. The basis for trust may be either organisational or technical in nature, for example through cryptographically secured communication channels, standardisation and certification, through contracts regulated by governance or through regulatory and legally authorised trust anchors and accreditation bodies. All legitimate participants in an ecosystem have a justified interest in secure and trustworthy exchange of information. However, an open ecosystem also potentially enables the development of abusive wallets that could allow the extraction and sharing of credentials and their keys. This means the issuer and the verifier have a legitimate interest that the issued credentials are kept safe and protected from misuse, because otherwise the issuer risks their own credibility and the verifier bases its business processes on the assumption of verified data.

While much of the research on the security of identity ecosystems has focused on the trust relationships of the issuing process and the various technologies of verifiable data registries, developments relating to the holder and their wallet have mostly been neglected (Bastian et al. 2021). The reasons for this could be the relatively young state of research for open identity ecosystems and low security requirements in the first prototype use cases implemented. Since then, government-funded projects and pilot projects have brought decentralised identity into focus, creating new demands for regulated and highly secure use cases for verifiable credentials.

Our aim with this paper is to close the gap mentioned above. The object of our investigations was to assess how security concepts for wallets with identity credentials can be implemented using available smartphones and technologies. Solution concepts were supposed to consider the requirements of regulated applications and be designed and evaluated for the best possible coverage, scaling, security, user-friendliness and privacy.

The approach included intensive concept work within the framework of the funded joint project IDunion, working groups in the accompanying research project Secure Digital Identities as well as presentation and discussion in the Wallet Security Working Group of the Decentralised Identity Foundation (DIF). The work was also based on results gained from the SSI pilot project of the German Federal Chancellery.

Below, our investigation first analyses the requirements from various regulated applications. We then look at the framework conditions for the possible solution space and offer an overview of the available security-related components. We derive the essential building blocks of a security concept for the wallet from the requirements. This is followed by a discussion and evaluation of the solution concepts discussed. Finally, we provide an outlook on further research and work in this environment.

4 Requirements

Participants in an open identity ecosystem can design their own security concepts as long as they are not subject to legal requirements. International, national or state laws regulate specific technical requirements that identity solutions must fulfil, e.g. use cases related to finance (German Money Laundering Act), identity credentials (German Act on Identity Cards and Electronic Identification), telecommunications (German Telecommunications Act) and other official forms of evidence. This is why national regulators issue guidelines specifying minimum security standards for these laws. These guidelines then contain a set of security measures for different levels of assurance, for example NIST IAL and AAL (NIST 2020), eIDAS LoA (European Commission 2022a) or BSI TR-03107/03147 (BSI 2022a, 2022b). Typical evaluation factors for electronic identities include the issuing process with identity verification of the user and issuance security, the security of the means of authentication as multi-factor authentication with the factors possession, knowledge and inherence (biometrics), communication security, cryptographic algorithms and revocation mechanisms.

To support the security evaluation of technical solutions and their level of reliability, the threats and attack vectors are mapped and the potential of an attack on the authentication mechanism is analysed. These threat models are usually assessed according to the ISO 29115 standard (ISO/IEC JTC 1/SC 27 2013), which describes standardised attack vectors for an IT system:

- Online/offline guessing (repeatedly trying out the credentials or keys)
- Credential duplication (copy of credentials and their keys)
- Phishing (interception of credentials via fake websites/emails and social manipulation)
- Eavesdropping
- Replay attack (reuse of recorded messages)
- Session hijacking
- Man-in-the-middle attack (MitM; active attacker positions himself between the communication partners and pretends to be the respective counterpart)
- Credential theft
- Spoofing and masquerading

The targets to be assessed are then analysed against these attack potentials in terms of the expected time, the attacker's expertise and equipment, and the knowledge of the technical solution and window of opportunity, as described in ISO 18045(ISO/IEC JTC 1/SC 27 2022). Subsequently, these metrics are summarised and result in an attack potential that is assigned a certain level of assurance. The levels of assurance are assigned to an overall system and lifecycle around the identity, including the source of the identity data, i.e. the wallet is only part of the assessment and is not itself assigned to a particular level of assurance. Many attack vectors are covered by the architecture and protocols themselves, such as eavesdropping and replay attack.

However, the holder's wallet plays a key role in achieving a certain level of assurance, especially with regard to online/offline guessing, credential duplication, man-in-the-middle attacks and credential theft. In the following chapters, we will focus on describing solution concepts and security measures for these attack vectors that meet the requirements of a wallet in an open identity ecosystem.

5 General framework conditions

5.1 Wallet architecture

Smartphones and wearables are now the centre of many citizens' digital lives, making them a predestined platform for a user-centric SSI wallet. However, the specific technical architecture of a wallet offers many design options. The solutions are mainly characterised by the location of the credentials, the secure key management mechanism and user authentication. The spectrum of possible architectures ranges from native, mobile wallet app solutions (also called edge or non-custodial wallets) without additional backend services to fully cloud-based wallet services (also called managed or custodial wallets) that are not tied to a user device. Between these solutions, there are a variety of hybrid architectures that employ both user devices and backend services. The technical design has a decisive influence on the safety concept. It also affects such features as backup & recovery, locking mechanisms, offline support and privacy.

In our article, we focus exclusively on mobile wallet architecture. A native mobile wallet solution is the most decentralised approach to this architecture. Here, all the credentials and the keys are stored on the smartphone and user authentication also takes place exclusively locally with respect to the device. This solution is technically complex due to the fragmentation of the smartphone market and does not allow for backup mechanisms for device-bound credentials, but offers a high level of user control, good offline capabilities and privacy.

5.2 Usage scenarios

An important differentiation for the consideration of security concepts is the type and location of use or the associated communication channels. A distinction must be made, however, between on-site verification of credentials and remote verification via the Internet. The location of the verifier has a significant influence on the trust relationship between holder and verifier and in particular on the possibilities of user authentication.

In the case of on-site verification, these are classic identification processes equivalent to analogue ID documents: The verifier requests credentials and the holder submits a presentation of his credentials. Communication can take place offline by means of local connections such as NFC, Bluetooth or WiFi, as well as online, provided both the holder's wallet and that of the verifier are connected to the internet. Remote verification, on the other hand, always takes place online. Physical proximity to the verifier often enables a better trust relationship, while the remote scenario, especially together with biometrics, poses challenges for the security and privacy of a user authentication.

Both usage scenarios fulfil common and important use cases and must therefore be dealt with in the investigations.

5.3 Security systems in smartphones

The technical prerequisites for a secure SSI wallet are available on almost all mobile devices today. The high level of system fragmentation, especially in Android, makes a uniform solution difficult and makes device binding and user authentication technically complex. Fig. 3 shows the existing technologies in relation to their estimated distribution in the market and the level of assurance that can be achieved.

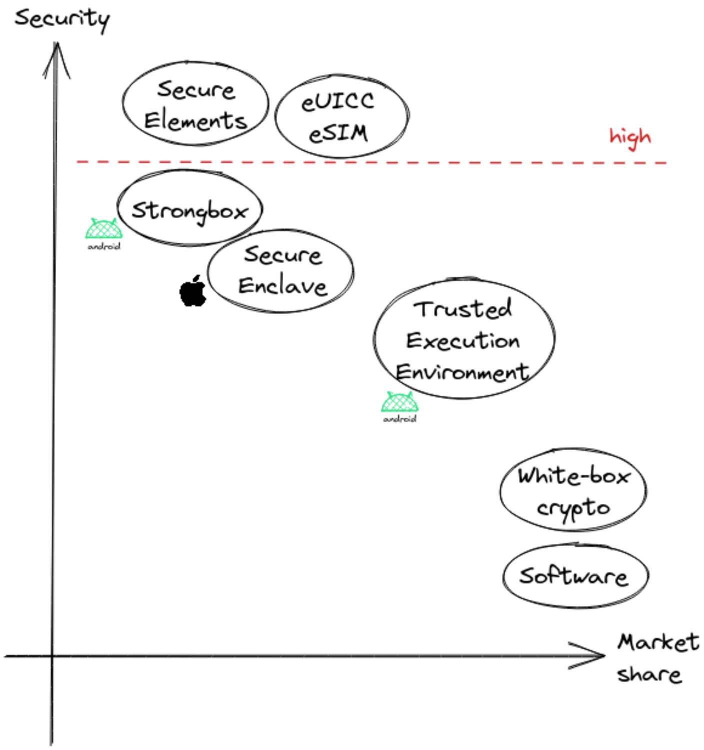

Fig. 3: Security technologies in smartphones

In a software-based approach, the keys are managed in the app's memory area and loaded by the main processor in normal RAM during cryptographic operations. Like white-box cryptography, in which the keys are algorithmically obfuscated in the source code, this method does not offer any significant protection against extraction and duplication (Sanfelix et al. 2015).

The Trusted Execution Environment (TEE) is a protected, hardware-supported processor environment that provides secure applications for mobile devices such as smartphones, tablets or smartwatches. The TEE is isolated from the main operating system (the rich OS execution environment) and runs on a separate operating system with its own memory area. In the smartphone market, TEEs have been mandatory since Android version 6 (Google 2022e), here the Arm TrustZone[2] technology in conjunction with the GlobalPlatform specification has become established and is offered by more than a dozen chip manufacturers. The TEE is coupled to a secure boot process and loads signed code in the form of trusted applets that secure their memory with chip-specific keys. Key management is organised in trusted applets within the TEE and provided to the wallet via Android's Keymaster API (Google 2022a). The API also provides a way to attest the key properties and location using certificate verification to (Google 2022b). Activation of the keys in the TEE can be linked to the system PIN and biometrics. Even though the TEE offers an increased level of security and is used today to secure payment transactions and DRM applications, there have been several successful exploits against TEEs on Android phones in the past (Sanfelix 2019; Cerdeira et al. 2020).

The secure enclave is a dedicated secure subsystem of the Apple ecosystem that is integrated as a system-on-a-chip (SOC) (Apple 2022c). The secure enclave has its own processor and accesses the main memory encrypted by a memory protection engine. It also has a secure random number generator, non-volatile memory and a hardware engine for AES and public keys. It is protected against manipulation and side-channel attacks and contains its own operating system called sepOS. The secure enclave was first installed in the A7 chip of the iPhone 5s and is now available in all Apple devices with iOS version 14. Apple uses the technology for Touch ID and Face ID as well as for Apple Pay; other future use cases include the Car Connectivity Consortium's mobile driving license and car keys. For signatures and encryption with asymmetric cryptography,

keys are only supported on the elliptic curve NIST P-256, which can only be generated on the device (no key import is possible), similarly, attestation of hardware-bound keys is as yet not possible (Apple 2022b). Secure enclave is currently certified to FIPS 140-2/3, with common criteria certification pending (Apple 2021). In the past, the secure enclave, as part of the Apple ecosystem, has been a target of exploits (Gian 2020).

The StrongBox is an implementation for the Android Keymaster from Android version 9 and is based on a separate, hardware-based security chip, Google refers to embedded Secure Elements (eSE) and on-SoC secure processing units (iSE) (Google 2022a). The first of these was an in-house development called "Titan M" for the Pixel 3 in 2018, and Google is using the "Titan M2", based on the open instruction set architecture RISC-V, for the second youngest generation Pixel 6. The requirements for a StrongBox are a dedicated processor, secure memory, a true random number generator and resistance to physical and logical attacks such as side-channel attacks. By analogy to the TEEs, the Android Keymaster supports various algorithms[3] and attestation of the keys from a strongbox. In March 2021, Google announced the Android Ready SE Alliance, a partnership with leading secure elements manufacturers to strengthen the security of the Android ecosystem and accelerate a move to strongbox-compatible hardware chips (Google 2021). This is to enable use cases for digital keys, driving licences, identities and electronic money. This will probably bring a security gain for the Android platform in the medium term, as the Strongbox was previously reserved for Pixel devices and thus could only cover a small share of the market. To date, there are no publicly known attacks on the Strongbox chips of the Titan family currently in use.

A secure element describes a tamper-proof, certified security chip with its own processor, RAM and hardware-supported co-processors for cryptographic operations. In addition, it has genuine random number generators to create keys and its own memory to store certificates, keys and other data worth protecting. Secure elements appear in different form factors and under different names depending on the area of application. As contactless or contact smart cards (chip cards), they enable numerous applications for credit cards, ID cards, health cards or PKI and access cards in the corporate environment. Secure elements can be inserted as UICC (SIM cards) or SD cards. The general trend in the mobile market is towards the integrated eUICC (embedded UICC) and eSE (embedded secure elements), which are managed by the telecommunications providers or the device manufacturers. The market is dominated by only a dozen incumbents, all using their own proprietary operating systems.

[2] TEE is a collective term that is also used for server-based security technologies such as Intel SGX and AMD Secure Processor.
[3] KeyMaster API offers RSA-2048, AES 128/256, ECDSA P-256, HMAC-SHA256, Triple DES 168.

Depending on the chip, a variety of cryptographic algorithms for encryption and signature with symmetric and asymmetric keys are supported. A majority of secure elements can also execute reloadable and interoperable code as a Java Card applet, which is interpreted via a Java Card runtime environment implemented on the chip (Chen 2000). The GlobalPlatform standard enables the administration and installation of Java Card applets; the keys for authorisation are usually held by the manufacturer. Java Card makes it possible to map one's own logic and protocols in a secure element, which offers more room for manoeuvring than the predefined APIs of the other security chips. Secure elements have been among the most secure systems available for more than 20 years. They owe this primarily to a very strict certification process according to common criteria, in which the hardware and software are approved according to standardised requirements by specialised, accredited test laboratories.

Apart from the key stores mentioned, there are also hardware security modules (HSM) for the server sector and trusted platform modules (TPM) for the desktop and notebook market, which, however, only play a subordinate role in mobile, native wallet architecture.

In summary, it can be said that market coverage decreases as security levels increase. TEEs as a basis for a medium security level exist on almost all devices. Secure elements, which are necessary to achieve a high level of security, are only installed on flagship mobile phones made by premium phone manufacturers. In the medium term, the eUICCs may also provide a remedy, but even with them, comprehensive market coverage is not possible at the present time. These challenges are also met by the Smart-eID, which is intended to bring the online ID function of official ID cards to the smartphone (BMI 2021).

6 Solution components

In order to implement multi-factor authentication as a means of identification and to address the required attack vectors, a mobile wallet should implement the following important core components:

- **Device binding** – binding credentials to the holder's device maps the required authentication factor of ownership. The use of hardware-bound keys prevents unauthorised copying and thus prevents credential duplication.
- **User binding** – the binding of credentials to the holder identified by the issuer by means of a PIN or biometrics – implements the second authentication factor (knowledge or inherence). A secure implementation for storing and verifying the credentials on the smartphone protects the means of identification against online/offline guessing and credential theft.
- **Wallet authentication** – proving the integrity and authenticity of the wallet to the issuer and verifier enables the issuer and verifier to trust that the wallet is managing credentials and user authentication according to secure, certified standards and has not been tampered with. This supports device and user binding and mitigates spoofing and masquerading attacks.
- **Trusted party** – the verifiability of the identities of the issuer and verifier through the wallet supports the holder in their self-sovereign decision-making and prevents man-in-the-middle attacks and makes potential phishing more difficult.

6.1 Device binding

The main task of the wallet is the management of the credentials and the associated cryptographic keys. While the signature of the issuer confirms the integrity and authenticity of the attributes contained in the credentials, the credentials themselves contain a further key in order to bind them to the wallet of the holder. The public part of the key can be stored in the credentials themselves, or in a DID document[4] referenced in the credentials or mapped through a zero knowledge proof secret. The private key is managed in the wallet of the holder. This is used to sign a presentation requested by the verifier. It is also this key pair that can be used to bind a device to the holder's smartphone.

[4] A DID (decentralised identifier) can be output according to the W3C DID core specification as a DID document that contains keys, endpoints or other data for this DID. This approach offers advantages for organisations and legal entities, such as key rotation and versioning. In contrast, the use of DIDs, especially using a verifiable data registry, is more critical for natural persons in their role as holders and poses risks in terms of potential traceability and additional complexities of the DID method used for key binding, so it is probably not suitable for regulatory applications.

There are basically two approaches to choose from for device binding: The *trusted storage* approach assumes that the wallet as a whole is operated in a secure environment that ensures integrity and authenticity, and that the transmitted credentials data is also trustworthy. This approach is used, for example, by the EAC protocol of the new German ID card. First, the authenticity of the verifier and the wallet is ensured by means of terminal and chip authentication and a secure channel is established. The requested data is then transmitted unsigned within the secure messaging procedure. While this approach offers good privacy properties, it requires the device to have a secure element to perform the protocol logic within the secure environment. The APIs of a strongbox, secure enclave or TEE do not provide this functionality.

The preferred solution for device binding is therefore the *cryptographic key binding* approach. Here, the credentials are bound to a secure key that lies within the secure environment, but the credentials themselves are managed in the (encrypted) memory of the wallet app. This approach is applied by all decentralised identity ecosystems today. The advantage of this method is that it is compatible with all mobile security environments presented and thus achieves a high market coverage. The disadvantage, however, is that the unlinkable, privacy-orientated signature algorithms based on CL (Camenisch und Lysyanskaya 2004; Khovratovich und Lodder 2018; Curran et al. 2022) and BBS+ (Boneh et al. 2004; Looker und Steele 2021) with selective disclosure are not supported by any hardware available today and will not be rolled out across the board in the foreseeable future either.[5] Instead, a selection of the most widely used algorithms makes sense in order to keep the implementation effort low and also to guarantee cryptographic agility[6]. As RSA is viewed increasingly sceptically by regulators and is not available on the secure enclave, the most promising solutions for this are the following[7]:

- ECDSA/SHA256 with NIST P-256/secp256r1
- ECDSA/SHA256 on BrainpoolP256r1

On this basis, a simple challenge-response scheme is used to authenticate the credentials. The security mechanisms available on the respective terminal should be fully applied and the type of security chip used should be indicated in the credentials themselves. Optionally, it is also possible to include the achievable level of assurance in the credentials themselves.

6.2 User binding

User binding in a native mobile wallet takes place via the second factor of knowledge (PIN/password) or inherence (biometrics). Authentication of the holder is a core requirement for identity credentials, while attestation credentials do not usually need to authenticate the user.

In the on-site application, user binding can be achieved relatively easily by the verifier performing a manual identification of the holder by matching the photograph or other biometric data verified by the issuer and stored as attributes in the credentials. It should be noted that the transmission of verified and signed biometric data poses a certain risk to privacy. Due to the physical proximity, the person using the wallet must be able to rely on the fact that the verifier is directly in front of him and that no relay or phishing attacks are possible. This trust relationship can be supported via an offline communication protocol with limited range or via local initiation of communication, for example by means of a QR code. In addition, the verifier (see Chapter 6.4) should be authenticated in such a way that the holder can check this in their wallet.

In contrast, user binding in the remote use case, especially by means of biometric authentication, entails considerable technical complexity and poses risks for data misuse and privacy. Depending on the use case, a wallet must itself authenticate the holder or leave it to a trusted verifier. The preferred user-centric and privacy-orientated approach for the remote use case stores the reference data of the holder only in the wallet, similarly, matching only takes place locally on the device, a technical solution authorises the use of the (device-bound) ownership factor through successful user authentication. This architecture simultaneously requires a relationship of trust with the wallet as a means of identification, since the issuer and verifier require an assurance of local user binding. The resulting verifiable authenticity of the wallet is described in the following chapter. First of all, the user binding actions are carried out.

As an established procedure, the use of PINs and passwords is still widespread today and is also the backup mechanism when unlocking a smartphone with biometrics. For an application in a regulated environment, the length and the alphabet of the

[5] An academic study where Java Cards were tested found them to be insufficient in terms of performance (Bichsel et al. 2009).

[6] Cryptographic agility enables rapid responses when systems have become insecure by implementing and using alternative methods of encryption, verification and authentication.

[7] Unfortunately, EdDSA is also not supported by many chips.

characters used in the PIN, or the password, and the resulting combination space together with the retry counter are decisive. In addition, the implementation and storage location of the reference values and the retry counter, as appropriate, must meet the requirements of the level of assurance. Both iOS and Android offer the system PIN as an authentication means, which can also be linked to the hardware-bound keys from TEE, strongbox and secure enclave. It should be noted, however, that the wallet app cannot specify a minimum PIN strength and there is no other mechanism to determine the properties. As a result, under Android, the system PIN can be a swipe pattern, a PIN or a password. Another option is to implement a separate PIN that is managed by the wallet app. A PIN policy can be set by the user and the retry counter can be stored in the secure memory of the smartphone. The disadvantage with this variant is that the PIN logic is performed in software, which is also not sufficient for a high level of assurance. Finally, there is the possibility of storing the PIN in a secure element, so authorisation of a device-bound key from the secure element can also be easily linked; the only disadvantage is the market share of this solution. Similar to device binding, smartphones offer different approaches that shall be used and reflected in the level of assurance of the credentials.

Biometrics enables user authentication based on behavioural and physical characteristics such as fingerprints, face, iris, voice or gait and offers the convenience of easy authentication as the user does not have to remember PINs or passwords. Biometric sensors for biometric modalities fingerprint and face recognition can be found on almost all smartphones today, and the trend towards increasing use has been constantly increasing for years. In order to systematise the requirements for a level of assurance of biometric authentication and to make the technical solutions comparable, definitions for biometric performance values as well as a presentation attack detection are necessary. The BSI has published the "Technical Guideline for Biometric Authentication Components in Devices for Authentication" (BSI 2021) on this topic, which provides specifications and recommendations regarding biometrics. However, the various implementations by the manufacturers have been regularly attacked in recent years and can be overcome with mostly trivial methods. Virtually all smartphones are therefore inadequate for regulated use cases according to the state of the art, with Apple's Face ID being one of the few exceptions.

6.3 Wallet authentication

To prove authenticity, the wallet must provide two forms of credentials:

- Attestation of the keys for device binding
- Attestation of the wallet as a means of identification for user binding

Attestation of cryptographic keys proves that the private keys were actually generated within the hardware unit. For this purpose, there is an integrated mechanism under Android for TEE and Strongbox by means of Keymaster, which makes the key attestation verifiable as an X.509 certificate chain. Under iOS, there is no such mechanism for the secure enclave. For implementations using secure elements, attestation is possible in principle, but uniform interfaces and protocols are still not available.

An attestation of the wallet is intended to prove that the app is indeed the authentic, unmanipulated and certified version of a wallet manufacturer. Since the app runs on the smartphone's "insecure" operating system, this security level represents the best possible mitigation approach. The wallet app must implement state-of-the-art measures against emulation, debugging, code injection, cloning and other attacks. In addition, it should prove its authenticity. The best way of doing this is the APIs provided by the platform. With the Play Integrity API (Google 2022c) (and its predecessor SafetyNet Attestations API (Google 2022d)), Android offers an interface for integrity assessment and authentication of the app, which checks the software and hardware environment for manipulations. Verification is linked to a nonce that the app sends to a Google server together with current status data of the smartphone. This server evaluates the data and generates a cryptographically signed attestation, which the app forwards to the backend of the wallet manufacturer where it is evaluated. Apple offers a similar interface for its ecosystem called iOS DeviceCheck (Apple 2022a). Its primary function is the recognition of devices, so that the function can only determine authenticity and does not make any statements about the integrity of the smartphone. However, iOS as a closed Apple ecosystem basically offers a higher level of security than the open Android platform, so an additional integrity check is not absolutely necessary.

Both mechanisms are only a form of support for the authenticity check of the app and have to be complemented with other measures. One of the biggest challenges when proving the authenticity of the wallet is to prove to verifiers that it cannot be correlated.

In order to meet the regulatory requirements, certification of the means of identification is usually also necessary. The software and development processes are audited and confirmed by independent auditors. The certifications of various approved wallets in an identity ecosystem should then be compiled in a trust registry or trusted list, for example a verifiable data registry. This endpoint then serves as a contact point for the issuer and verifier to determine the trustworthiness of a wallet identified via wallet authentication.

6.4 Trustworthy party

In practice, users need security when dealing with the wallet, especially through clearly recognisable mechanisms when authenticating an issuer or a requesting verifier.

The solution concept is to display the identity and trustworthiness of the requesting party to the user in an easily understandable form and to provide the tools for a self-sovereign decision. The heterogeneous applications in a decentralised identity ecosystem require different levels of assurance and authentication. The applications range from self-sovereign direct presentation of credentials to each other without authentication of the requesting party to confirmation of the trustworthiness of the requesting party through the use of trust registries. These differences should also be reflected in the possibilities for identifying the requesting party. Dividing the authentication procedure into two or more levels therefore makes sense.

It should also be possible to share data in a self-sovereign identity ecosystem with requesters who cannot sufficiently prove their identity (for example, private individuals); in the process, they will be marked with clear warnings (Lissi 2022).

Authentication by means of EV certificates (extended validation) or QWACs (qualified website authentication certificates) and the display of the name of the requesting company is advantageous. Extended validation certificates serve as X.509 TLS certificates, for which the issuance to organisations is bound to strong criteria for the identification by the issuer. QWACs are qualified certificates for website authentication (eIDAS Art. 3 (39)). Issued by European qualified trust service providers (QTSP), they also provide an in-depth check of the organisation’s identity.

In addition, registration of the requesting party in a trust registy is possible. These registers can apply to the entire ecosystem or to specific applications and domains. A wallet may only release presentations to verifiers who have verified them using these trusted lists. The trust anchor can be implemented in the verifiable data registry or in existing lists such as the EU Trusted List.

7 Solution concept

The proposed process flow for securing a mobile native wallet combines the requirements with the technical possibilities and their limitations in a meaningful way. This integrates the security building blocks for user binding, device binding, wallet authentication and trusted party. The process flow is shown as a sequence diagram in Fig. 4 and includes six entities: the issuer who wishes to issue credentials for a regulated use case; the holder and their wallet; the verifier who wishes to verify the credentials and must check the regulatory requirements; an attestation service that performs the attestation of the wallet (operated by the wallet provider or by a trust service provider), and a trust registry that provides a trust anchor and registry for certified wallets.

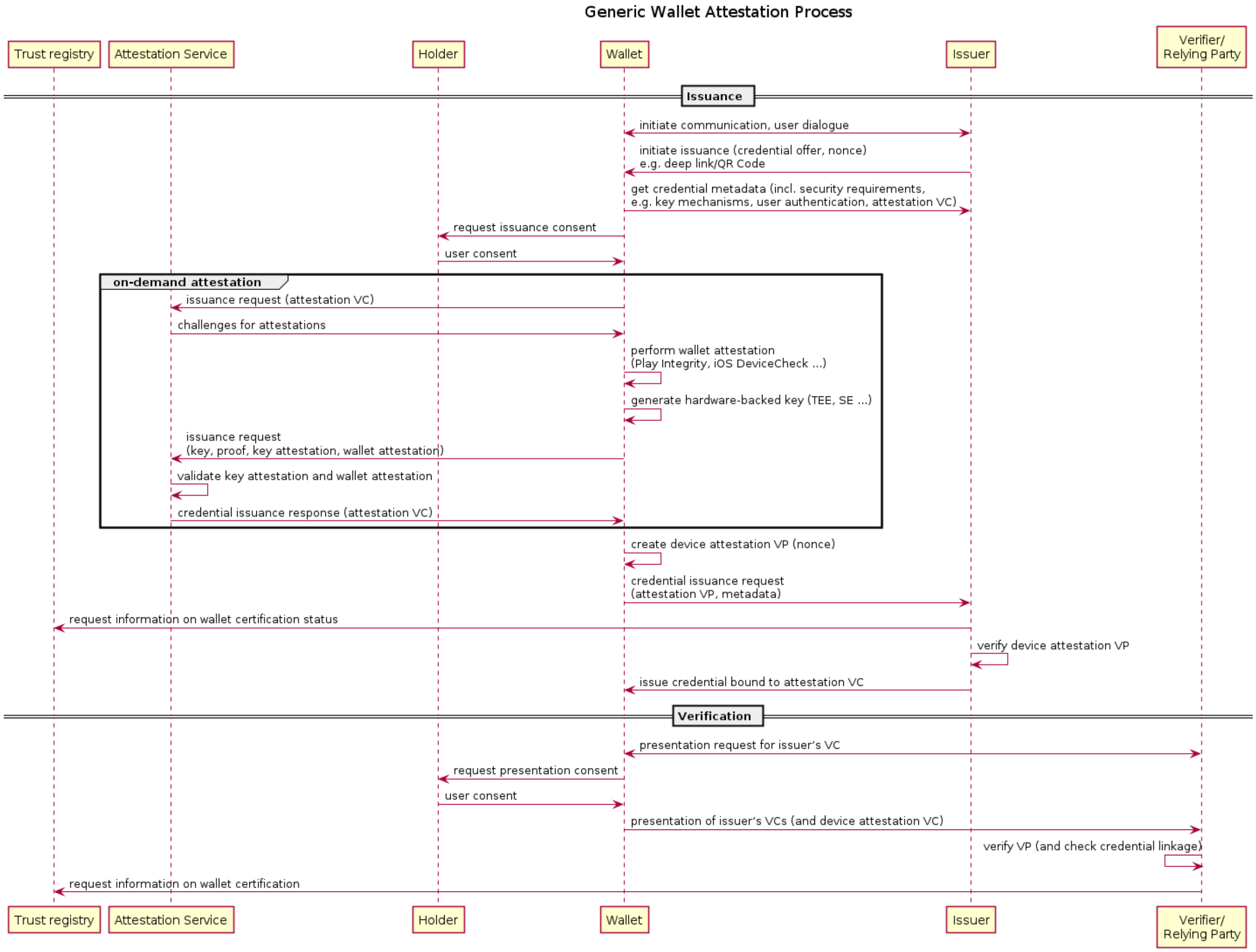
Generic Wallet Attestation Process
Trust registry
Attestation Service
Holder
Wallet
Issuer
Verifier/ Relying Party
Issuance
initiate communication, user dialogue
initiate issuance (credential offer, nonce) e.g. deep link/QR Code
get credential metadata (incl. security requirements, e.g. key mechanisms, user authentication, attestation VC)
request issuance consent
user consent
on-demand attestation
issuance request (attestation VC)
challenges for attestations
perform wallet attestation (Play Integrity, iOS DeviceCheck ...)
generate hardware-backed key (TEE, SE ...)
issuance request (key, proof, key attestation, wallet attestation)
validate key attestation and wallet attestation
credential issuance response (attestation VC)
create device attestation VP (nonce)
credential issuance request (attestation VP, metadata)
request information on wallet certification status
verify device attestation VP
issue credential bound to attestation VC
Verification
presentation request for issuer's VC
request presentation consent
user consent
presentation of issuer's VCs (and device attestation VC)
verify VP (and check credential linkage)
request information on wallet certification
Trust registry
Attestation Service
Holder
Wallet
Issuer
Verifier/ Relying Party

Fig. 4: Sequence diagram for the process flow for securing a mobile wallet

7.1 Issuance

Issuance begins with the establishment of a communication channel between the issuer and the wallet. The issuer identifies the user[8] and then offers the issuance as a credentials offer. The wallet now retrieves the security requirements from the issuer's publicly accessible metadata. Here, the issuer specifies the requirements it places on the wallet in terms of device binding and user binding. In this way, the wallet can check on its own whether it can meet the regulatory requirements. It then requests the holder 's consent to proceed with the issuance of the credentials.

The wallet then requests an attestation from its attestation service. The attestation service is a back-end service that is operated by the manufacturer and operator of the wallet itself or is outsourced to a trust service provider, for example. Its task is to validate and confirm the device binding and wallet authentication of a wallet instance. The attestation of the wallet and the hardware-bound keys take place on demand in response to an issuance request. The advantages of ad-hoc issuance lie in the increased security (freshness of the attestation, reduced time window for potential attacks) and in the scaling of the system (only as many attestations are carried out as are actually needed). The credential format is proposed as a W3C verifiable credentials, so that the hardware key itself can be used as a proof mechanism and no additional protocol logic is required to validate the attestations (as opposed to Anoncreds).

The attestation service and the wallet establish a secure channel[9] and the wallet transmits its request. The attestation service generates and transmits the challenges necessary for the attestation and the wallet proves its authenticity and integrity using

[8] It is possible to make a preference for identity credentials to be derived from government documents.
[9] Supported, for example, by certificate spinning and API keys.

the Android Integrity API, iOS DeviceCheck or similar mechanisms. It then generates a hardware-bound key and transmits the public key together with the key attestation. The authorisation of the private keys is configured by the wallet according to the specifications of the issuer. The attestation service verifies all the data and issues a device attestation credentials to the wallet at the end of the process. This contains the validated information gathered from device binding, user binding and wallet authentication:

- Identity of the attestation service (issuer)
- Name and version of the wallet
- Public key of the device binding and signature protocols used
- Hardware type used (TEE, Strongbox, SE, eUICC ...)
- User binding mechanism (Face ID, system PIN, app PIN with four digits, SE PIN with six digits ...)
- Date of issue and optionally expiry date

As an option, the attestation service can also link directly to an accessible level of assurance. However, since the attestation service does not know the use case and a variety of different levels of assurance exist, this approach is not necessarily useful. Here is an example in JSON-LD W3C VC:

```
{
  "@context": [
    https://www.w3.org/2018/credentials/v1",
    https://www.w3.org/2022/credentials/walletAttestation/v1"
  ],
  "id": "http://attestationservice.com/id/826529163",
  "credentialSchema": {
    "id": "https://www.schema.org/examples/deviceAttestation.json",
    "type": "WalletAttestationSchema"
  },
  "type": [
    "VerifiableCredential",
    "WalletAttestationCredential"
  ],
  "issuer": "https://attestationservice.com/issuers/14",
  "issuanceDate": "2021-09-11T16:02:04Z",
  "expirationDate": "2021-12-11T16:02:04Z",
  "credentialSubject": {
    "walletName": "Example Wallet",
    "walletVersion": "1.0.0",
    "hardwareType": "SE",
    "hardwarePublicKey": " jwk:123456...",
    "holderAuthentication": [ "SE-PIN_6digit" ]
  }
}
```

The wallet can now respond to the credential offer of the issuer with a credential request which includes a (W3C verifiable) presentation of the device attestation credential, which thus proves possession of the hardware key (and therefore indirectly confirms user authentication). The issuer asks the trust registry for the status of the certification using the unique name[10] of the authenticated wallet and can thus ensure that regulatory requirements are met. Finally, the issuer issues the credential to the wallet (and links them to the device attestation credential). In the context of W3C verifiable credentials, this mechanism is discussed under the term holder binding (Curran, Stephen 2021; Bastian et al. 2023) .There are two options available to the issuer: In the preferred variant, they can link to the existing device attestation credentials and specify, for example, the issuer and ID. Alternatively, the issuer can integrate the hardware-bound key into their credentials and optionally copy additional parts of the attestation.

[10] The identifiers provided by Android IntegrityAPI and iOS DeviceCheck are unique and protected by the app manufacturer's certificates and keys.

7.2 Verification

The verification process starts analogously with the establishment of a secure communication channel and the verifier transmits its request for the requested credentials and the necessary level of assurance. The level of assurance should be anchored explicitly in the credentials or implicitly in the trust registry so that the wallet can match the verifier's requirements before presentation, as otherwise the holder might submit credentials and the verifier may find out only after validation that the credentials are inadequate for its regulatory processes, which would be detrimental to the holder's user experience. The wallet then generates two verifiable presentations, one for the requested credentials of the issuer (W3C VC, AnonCreds or some other format) and one for the device attestation credentials, which uses the hardware-bound key on the smartphone to create a presentation. [11] The wallet shows the holder the data to be transmitted and the identity of the verifier (see Chapter 6.4) and obtains the authorisation of the holder through the user authentication mechanism linked to the device binding. The wallet may, at this point, provide an indication that an additional traceable attribute is disclosed by the unique, hardware-bound public key. The two-part presentation is now transmitted to the verifier, which checks the two signatures and validates the linking of the credentials. The verifier can query the trust registry for the certification of the wallet and thus confirm the level of assurance of the credentials.

The assurance model in this system, shown in Fig. 5, traces back the issuance chain: The verifier trusts the issuer to have properly validated and linked the device attestation credentials in the issuance process, the issuer trusts the attestation service to have properly verified the device and that the details of the device attestation credentials are correct, and the attestation service trusts the certification and auditing of the wallet. Embedding the attestations themselves in the credentials is redundant and would create additional data protection problems.

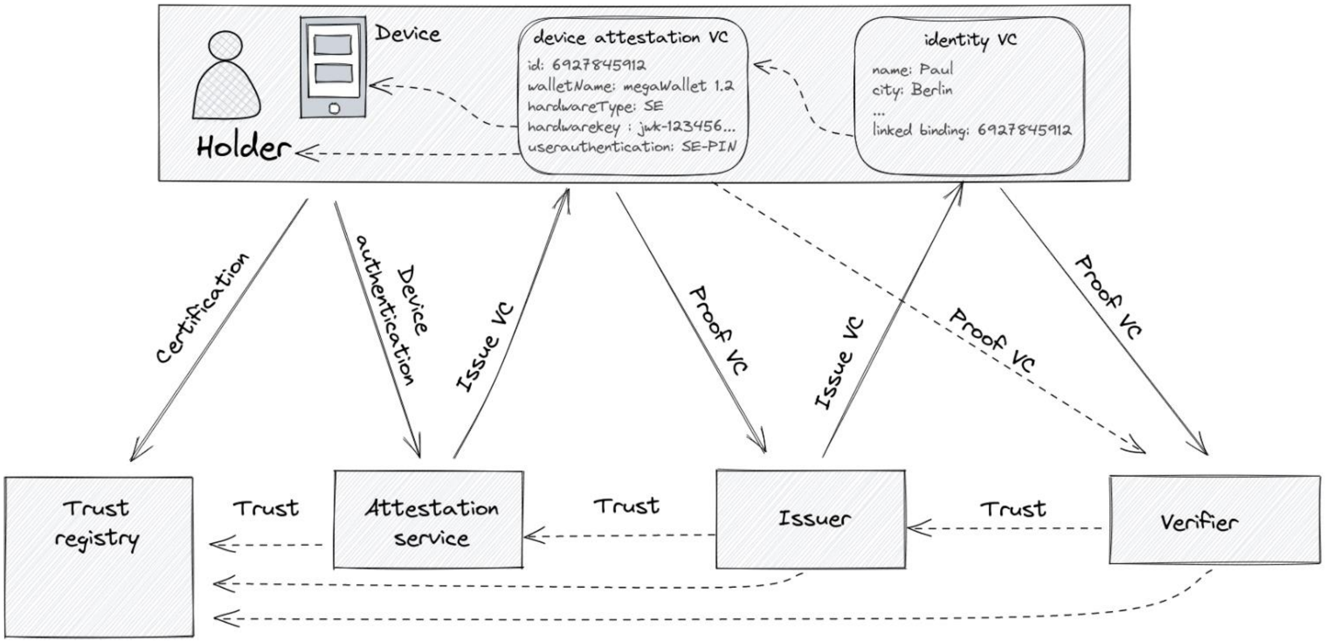

Fig. 5: Sequence diagram for securing a mobile wallet

8 Discussion and evaluation

The components and processes presented in our concept for securing a mobile wallet form the basis for real-life use of identity credentials for regulated applications.

Device binding using cryptographic key binding offers a simple and scalable solution for securing credentials against unauthorised copying. The security level achievable is based on existing technologies on modern devices and is therefore

[11] If the issuer does not link the device attestation but integrates it into its own credentials, the verifier does not need to have the device attestation credentials presented.

universally applicable. However, the use of hardware-bound keys in the credentials creates a potential privacy risk, as they are intrinsically unique and thus allow for additional traceability and correlatability. To mitigate this problem, a separate, hardware-bound key should be generated for each instance of device-bound credentials. Further mitigation results from binding to multiple one-time-use keys or transient, automatically renewed credentials. An implicit consequence of being device-bound to a smartphone is the expense of replacing or losing the device, as device-bound credentials cannot be backed up. In the future, research must continue on the integration of signatures for zero-knowledge proofs. Similarly, the design must today take into account cryptoagility for switching to post-quantum secure procedures[12].

User binding for high levels of assurance can still best be realised today via PINs and passwords. The security solutions on smartphones sometimes restrict the communication of more precise information about the strength of user secrets. To maximise user experience, applications with low and medium requirements should preferably enable biometrics. Biometrics can serve as an additional authentication factor and should be further investigated on an ongoing basis, as user-friendliness is crucial for the success of the solutions.

Wallet authentication combined with software certification provides a trusted platform for a decentralised identity ecosystem. These measures provide a high level of security that is otherwise difficult to achieve on mobile platforms. To ensure scaling, explicit verification of the wallet only takes place at the issuance stage. The concept ties in with Apple and Google services, which create a certain dependency and restrictions for rooted devices, similar to banking apps, for example.

The authentication of the requesting party is ensured via a step concept and uses established scaling processes. They enable trustworthy communication between the issuer and verifier and give the holder the tools to prevent phishing and MitM attacks and make a self-sovereign decision. Restrictions to minimum authentication requirements for the requesting party, for example in the case of medical details, are reasonable and within the discretion of the respective applications.

The concept describes how the components are brought together in a process flow. The attestation service plays an important role here. This service takes on the task of complex attestation checks and facilitates the implementation of security-critical applications for the issuer and verifier. It also enables scaling attestation of the wallet in a uniform format and takes into account the legal framework of the attestation APIs. As the wallet is the focus of the communication, the attestation service does not obtain any information about what the device attestation credentials are used for. The process works with multiple communication protocols such as DIDComm and OpenID4VC and supports various privacy-orientated credentials formats such as W3C with BBS+ or SD-JWT and AnonCreds.

The solution approach presented here supports multiple levels of assurance and applies the high regulatory requirements only to the most important identity credentials (with the resulting disadvantages), while all other credentials experience the advantages of biometric authentication, convenient backup and privacy-enhancing signature procedures.

The solutions have already been successfully tested in cooperation between the Bundesdruckerei and Lissi using DIDComm, and a demonstration using OpenID4VC is currently being implemented.

9 Outlook

Decentralised identity ecosystems will be an important pillar of digitalisation in the near future and promising implementation concepts for secure wallets will play a central role in this.

Identity credentials require a high level of protection, especially for regulatory use cases, which makes the measures described necessary. These measures can be successfully implemented on mobile devices even today using the solutions and processes shown here. The trend towards an increasing number of secure elements makes this path attractive for the future. The proposed measures can also be applied to other credentials. In the future hybrid architectures may also complement wallet security solutions, depending on requirements.

At the same time, other topics in the context of wallet security are the subject of current investigations and developments, such as linking multiple presented credentials, the privacy of wallet authentication and revocation, biometrics, user-friendly

[12] Quantum computers could break widely used cryptographic algorithms and become a serious threat to IT systems within the current decade. Intensive work is already being done on post-quantum cryptography (PQC) to secure classic security systems against attacks from quantum computers.

processes, post-quantum security, backup mechanisms and the interaction of different identity protocols and credentials formats (Bastian 2022).

International and European efforts are underway as part of the eIDAS revision and the community-driven developments, accelerated by open standards, offer scope for further collaborative design of privacy-friendly, interoperable and secure implementations for digital identities.

10 Acknowledgement

This publication is part of the funded joint project: IDunion – Building a decentralised identity ecosystem (funding code: 01MN21002L). The IDunion joint project was funded within the innovation competition "Showcase for Secure Digital Identities" by the German Federal Ministry of Economics and Climate Protection (BMWK) via the German Aerospace Centre (DLR project sponsor) based on a resolution of the German Bundestag.

We would like to thank the BMWK for its support and all the contributing partners from IDunion for the stimulating discussions and valuable contributions.

11 Bibliography

Apple (2021) Secure Enclave Processor security certifications (SEP). Apple Support. https://support.apple.com/en-gb/guide/certifications/apc3a7433eb89/1/web/1.0 Accessed on 22 September 2022

Apple (2022a) DeviceCheck | Apple Developer Documentation. https://developer.apple.com/documentation/devicecheck. Accessed on 23 September 2022

Apple (2022b) Protecting keys with the Secure Enclave | Apple Developer Documentation. https://developer.apple.com/documentation/security/certificate_key_and_trust_services/keys/protecting_keys_with_the_secure_enclave. Accessed on 22 September 2022

Apple (2022c) Secure Enclave. Apple Support. https://support.apple.com/de-de/guide/security/sec59b0b31ff/web. Accessed on 22 September 2022

Bastian P (2022) Credential Format Comparison. https://www.linkedin.com/posts/idunion_credential-format-comparison-and-idunion-activity-7008024119598276609-0pS-/. Accessed on 6 January 2023

Bastian P, Terbu O, Joosten R, Rivai Z et al (2023) Identifier Binding: defining the Core of Holder Binding. https://github.com/WebOfTrustInfo/rwot11-the-hague/blob/master/final-documents/identifier-binding.md. Accessed on 6 February 2023

Bastian P, Kraus M, Fischer J, Bösch C (2021) Self-Sovereign Identity – Vertrauensbasis für selbstbestimmte Identitätsnetzwerke (Foundation of trust for self-sovereign identity networks). 17th German IT Security Conference by BSI

Bichsel P, Camenisch J, Groß T, Shoup V (2009) Anonymous credentials on a standard java card. In Proceedings of the 16th ACM conference on Computer and communications security, CCS '09, 600–610. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1653662.1653734. Accessed on 22 September 2022

BMI (2021) Online-Ausweis kann bald im Smartphone gespeichert werden. (Online ID card soon to be available as a smartphone app) German Federal Ministry of the Interior and Community https://www.bmi.bund.de/SharedDocs/pressemitteilungen/DE/2021/09/smart-eID-gesetz-in-kraft.html?nn=9390260. Accessed on 22 September 2022

Boneh D, Boyen X, Shacham H (2004) Short Group Signatures. In Advances in Cryptology – CRYPTO 2004, Lecture Notes in Computer Science, Pub. Matt Franklin, 41–55. Berlin, Heidelberg: Springer

BSI (2021) TR-03166 Technical Guideline for Biometric Authentication Components in Devices for Authentication. German Federal Office for Information Security. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03166/BSI-TR-03166.html?nn=132646. Accessed on 22 September 2022

BSI (2022a) TR-03107 Elektronische Identitäten und Vertrauensdienste im E-Government. (Digital identities and trust services in eGovernment) German Federal Office for Information Security. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03107/tr-03107.html?nn=450536. Accessed on 22 September 2022

BSI (2022b) TR-03147 Vertrauensniveaubewertung von Verfahren zur Identitätsprüfung natürlicher Personen. (Assurance level assessment in processes for checking the identity of people) German Federal Office for Information Security. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03147/identitaetspruefung.html?nn=130418. Accessed on 22 September 2022

Camenisch J, Lysyanskaya A (2004). Signature Schemes and Anonymous Credentials from Bilinear Maps. In Advances in Cryptology – CRYPTO 2004, Lecture Notes in Computer Science, Pub. Matt Franklin, 56–72. Berlin, Heidelberg: Springer

Cerdeira D, Santos N, Fonseca P, Pinto S (2020) Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems. In 2020 IEEE Symposium on Security and Privacy (SP), 1416–1432

Chen, Z (2000) Java card technology for smart cards: architecture and programmer's guide. Addison-Wesley Professional

Curran S (2021) W3C VC Data Model Github Issue#789. GitHub. https://github.com/w3c/vc-data-model/issues/902. Accessed on 6 January 2023

Curran S, Yildiz H, Curren S (2022) AnonCreds Specification. AnonCreds WG

European Commission (2022a) eIDAS Levels of Assurance. Digital https://ec.europa.eu/cefdigital/wiki/cefdigital/wiki/display/CEFDIGITAL/eIDAS+Levels+of+Assurance. Accessed on 23 September 2022

European Commission (2022b) eIDAS Regulation | Shaping Europe’s digital future. https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation. Accessed on 23 September 2022

European Commission (2022c) European Digital Identity Architecture and Reference Framework – Outline | Shaping Europe’s digital future. https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-architecture-and-reference-framework-outline. Accessed on 22 September 2022

Gian (2020) Code Execution achieved in the Secure Enclave chip. Yalu Jailbreak. https://yalujailbreak.net/seprom-code-execution/. Accessed on 22 September 2022

Google (2021) Google Online Security Blog: Announcing the Android Ready SE Alliance. https://security.googleblog.com/2021/03/announcing-android-ready-se-alliance.html. Accessed on 22 September 2022

Google (2022a) Android keystore system – Strongbox. Android Developers. https://developer.android.com/training/articles/keystore#HardwareSecurityModule. Accessed on 22 September 2022

Google (2022b) Key and ID Attestation. Android Open Source Project. https://source.android.com/docs/security/features/keystore/attestation. Accessed on 22 September 2022

Google (2022c) Overview of the Play Integrity API | Google Play. Android Developers. https://developer.android.com/google/play/integrity/overview. Accessed on 22 September 2022

Google (2022d) SafetyNet Attestation API. Android Developers. https://developer.android.com/training/safetynet/attestation. Accessed on 22 September 2022

Google (2022e) Security Enhancements in Android 6.0. Android Open Source Project. https://source.android.com/docs/security/enhancements/enhancements60. Accessed on 22 September 2022

ISO/IEC JTC 1/SC 27 (2013) ISO/IEC 29115:2013. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/51/45138.html. Accessed on 22 September 2022

ISO/IEC JTC 1/SC 27 (2021) ISO/IEC 18013-5:2021. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/90/69084.html. Accessed on 22 September 2022

ISO/IEC JTC 1/SC 27 (2022) ISO/IEC 18045:2022. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/28/72889.html. Accessed on 22 September 2022

Khovratovich D, Lodder M (2018) Anonymous credentials with type-3 revocation

Lissi (2022) Vertrauen im digitalen Raum. (Trust in a digital space) Medium. https://lissi-id.medium.com/vertrauen-im-digitalen-raum-cc22a9fcbd0a. Accessed on 6 January 2023

Looker T, Steele O (2021) BBS+ signatures 2020. W3C Credentials Community Group

Sporny M, Longley D, Chadwick D (2022) Verifiable Credentials Data Model v1.1. https://www.w3.org/TR/vc-data-model/. Accessed on 22 September 2022

NIST (2020) NIST Special Publication 800-63-3. https://pages.nist.gov/sp800-63-3.html. Accessed on 22 September 2022

Sanfelix E (2019) TEE Exploitation-Exploiting Trusted Apps on Samsung’s TEE. Accessed on 3 July 2020

Sanfelix E, Mune C, Job de Haas (2015) Practical attacks against Obfuscated Ciphers

Strüker J et al. (2021) Self-Sovereign Identity: Grundlagen, Anwendungen und Potenziale portabler digitaler Identitäten. (Fundamentals, application and potentials of portable digital identities) Sankt Augustin. https://www.fim-rc.de/wp-content/uploads/2021/06/Fraunhofer-FIT_SSI_Whitepaper.pdf. Accessed on 22 September 2022

Young K (2021) Verifiable Credentials Flavors Explained