applied sciences

MDPI

Article

Privacy-Preserving Solution for European Union Digital Vaccine Certificates

Petr Dzurenda [1,*], Sara Ricci [1], Petr Ilgner [1], Lukas Malina [1] and Carles Anglès-Tafalla [2]

[1] Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technicka 12, 61600 Brno, Czech Republic; ricci@vut.cz (S.R.); ilgnerp@vut.cz (P.I.); malina@vut.cz (L.M.)
[2] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Avinguda Països Catalans 26, 43007 Tarragona, Spain; carles.angles@urv.cat
* Correspondence: dzurenda@vut.cz; Tel.: +420-541-146-925

Abstract: The recent COVID-19 pandemic situation highlights the importance of digital vaccine certificates. In response, the European Union (EU) developed EU Digital Vaccine Certificates to enable proof of non-infectivity and completed vaccinations. However, these solutions suffer from several shortcomings, such as ineffective certificate holder identification and a high violation of user privacy with the disclosure of sensitive information. In this work, we present a novel solution for privacy-preserving EU Digital Vaccine Certificates. Our solution solves the aforementioned privacy and security shortcomings and is in line with current EU legislation, i.e., the General Data Protection Regulation (GDPR), the upcoming revision of the electronic IDentification, Authentication, and trust Services (eIDAS), called regulation eIDAS 2.0, and the new tools that it envisages to be led by European digital identity. This identity is intended to allow citizens to prove their identity to access online services, share digital documents, or simply prove specific personal characteristics such as age without revealing their identity or other personal information. The core of our proposal is built on our novel attribute-based credential scheme, which can be easily implemented on various handheld devices, especially on Android smartphones and smartwatches. However, due to the lightweight nature of our scheme, it can also be implemented on constrained devices such as smart cards. In order to demonstrate the security, privacy, and practicality inherent in our proposal, we provide the security analysis of the cryptographic core along with a set of experimental results conducted on smartphones and smart cards.

Keywords: digital vaccine certificate; COVID-19; attribute-based credential; authentication; cryptography; security; privacy; Android; Bluetooth Low Energy; smart cards

Citation: Dzurenda, P.; Ricci, S.; Ilgner, P.; Malina, L.; Anglès-Tafalla, C. Privacy-Preserving Solution for European Union Digital Vaccine Certificates. *Appl. Sci.* **2023**, *13*, 10986. https://doi.org/10.3390/app131910986

Academic Editor: Giacomo Fiumara

Received: 9 September 2023
Revised: 27 September 2023
Accepted: 3 October 2023
Published: 5 October 2023

1. Introduction

The COVID-19 pandemic era has significantly accelerated cooperation and development of e-healthcare. Various experts, researchers, medics, technicians, and society as a whole have to promptly design and develop many efficient tools that help to beat and mitigate this worldwide disease. Since 2020, several digital approaches, tools, and applications have been proposed and deployed. These technologies and services usually aid in digital contact/contactless tracing, infection tracking, patient monitoring, checking immunity, or injecting vaccination doses. Examples of such applications include COVID-19 certificates. On the one hand, these solutions have to be functional, robust, and resilient across borders; secure; and immune to their misusing. In fact, keeping users' privacy is an essential requirement for society. A way to achieve the aforementioned properties is by deploying Privacy-Enhancing Technologies (PETs) and, more specifically, attribute-based authentication schemes.

Attribute-based authentication schemes help preserve user privacy by verifying only necessary pieces of a user's private information. In comparison with classic authentication

schemes, these schemes verify only the possession of specific attributes, e.g., age, driving license, authorization to access a protected area, type of traffic ticket, or possession of a COVID-19 certificate. Attribute-based schemes do not require the verification of specific user identities, such as IDs, user public keys, and shared passwords, that could cause unwanted linking and tracking of individuals.

Despite the fact that global pandemic situation is slowly weakening in the world, secure and privacy-preserving COVID-19 or non-infection certificate solutions will be required further. COVID-19 certificate applications can be divided into two basic groups: (1) currently deployed/used solutions, and (2) experimental/academic solutions still under development. The first group includes applications that are used to prove non-infectiousness. These solutions are mainly focused on the usability of the application rather than on data privacy protection, e.g., European Union (EU) Digital COVID Certificates (DCC) (https://commission.europa.eu/strategy-and-policy/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_en, accessed on 5 March 2023). Further, some commercial applications are providing a certain level of privacy protection and are often integrated into larger health data management systems. An example is the IBM Digital Health Pass [1], which integrates test results and other health data into one system. Finally, the ongoing development of experimental solutions try to overcome the main issues of the previous applications, primarily focusing on protecting personal data. Well-known and experimental solutions from both groups are extensively explored in Section 2.

Generally speaking, current solutions are not compatible with the EU DCC, and those that are compatible suffer from several shortcomings, such as inefficient identification of the certificate holder and high invasion of user privacy with disclosure of sensitive information. Therefore, we need to find a solution for digital EU vaccination certificates that will protect citizens' privacy. This solution should solve the mentioned shortcomings in the field of privacy and security and be in accordance with the applicable EU legislation.

The paper is organized as follows. Section 2 presents the related work and our contribution. Section 3 presents the security and privacy issues of existing EU DCC solutions. Section 4 outlines the used notation and cryptographic primitives used in our proposal. Section 5 introduces our privacy-preserving solution for EU Digital Vaccine Certificates. Section 6 presents the security analysis of our proposal. Section 7 shows the implementation details, and Section 8 shows our experimental results. In Section 9, we discuss usability, benefits, and possible future extensions of our solution. In the last section, we conclude this work.

2. Related Work

There exist several proposals for the COVID-19 Pass, COVID-19 Certificates, or the Green Pass. Recently, Karopoulos et al. [2], Mbunge et al. [3], and Kissi et al. [4] surveyed proposals and works focused on COVID-19 digital certificates. In this section, we firstly present well-known deployed solutions; then, we present privacy-preserving proposals and concepts that are relevant to our scheme.

In 2021, the EU worked on uniforming documents on COVID-19 vaccinations, completed tests, and suffered disease to facilitate and make the proof of these facts in Europe efficient. The created system was called EU Digital COVID Certificates (DCC). This system was gradually adopted in several European countries [5]. In order to enhance the user experience for presenting and verifying the certificate, several countries, including the Czech Republic, developed dedicated smartphone applications. The applications are developed and managed by each EU member state independently, but they are compatible with each other due to unified rules on how to generate the certificates. For instance, the čTečka and Tečka [6] applications are considered the official applications of the digital Czech COVID-19 certificate. This application is available free of charge for both Android and iOS mobile platforms. The certificate compliance can be demonstrated through either a QR code or the raw data. Therefore, the app is used to show either a valid test, vaccination,

or recent illness. In order for the system to function, there must be a central authority (i.e., EU), and so-called EU gateways must be operated in individual countries, mediating the mutual recognition of certificates. Therefore, a specific infrastructure and a defined chain of cryptographic trust are necessary for the system to function. All participating countries are required to adhere to the central technical specifications, but at the same time, they also have the flexibility to incorporate nationally defined rules, e.g., duration of recognition of the validity of Polymerase Chain Reaction (PCR) or antigen tests. The data structure presented by the QR code contains the following information: (1) Name and surname of the given person (in the format according to the national alphabet and also separately without diacritics or other characters of the national alphabet), (2) Date of birth, (3) Vaccination-related data (vaccine used, number of vaccinations, date of last vaccination, member state, issuer, certificate identifier, and validity of vaccination according to national rules), or relevant data regarding tests or the disease experienced, and (4) Certificate signature (seal) to verify the authenticity of the given certificate. Everything stored in a QR code is in the JavaScript Object Notation (JSON) format, as prescribed by the relevant [7] specification.

Recently, Halpin [8] discussed the contradiction between privacy and unforgeability properties in EU Digital COVID Certificates (DCC). He concludes that DCCs could be more privacy-preserving than various blockchain-based vaccine passport solutions. Nevertheless, the privacy of DCCs could be improved by minimizing identifiers, abandoning transaction authentication numbers, and solving the unlinkability problem. On the other hand, maintaining unforgeability is impossible without revealing some personal data.

The IBM Digital Health Pass (IDHP) [1] is IBM's solution for authentication using COVID-19 vaccination certificates or test results. It is a complex system combining testing and vaccination into one unit using blockchain technology. The IDHP system includes the entire process, spanning from authorizing issuers and establishing the format of individual health certificates, to the issuance of certificates and their subsequent verification. The user end can be imagined as a certain form of digital wallet, wherein the user has his health data stored. This gives the user a degree of control over their sensitive health data. Nevertheless, only the following data can be inputted into the wallet: (1) SMART Health card, (2) Good Health Pass, and (3) IBM Digital Health Pass credentials (https://www.ibm.com/downloads/cas/EVD69GQ0, accessed on 5 March 2023). Consequently, it is therefore not possible to load a standard European vaccination certificate onto the system. According to IBM, the application does not hold any information that would link the presented data to the person in question. The user application is freely available for Android devices through the Google Play application. This solution is integrated into Amadeus' Traveler ID system.

The CoronaCheck app [9] is the official CovidPass app of the Netherlands. Its functionality is similar to ČTečka and Tečka. These are two separate applications for proof and verification of COVID certificates. However, the application takes into account the protection of personal data. The verifier application, CoronaCheck Scanner, does not show all the data read from the certificate. After identity verification, it only reveals the initial letters of the user's first and last name, along with the day and month of birth, while keeping the year concealed.

The COOV app [10] is South Korea's solution to the COVID Pass app. The application is focused on the protection of personal data. The vaccination certificate against COVID-19 is just one of the attributes that can be used to authenticate with this application. In addition to vaccination status and test results, this application can also be used to prove maturity. PASS-INFRA, which includes the COOV application, is a freely available solution and can also be used outside the Republic of South Korea. This solution uses a blockchain technology called InfraBlockchain [11]. This technology is independent of native cryptocurrencies, so it is suitable for this use. The technology is developed by Blockchain Labs, which is an active member of the Decentralized Identity Fund (DIF) foundation.

According to Halpin [8], IRMA [12] technology, with the modified Idemix [13] attribute-based authentication scheme, has been already used in Dutch COVID-19 credentials ensuring strong privacy. Nevertheless, as EU DCCs do not support IRMA/Idemix technology, there can be a problem when borders are crossed.

COVID Credentials Initiative (CCI) [14], as an open global community, collaborates on issuing open-standard-based privacy-preserving credentials and other related technologies for public health purposes. CCI deploys Verifiable Credentials (VCs) for representing user health and personal information in a digital, trustful, and tamper-evident manner.

In the recent years, numerous research works in the literature have adopted blockchain and smart contract techonologies, with the aim of harnessing the high availability, immutability, and traceability features that stem from utilizing a distributed public ledger [2]. Early decentralized approaches using Decentralized Identifiers (DIDs), public ledgers, and VCs have been analyzed by Halpin [15]. The work points to several issues such as lacking advanced cryptography or putting personal data on the blockchains. The work also raises ethical questions related to the use of immunity passports.

Recently, Barros et al. [16] proposed a privacy-preserving vaccination pass solution based on a self-sovereign identity concept with decentralized identifiers, VCs, zero-knowledge proofs, and blockchain. Their implemented prototype uses the Sovrin blockchain; Hyperledger Indy; JavaScript-based websites; Aries agents; Near-Field Communication (NFC) and Quick Response (QR) codes for local transfer of data; and digital wallet (connect.me) that is capable of managing VCs. There are also other blockchain-based privacy-preserving digital vaccine certificate solutions such as [17–19] that share similar advantages with Barros's solution, but also have several drawbacks such as robust user registration on the blockchain [17,18] or costly Ethereum transactions [19].

Karopoulos et al. [2] surveyed numerous proposals and works focused on blockchain-based COVID certificates. Their study observes that performance results, implementation details, and deployment aspects are often not given in those proposals. For instance, Kobbaey et al. [20] proposed a blockchain-based vaccination certificate model. Nevertheless, the proposal just employs underlying techniques without more information, and it does not specify how zero-knowledge-proofs and other techniques are designed or implemented.

In this paper, we present a novel solution for privacy-preserving EU Digital Vaccine Certificates (DVC). Our proposal is based on our Attribute-Based Credential (ABC) scheme, which allows us to disclose only the necessary attributes from a certificate. These attributes can represent, e.g., a photo of the certificate holder and the date of the last vaccination. Our solution addresses significant shortcomings of the current EU DVC, such as (1) insufficient identification of the person presenting the given digital certificate and (2) disclosure of all sensitive personal information about the holder and his/her vaccination status. We clarified, defined, and discussed the main security requirements for a privacy-preserving DVC. Furthermore, we provide a security analysis of our cryptographic core and system design. Finally, we provide implementation details about our proof-of-concept application and present several experimental results.

3. Identified Security and Privacy Issues of EU Digital COVID-19 Certificates

In this section, we present the most relevant security and privacy issues related to the implementation of EU DCCs, with a specific focus on the presentation of certificates through smartphone apps. Since there are several smartphone applications with equivalent characteristics, we use Czech Tečka [6] as a reference. In this application, the following issues can be identified:

- **Ineffective certificate holder identification:** The first issue of the EU DCC is the insufficient identification of the certificate holder. Although the certificate contains information about the holder such as name, date of birth, and nationality, it is not possible to verify and bind this information to the holder of the certificate without the presentation of additional identity documents, such as a national identity card, passport, or other similar documents according with the regulations of the respective

state. These documents contain a photo of the holder, enabling their binding with the certificate holder. The problem arises when the verifying person lacks the authority to request these additional documents from the person being verified or this person refuses to provide such documents, is not capable of doing so, or does not possess them. Without this step, the binding of the presented certificate to the given person is not verified, and the meaning of the entire control process is not achieved. Consequently, nothing prevents multiple people from using a single certificate.
***Solution**: Include a photo of the certificate holder in the digital vaccination certificate.*

- **High violation of user privacy:** The second issue of the EU DCCs is the high violation of user privacy by revealing much personal information about the user. As part of the non-infectivity or completed vaccination verification phase, the user has to present a complete vaccination certificate in the form of a QR code to a verifier. Therefore, a large amount of sensitive information about the user is revealed to the verifier. The verifying person not only gets basic information about non-infectivity or completed vaccinations, but also lots of other sensitive information such as name, surname, date of birth, nationality, or applied vaccination type. Moreover, the verifier may even take pictures of the QR codes, which implies that they do not only verify the validity of the certificate but also handle and store the personal data of all entering persons. In several situations, it is not necessary to disclose all of this information. Leaving aside the matter of binding the certificate to a specific person, the only relevant information is whether the user has a valid certificate or not.
  ***Solution**: Disclose only the necessary information from the certificate. Keep the rest of the information hidden from the verifier.*
- **Impersonation, alienation, or misuse of certificates:** The third issue of the EU DCCs is that QR codes only include static data about the certificate holder and vaccination details. There is no interactive cryptographic protocol used in order to prevent replay attacks. Therefore, an eavesdropper or anyone with access to the QR code can easily load the certificate onto one's own smartphone application, and the given certificate can be presented as their own. In fact, in some businesses, staff take pictures of the QR codes under the pretense that they will verify the QR code later or as evidence that they conducted the verification process.
  ***Solution**: Implement an interactive cryptographic protocol (i.e., a challenge-response protocol) and integrate the user's secret keys to the certificate verification phase.*
- **Missing revocation mechanisms:** The fourth issue of the EU DCCs is the absence of a certificate revocation mechanism. No certificate revocation list is used in smartphone applications and throughout the EU DCC system. Therefore, there is no standardized procedure to revoke compromised, stolen or fake certificates, such as publicly available QR codes of certificates issued in the name of Adolf Hitler or Mickey Mouse. The same problem occurs with invalid certificates, e.g., due to illness of the certificate holder.
  ***Solution**: Deploy online revocation databases which will include compromised, stolen, fake, and invalid certificates.*
- **Application hacking:** The fifth issue of the EU DCC is the time parameter of the digital certificate and the graphical display of certificate validation in smartphone applications. The applications retrieve the time information from the digital certificate and compare it with the system time of the smartphone. If someone changes the system time, the application will display the desired information about the validity of the given certificate (i.e., valid or invalid). In case the QR code is not read by the verifying application and the verifying person is satisfied with the displayed information presented by the certificate holder's smartphone application, changing the system time can be a serious problem. In fact, by adjusting the system time, the application will display the certificate as valid, even if the presented certificate has already expired.
  ***Solution**: Use a verifying application to verify the validity of the digital vaccination certificates.*

4. Cryptographic Preliminaries

In this section, at first, we outline the used notation and the security assumptions needed to understand our scheme and our security proofs. Second, we briefly introduce the attribute-based credential scheme [21] and the elliptic curve integrated encryption scheme [22] as the main building blocks of the cryptographic core of our privacy-preserving solution for EU digital vaccine certificates.

4.1. Notation and the Security Assumptions

From now on, the symbol ":" states for "such that", "$|x|$" the bitlength of x, and "$||$" the concatenation of two binary strings. We write $a \in_R A$ when a is sampled uniformly at random from A. A secure hash function is denoted as $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\kappa$, where κ is the given security parameter. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups of the same prime order q, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive groups and $\mathbb{G}_T$ is a multiplicative group. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be points of the respective group, and $\mathcal{O}$ be the point at infinity.

Definition 1 (Bilinear pairing)**.** *A bilinear pairing is a map* $\boldsymbol{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *which satisfies the following properties:*

- ***Bilinearity****:* $\forall x, y \in \mathbb{Z}_q, g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2 : \boldsymbol{e}(g_1^x, g_2^y) = \boldsymbol{e}(g_1, g_2)^{xy}$.
- ***Non-degeneracy****:* $\forall g_1 \neq \mathcal{O}\ \exists g_2 \in \mathbb{G}_2 : \boldsymbol{e}(g_1, g_2) \neq 1 \in \mathbb{G}_T$ *and* $\forall g_2 \neq \mathcal{O}\ \exists g_1 \in \mathbb{G}_1 : \boldsymbol{e}(g_1, g_2) \neq 1 \in \mathbb{G}_T$.
- ***Computability****: There exists an efficient algorithm* $\mathcal{G}(1^\kappa)$ *to compute* $\boldsymbol{e}(g_1, g_2)$.

By definition, $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$ is a bilinear group if it satisfies all above properties.

Let $\mathbb{G}$ be a cyclic group of order q and $g \in \mathbb{G}$ be its generator. The security of the proposed protocol relies on the n-Strong Computational Diffie–Hellman Inversion (n-SCDHI) assumption [21]:

Definition 2 (n-SCDHI Problem)**.** *Let* $\mathcal{O}(\cdot)$ *be the oracle that on input* $(m_1, \ldots, m_n) \in (\mathbb{Z}_q^*)^n$ *adds* $(m_1, \ldots, m_n)$ *to* Q *and outputs* $\{0,1\}$*. Let* $O^i(\cdot)$ *be the oracle that on input* h *outputs* h^{x^i}*. Then for all probabilistic polynomial time adversary* B*, the advantage is defined as follows:*

$$Adv_B^{n-SCDHI} = \mathbf{Pr}[(x_0, \ldots, x_n) \leftarrow \{0, \ldots, q-1\}; (y, m_1^*, \ldots, m_n^*) \leftarrow B^{\mathcal{O}(\cdot), \mathcal{O}^0(\cdot), \ldots, \mathcal{O}^n(\cdot)} : y = g^{\frac{1}{x_0 + \sum_{i=1}^n m_i^* x_i}} \wedge (m_1^*, \ldots, m_n^*) \notin Q]$$

SCDHI is (t, ϵ)*-hard if no* t*-time adversary has the advantage at least* ϵ*.*

Theorem 1. *The n-SCDHI problem is hard in the generic group model. More precisely, an adversary working in a generic group of order q with advantage* ϵ *requires time* $\Omega(3\sqrt{\epsilon q})$*.*

In this article, we consider the case $\mathbb{G}_1 \neq \mathbb{G}_2$; that is, when $\mathbf{e}$ is an asymmetric bilinear map and the SCDHI assumption holds. Moreover, having $\mathbb{G}_1 \neq \mathbb{G}_2$ permits obtaining the shortest possible signature; see [23] for more details.

4.2. Attribute-Based Credentials from wBB Signature

The Weak Boneh–Boyen (wBB) signature scheme [23] is a pairing-based short signature scheme. The scheme is provably secure, and it is proven to be existentially unforgeable against a weak (non-adaptive) chosen message attack. The scheme can be easily combined with zero-knowledge proofs as shown in the attribute-based credential scheme [21]. In this way, it is possible to prove the ownership of issued attributes in an unlinkable, anonymous, and efficient manner. The scheme was designed to be easily implementable and efficiently executed on computationally and memory-constrained devices such as smart cards. The scheme supports all the standard privacy-enhancing features of attribute-based credential schemes, such as anonymity, unlinkability, untraceability, and selective disclosure of

attributes. The main sign of the scheme is its symmetrical character, namely that the credential issuer also serves as the verifier, and therefore, they share secret keys. The scheme is briefly described below:

- $(par) \leftarrow \texttt{SetupI}(1^\kappa)$: on the input of the security parameter κ, the protocol generates the public system parameters $par = (\mathbb{G}, g, q)$ satisfying $|q| = \kappa$.
- $(sk, ipar) \leftarrow \texttt{CredKeygen}(par)$: on the input of the public system parameters par, protocol chooses $x_i \in_R \mathbf{Z}_q^*$ for $i = (0, \ldots, n)$ and outputs secret key $sk = (x_0, \ldots, x_n)$ and issuer parameters $ipar = (X_0, \ldots, X_n)$ where $X_i = g^{x_i}$.
- $(\sigma, \sigma_{x_0}, \ldots, \sigma_{x_n}) \leftarrow \texttt{Issue}(sk, (m_1, \ldots, m_n), par)$: on the input of the issuer's private key sk and the user's attributes $(m_1, \ldots, m_n)$, the protocol computes $\sigma = g^{\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}}$ and auxiliary values $\sigma_{x_i} \leftarrow \sigma^{x_i}$ for $i = (1, \ldots, n)$. The protocol outputs the credential $cred = (\sigma, \sigma_{x_0}, \ldots, \sigma_{x_n})$.
- $(0, 1) \leftarrow \texttt{Show}(ipar, par, cred, (m_1, \ldots, m_n), nonce, (D, \langle m_i \rangle_{i \in D}))$: on the input of the issuer parameters $ipar$, the public system parameters par, credential $cred$, user's attributes $(m_1, \ldots, m_n)$, and an authentication challenge $nonce$ of the verifier, the protocol randomizes the credential $cred$ by taking a random $r \in_R \mathbf{Z}_q^*$ and computing $\hat{\sigma} = \sigma^r$, taking $\rho_r, \rho_{m_{i \notin D}} \in_R \mathbf{Z}_q^*$ and computing:

$$t = \prod_{i \notin D} \sigma_{x_i}^{\rho_{m_i} \cdot r} g^{\rho_r}, \quad c = \mathcal{H}((D, \langle m_i \rangle_{i \in D}), t, \hat{\sigma}, par, ipar, nonce), \tag{1}$$
$$s_r = \rho_r + cr, \langle s_{m_i} = \rho_{m_i} - c m_i \rangle_{i \notin D}.$$

  The protocol outputs user's proof $proof = (\hat{\sigma}, c, s_r, \langle s_{m_i} \rangle_{i \notin D})$.
- $(0/1) \leftarrow (sk, par, (D, \langle m_i \rangle_{i \in D}), proof)$: on the input of the secret key sk, disclosed attributes $\langle m_i \rangle_{i \in D}$, and user's proof $proof = (\hat{\sigma}, c, s_r, \langle s_{m_i} \rangle_{i \notin D})$, the protocol computes:

$$t_{verify} \leftarrow g^{s_r} \cdot \hat{\sigma}^{-c \cdot x_0 + \sum_{i \notin D} (x_i \cdot s_{m_i}) - \sum_{i \in D} (x_i \cdot m_i \cdot c)} \tag{2}$$

  and checks that $c = \mathcal{H}((D, \langle m_i \rangle_{i \in D}), t, \hat{\sigma}, par, ipar, nonce)$. Output 1 if valid and 0 otherwise.

4.3. Elliptic Curve Integrated Encryption Scheme (ECIES)

ECIES [22] is an efficient and provable-secure encryption scheme based on the elliptic curve discrete logarithm problem. Let $\mathbb{G}$ denote a group of prime order q with generator g. Then, the public system parameters are $par = (\mathbb{G}, g, q)$. The scheme needs a symmetric encryption scheme $SYM = (E_k, D_k)$, a message authentication code MAC_k, and a key derivation function KDF. The ECIES scheme is briefly described below:

- $(pk, sk) \leftarrow \texttt{KeyGen}(par)$: on the input of the system parameters par, the protocol randomly chooses the secret key $v \in_R \mathbb{Z}_q$ and computes the public key $pk = g^v$.
- $(e) \leftarrow \texttt{Enc}(par, pk, m)$: on the input of the public key pk and a message m, the protocol randomly chooses $x \in_R \mathbb{Z}_q$ and computes $u = g^x$ and $t = pk^x$. Then, it computes the keys $(k_1, k_2) = KDF(t)$ which are used for encrypting the message $c = E_{k_1}(m)$ and for generating the message authentication code $r = MAC_{k_2}(c)$ of ciphertext c. The algorithm outputs $e = u||r||c$.
- $(\perp \ / m) \leftarrow \texttt{Dec}(par, sk, e)$: on the input of the secret key sk and the ciphertext c, the protocol parses e as $u||r||c$ and computes $t = u^{sk}$ and $(k_1, k_2) = KDF(t)$. If $r = MAC_{k_2}(c)$, then the algorithm returns $m = D_{k_1}(c)$; otherwise it returns an invalid $\perp$ result.

In addition to proving that the algorithm is secure, Smart [24] provides several specifications on the choice of SYM and KDF. Our scheme builds on the ECIES scheme and takes into consideration Smart's recommendations.

5. Privacy-Preserving Solution for EU Digital Vaccine Certificates

In this section, we define the whole system architecture, the roles of involved entities, and the used algorithms. Specifically, in Section 5.1, we describe the high-level architecture of our proposal, all involved entities, their roles in the system, and the communication model. In Section 5.2, we present our cryptographic core and all involved cryptographic algorithms.

5.1. System Architecture

- **Issuer/Revocation Authority (I/RA)**. This is the entity responsible for issuing individual attributes to the end-users. To do so, it runs the `Issue` protocol. The I/RA signs all issued attributes with its private key. We suppose that the I/RA is entrusted with the authority to process the users' personal data.
- **Verifier (VER)**. This is the entity responsible for verifying the ownership of required attributes by users who are interested in using the service. If revocation mechanisms are implemented in the system, the VER also verifies the revocation status of the presented certificates. Certificates are verified using the `Verify` protocol. In order to verify the attributes, the VER must possess the I/RA public key.
- **User (USR)**. This is the entity that holds the attributes issued by the I/RA and anonymously provides proof of their ownership to the VER in whose services it is interested. For this purpose, the USR runs the `Show` protocol.

Our cryptographic core is based on using the Fast Keyed-Verification Anonymous Credentials (FKVAC) [21]. Similarly to the FKVAC scheme, our scheme employs the weak Boneh–Boyen (wBB) [23] digital signatures to create the cryptographic credentials over user attributes. In contrast to the FKVAC scheme, our scheme offers to separate the cryptographic keys of the VER and the I/RA, resulting in an asymmetric cryptographic scheme. This way, it can be used in the application scenario of EU Digital Vaccine Certificates, where the VER and the I/RA are distinct entities which have different rights to the users' data. All this is performed, while ensuring the same (i.e., low) computing, memory, and cryptographic requirements on the user side (where users have an authentication device in the form of a smartphone or smart card). The system architecture of our solution is shown in Figure 1.

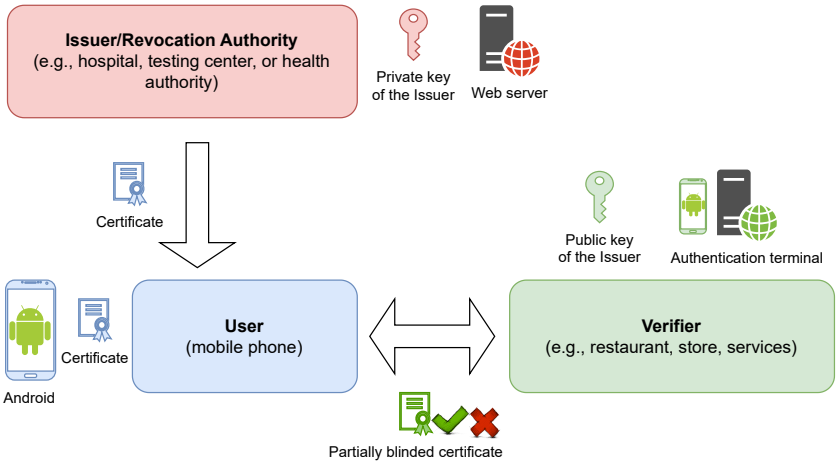


Figure 1. System architecture of the privacy-preserving solution for EU Digital Vaccine Certificates.

The communication model for presenting the vaccination certificate is shown in Figure 2. The USR is equipped with a smartphone capable of Bluetooth Low-Energy (BLE) communication. The service provider (acting as the VER) broadcasts the BLE advertisement (named, for example, PPCOVIDPASS). If the USR is interested in using the services of the service provider, it initiates a search and connection process through the BLE advertisement. The USR's mobile app will then connect to the VER app, and both apps will exchange

verification data. Namely, the VER sends the verifier certificate with the verifier public key and required attributes for disclosing and cryptographic nonce. Note that the verifier certificate is signed by the I/RA, which is trusted by both communication parties. The USR's app verifies the validity of the certificate, i.e., the validity of the generated I/RA signature. If the signature is valid, the USR decides whether to accept or decline the attributes required by the VER. If the user approves the attributes, the app sends an encrypted USR's photo (one of the user attributes), other required attributes (such as, for example: Dose number, Date of Vaccination), and unencrypted proof of knowledge of hidden attributes. In case of successful verification of the presented attributes and identification of the person presenting these attributes by the VER, the USR is granted access to the service; otherwise, it is denied.

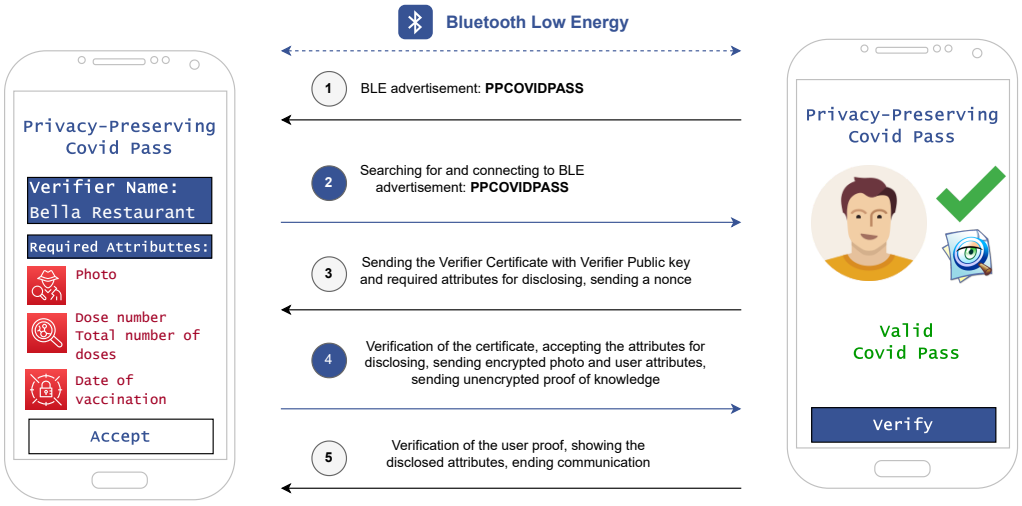


Figure 2. The communication model for presenting the privacy-preserving digital vaccine certificates.

The detailed structure of the digital vaccine certificate, including the representation of individual attributes, is shown in Figure 3. The cryptographic credential contains a total of 20 attributes. However, it can be expanded if it is needed. The first 16 attributes correspond to the attributes from EU Digital Vaccine Certificate. The following attributes extend the certificate by (1) a photo of the certificate holder enabling the connection of the holder with the certificate, (2) time stamp, (3) information about the completed vaccination (i.e., yes, no), and (4) one optional field according to the needs of the service provider.

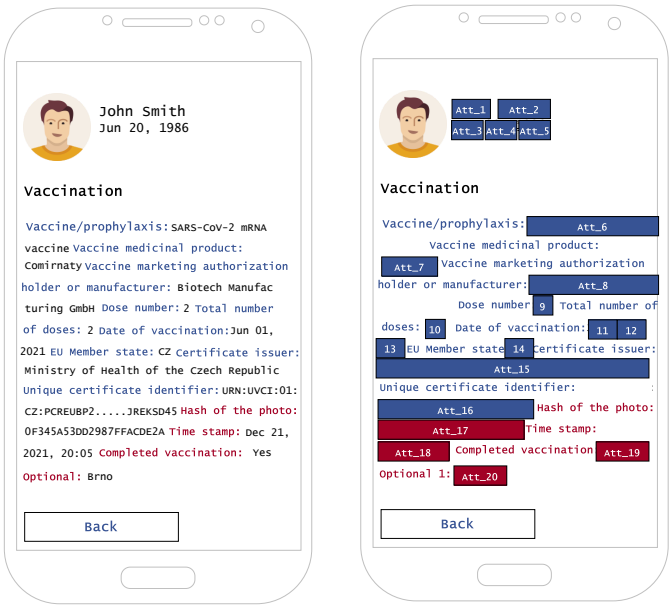


Figure 3. Representation of personal attributes in vaccine certificate (i.e., cryptographic credential).

5.2. Cryptographic Core

In this section, cryptographic algorithms are described in more detail, including their input and output variables.

1. $(sk_I, pk_I, params_I, pk_v, sk_v) \leftarrow$`Setup`$(1^\kappa)$: This algorithm works in two phases. At first, on the input of the security parameter κ, the I/RA generates and publishes the public parameters $params_I$ and generates the private/public key pair (sk_I, pk_I), where pk_I is published and sk_I is kept secret. Therefore, the I/RA works as follows:
   - Choose a bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are groups of the same prime order q, g_1 is a generator of $\mathbb{G}_1$, and g_2 is a generator of $\mathbb{G}_2$.
   - Define a secure hash function : $\mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q$.
   - Choose a symmetric encryption scheme $SYM = (Enc_{SYM}, Dec_{SYM})$.
   - Choose $sk_I = (x_0, \ldots, x_n) \in_R \mathbb{Z}_q$ as the issuer's private key, and set $pk_I = (h_0, \ldots, h_n) = (g_2^{x_0}, \ldots, g_2^{x_n})$ as the issuer's public key.
   - Publish the public system parameters $params_I = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \mathbf{e})$.

   Second, on the input of the parameter κ, the VER performs the following steps:
   - Randomly choose a private key $sk_V \in_R \mathbb{Z}_q$.
   - Compute and publish its public key $pk_V = g_1^{sk_V}$.
2. $(\sigma, \sigma_{x_0}, \ldots, \sigma_{x_n}) \leftarrow$`Issue`$(sk_I, (m_1, \ldots, m_n))$: On the input of the issuer's private key sk_I and the user's attributes $(m_1, \ldots, m_n)$, this protocol outputs the issuer's signature of the user's attributes $\sigma, \sigma_{x_0}, \ldots, \sigma_{x_n}$. This algorithm is run as an interactive protocol between the I/RA and the USR as follows:
   - The USR sends all its attributes $(m_1, \ldots, m_n)$ to the I/RA.
   - The I/RA computes the signatures of the USR's attributes as $\sigma = g_1^{\frac{1}{x_0 + m_1 x_1 + \cdots + m_n x_n}}$ where $sk_I = (x_0, \ldots, x_n)$ is its private key.
   - The I/RA calculates the auxiliary values as $\sigma_{x_i} = \sigma^{x_i}$, for $i = 0, 1, \ldots n$.
   - The I/RA sends σ and $\Lambda = (\sigma_{x_0}, \ldots, \sigma_{x_n})$ to the USR.

   `Issue` protocol steps are sketched in Figure 4.
3. $(0, 1) \leftarrow$`Show-Verify`$(m_{z \in D}, \Lambda, pk_V, pk_I, nonce)$: This algorithm works in two phases. At first, on the input of the verifier's public key pk_V, signatures σ, Λ, authentication challenge *nonce*, and disclosed attributes $m_{z \in D}$ (where D denotes the set of all revealed attributes), the USR outputs the encryption of the disclosed attributes $c_{z \in D}$, its cryptographic proof π, and the randomized credential $\hat{\sigma}$. This phase is defined as the `Show` algorithm. Secondly, on the input of the issuer's public key pk_I, the encrypted attributes $c_{z \in D}$, the proof π, and the randomized credential $\hat{\sigma}$, the VER outputs $0/1$, i.e., rejection or acceptance of the proof of knowledge of attributes. This phase is defined as the `Verify` algorithm. Therefore, the steps are as follows:
   - The VER generates a random authentication challenge *nonce* and sends it to the USR.
   - The USR randomizes its digital credential σ and Λ and constructs a proof of knowledge $(\hat{\sigma}, \pi)$, including the VER's authentication challenge *nonce*.
   - The USR generates a symmetric key $k_{\text{enc}} = KDF(j)$ and encrypts the attributes $c_i = Enc_{SYM}(m_i, k_{\text{enc}})$.
   - The VER reconstructs the symmetric key $k_{\text{enc}} = KDF(j)$, decrypts the attributes $m_i = Dec_{SYM}(c_i, k_{\text{enc}})$, and verifies the resulting proof π by using its challenge *nonce* and the issuer's public key pk_I.

   Figure 5 depicts the `Show-Verify` protocol in detail. The part of the protocol where the communication between the communicating parties is encrypted is marked in red.

User		**Issuer**
	$params_I = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \mathbf{e}), pk_I$	
$(m_1, \ldots, m_n)$		$sk_I = (x_0, \ldots, x_n)$
	$\xrightarrow{(m_1, \ldots, m_n)}$	
		$\sigma = g_1^{\frac{1}{x_0 + m_1 x_1 + \cdots + m_n x_n}}$
		$\sigma_{x_0} = \sigma^{x_0}$
		$\sigma_{x_1} = \sigma^{x_1}$
		$\vdots$
		$\sigma_{x_n} = \sigma^{x_n}$
	$\xleftarrow{\sigma, \sigma_{x_0}, \sigma_{x_1}, \ldots, \sigma_{x_n}}$	
Store: $\sigma, A = (\sigma_{x_0}, \sigma_{x_1}, \ldots, \sigma_{x_n})$		

Figure 4. `Issue protocol--Issue of Privacy-Preserving Digital Vaccine Certificate.`

User		**Verifier**
	$params_I = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \mathbf{e}), pk_I, pk_V$	
$(m_1, \ldots, m_n)$ $\sigma, A = (\sigma_{x_0}, \sigma_{x_1}, \ldots, \sigma_{x_n})$		$pk_I = (h_0, \ldots, h_n) = (g_2^{x_0}, \ldots, g_2^{x_n})$
	$\xleftarrow{nonce}$	
$\epsilon, \rho, \rho_v, \rho_{m_{i \notin D}} \in_R \mathbb{Z}_q$ $\hat{\sigma} \leftarrow \sigma^{\rho}$ $t \leftarrow g_1^{\rho_v} (\prod_{i \notin D} \sigma_{x_i}^{\rho_{m_i}})^{\rho}$ $e \leftarrow \mathcal{H}(t, \hat{\sigma}, nonce)$ $\langle s_{m_i} \leftarrow \rho_{m_i} - e m_i \rangle_{i \notin D}$ $s_v \leftarrow \rho_v + e\rho$ $\pi \leftarrow (t, s_{m_{i \notin D}}, s_v)$ $j \leftarrow pk_V^{\epsilon}, u \leftarrow g_1^{\epsilon}$ $k_{enc} \leftarrow KDF(j)$ $c_i \leftarrow Enc_{SYM}(m_i, k_{enc})$		
	$\xrightarrow{u, c_{i \in D}, \pi, \hat{\sigma}}$	
		$j' \leftarrow u^{Sk_V}$ $k'_{enc} \leftarrow KDF(j')$ $m'_i \leftarrow Dec_{SYM}(c_i, k'_{enc})$ $e' \leftarrow \mathcal{H}(t_{verify}, \hat{\sigma}, nonce)$ $\alpha \leftarrow h_0^{-e'} \cdot \prod_{i \notin D} h_i^{s_{m_i}} \prod_{i \in D} h_i^{-e' m'_i}$ $\mathbf{e}(t_{verify}, g_2) = \mathbf{e}(\hat{\sigma}, \alpha) \cdot \mathbf{e}(g_1^{s_v}, g_2)$

Figure 5. `Show-Verify` protocol—proof of possession of the attributes from the Privacy-Preserving Digital Vaccine Certificate.

6. Security Analysis

In this section, the security analysis of the cryptographic core takes place. In particular, we provide the proof of correctness, unforgeability, anonymity, and key–parameter consistency of our proposal. Note that the equality $t \stackrel{?}{=} t_{verify}$, i.e., Equations (1) and (2), has been proven in [21]. Therefore, the correctness of our proposal is proven as follows:

Theorem 2 (Correctness)**.** *The Verification process in Section 5.2, point 3, is correct.*

Proof. Since a symmetric cryptographic scheme is used to encrypt the attributes, at first, we show that the VER can reconstruct the USR's encryption key. In fact,

$$\begin{aligned} j' &= u^{sk_V} = (g_1^{\epsilon})^{sk_V} = pk_V^{\epsilon} = j \\ k'_{\text{enc}} &= KDF(j') = KDF(j) = k_{\text{enc}} \end{aligned} \tag{3}$$

and, therefore, $m'_i = Dec_{SYM}(c_i, k'_{\text{enc}}) = Dec_{SYM}(c_i, k_{\text{enc}}) = m_i$ for all $m_i \in D$. Accordingly, the decryption process is correct as shown in Equation (3). Once the message is correctly decrypted, we need to show that $\mathbf{e}(t_{verify}, g_2)$ is equal to $\mathbf{e}(\hat{\sigma}, \alpha) \cdot \mathbf{e}(g_1^{s_v}, g_2)$. Note that e' is equal to $\mathcal{H}(t_{verify}, \hat{\sigma}, nonce) = e$. The pairing equality can be proven as shown in Equations (4) and (5):

$$\begin{aligned} \mathbf{e}(\hat{\sigma}, \alpha) &= \mathbf{e}(g_1^{(\frac{1}{x_0 + m_1 x_1 + \cdots + m_n x_n})\rho}, h_0^{-e} \cdot \textstyle\prod_{i \notin D} h_i^{s m_i} \prod_{i \in D} h_i^{-e m_i}) \\ &= \mathbf{e}(g_1^{(\frac{1}{x_0 + m_1 x_1 + \cdots + m_n x_n})\rho}, g_2^{-e x_0 + \prod_{i \notin D}(\rho_{m_i} - e m_i) x_i + \prod_{i \in D} - e m_i x_i}) \\ &= \mathbf{e}(g_1^{(\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i})\rho}, g_2^{-e(x_0 + \sum_{i=1}^{n} m_i x_i) + \prod_{i \notin D} \rho_{m_i} x_i}) \\ &= \mathbf{e}(g_1, g_2)^{-e\rho + (\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}) \prod_{i \notin D} \rho_{m_i} x_i \rho} \end{aligned} \tag{4}$$

and, therefore,

$$\begin{aligned} \mathbf{e}(\hat{\sigma}, \alpha) \cdot \mathbf{e}(g_1^{s_v}, g_2) &= \mathbf{e}(g_1, g_2)^{-e\rho + (\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}) \prod_{i \notin D} \rho_{m_i} x_i \rho} \cdot \mathbf{e}(g_1, g_2)^{\rho_V + e\rho} \\ &= \mathbf{e}(g_1, g_2)^{\rho_V + (\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}) \prod_{i \notin D} \rho_{m_i} x_i \rho} \\ &= \mathbf{e}(g_1^{\rho_V} g_1^{(\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}) \prod_{i \notin D} \rho_{m_i} x_i \rho}, g_2) \\ &= \mathbf{e}(g_1^{\rho_V} \sigma^{\prod_{i \notin D} \rho_{m_i} x_i \rho}, g_2) \\ &= \mathbf{e}(g_1^{\rho_v} (\textstyle\prod_{i \notin D} \sigma_{x_i}^{\rho_{m_i}})^{\rho}, g_2) = \mathbf{e}(t_{verify}, g_2) \end{aligned} \tag{5}$$

□

Theorem 3 (Unforgeability)**.** *Our scheme is (t, ϵ, q_H)-unforgeable if the n-SCDHI problem (see Theorem 2 for more details) is $(2t, \epsilon')$-hard, where q_H is the maximum number of random oracle queries and $\epsilon' = \epsilon(\frac{\epsilon}{qH} - \frac{1}{q})$.*

Proof. Since no modifications are made on the `Issue` protocol, the proof follows directly from the unforgeability of the KVAC protocol [21]. □

We recall the definition of anonymity:

Definition 3 ([21])**.** *A keyed-verification credential system is anonymous if for all PPT adversaries $\mathcal{A}$, there exists an efficient algorithm `SimShow` such that for all κ, for all $\phi \in \Phi$ and $(m_1, \ldots, m_n) \in$ such that $(\phi(m_1, \ldots, m_n) = 1$, and for all $(sk_I, pk_I, params_I, pk_v, sk_v) \leftarrow$ `Setup`(1^{κ}): `Show`$(params_I, A, (m_1, \ldots, m_n), \phi) \leftrightarrow \mathcal{A} \rightarrow \approx SimShow(params_I, sk_I, \phi)$, i.e., the adversary's view given the proof can be simulated by `SimShow` given only ϕ and a valid secret key corresponding to $params_I$.*

Theorem 4 (Anonymity)**.** *Our scheme is anonymous, as defined in Definition 3, in the random oracle model.*

Proof. Since the modifications of the `Show-Verify` protocol do not affect the credential generation, the proof follows directly from the anonymity of the KVAC protocol [21]. □

We recall the definition of key-parameter consistency:

Definition 4. *A keyed-verification credential system is key-parameter consistent if for any PPT adversary, the probability that $\mathcal{A}$ given $params_I$ can produce $(params_I, sk_{I_1}, sk_{I_2})$ with $sk_{I_1} \neq sk_{I_2}$ such that $(params_I, sk_{I_1})$ and $(params_I, sk_{I_2})$ are both in the range of $\mathtt{Setup}(1^\kappa)$ is negligible in κ (where the probability is taken over the choice of $params_I$ and the random coins of $\mathcal{A}$).*

Theorem 5 (Key-parameter consistency)**.** *Our scheme is key-parameter consistent as defined in Definition 4.*

Proof. Following the KVAC protocol proof [21], `Setup` outputs $sk = (x_0, \ldots, x_n) \in \mathbb{Z}_q^*$ and $params_I = (x_0, \ldots, x_n)$, where $x_i = g_2^{x_i}$. Since $\mathbb{Z}_q^*$ is a prime order group, all of its elements have a unique discrete logarithm, and therefore, there is a unique sk corresponding to every $params_I$. □

7. Implementation Details

In this section, we present our proof of concept implementation of privacy-preserving digital vaccine certifications. The implementation serves primarily to evaluate performance tests and verify the functionality of the proposed solution. On one side, we have a smartphone application that is used by users to download and store their own digital vaccine certificates. If users need to prove themselves with the certificates, the application has the functionality to check the attributes required by the verifier (i.e., the service provider). These attributes are then displayed to the users, and their action is required (i.e., their approval or rejection). The user application is developed for the Android platform. We use the Kotlin programming language and the Android Studio development environment for its implementation. The application implements all algorithms presented in Section 5.2 and uses BLE communication to send the digital certificates from the user to the verifier. On the second side, we have a verifier's terminal application which is also implemented in Kotlin and partially shares the program code with a user's smartphone application. The source codes of our applications are publicly available on the online GitLab repository here: https://gitlab.com/brno-axe/mvcr-adiopsio, accessed on 1 September 2023.

7.1. Cryptographic Core

The cryptographic core is the same for both applications and is implemented using the MCL library [25]. This library supports the bilinear pairing operations and shows the best performance results according to [26]. The library is available in a native form for the Android platform and provides support for BN254 and BL12_381 elliptic curves. Note that our implementation uses the BN254 elliptic curve and requires bilinear pairing operations in the verification phase. Since the library is available only in native C/C++ form, we use the Android Native Development Kit (NDK) and a wrapper enabling the use of the library functions on the Android platform. The cryptographic core is represented with the `CryptoCore` class and implements eight methods: see Figure 6 for more detail. The first four methods implement the cryptographic algorithms (i.e., `Setup`, `Issue`, `Show`, `Verify`), and the other four methods provide support functions.

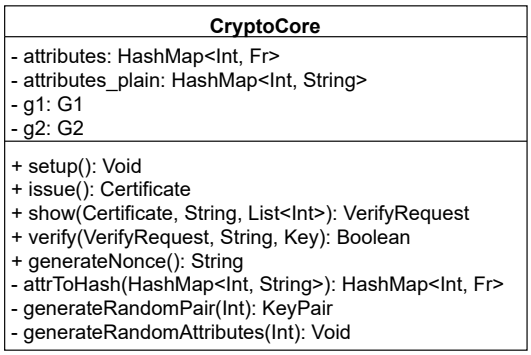


Figure 6. Structure of the `CryptoCore` class.

7.2. Data Structures

Both applications use several data classes. These classes define the structure of communication messages, certificates, and keys, and are defined in the `DataElements.kt` file. The data structures used in our implementation are detailed below:

- `Certificate`: This is the most important data structure. It represents the user's digital vaccine certificate. The structure itself contains the following values:
  - `attributes`: This is a value of type `HashMap<Int, String>` holding attributes in plain text form. Attributes are indexed based on their position (i.e., $1, 2, \ldots, 20$), not their name.
  - `attributesSigma`: Similarly to the `attributes`, `attributesSigma` is also of type `HashMap<Int, String>`, where in this case, the values are the σ_{x_i} signatures of all issued attributes.
  - `sigma`: This value represents the main signature of σ.
- `Pubkey`: This structure represents the issuer's public key. The class holds the value of `HashMap<Int, String>`, which is indexed in the same way as the `attributes` and `attributesSigma` structures.
- `NonceRequest`: This is the first data structure used within communication between the user's application and the verifier terminal. Its purpose is to transmit the verifier's challenge to the user.
- `VerifierCert`: This structure defines the digital certificate of the verifier issued by the I/RA. The verifier uses this certificate within the verification phase. In particular, the verifier decides which attributes the user should reveal during the verification phase. However, these attributes may vary based on the nature of the verifier. Therefore, the certificate is used to control whether a given verifier is entitled to see these required attributes or not. The data structure includes the following values:
  - `issuer_name`: The name of the I/RA. It is a variable of type `String`.
  - `verifier_name`: The name of the verifier. It is set when the certificate is issued. It is directly linked to the name of the entity that verifies digital certificates as part of the operation of services. `attributes`: A list of attributes that the verifier is entitled to query. This is a list of type `List<String>`.
  - `signature`: The digital signature of all values listed above.
- `VerifyRequest`: This data structure represents the user's response to the previous verifier's request `NonceRequest` and contains the following values: `sigma_roof`, `disclosedValues`, `t_verify`, `sm`, and `sv`. These values are used as cryptographic proof.
- `JsonResponse`: This data structure is used to process user digital vaccine certificates downloaded from the web server. The JSON containing this certificate has a slightly different form than the `Certificate` data structure, which is why this structure was created. A significant difference between the objects holding the certificates within the user application is the indexing method. Certificates downloaded from the web server

are indexed by attribute name, while the application indexes the attributes using their position. The data structure contains the following information:

  - `creation_time`: Date of certificate issuance, data type `String`.
  - `email`: User email, data type `String`.
  - `status`: This information indicates whether the certificate is valid, data type `Boolean`.
  - `certificate`. Attributes of the digital vaccine certificate, data type `Map`.
  - `sigma`: Signatures of σ, data type `Map`.

- `Device`: The structure created to display selected information about discovered BLE devices and their subsequent display. This structure holds the following information:
  - `name`: Advertised device name of type `String`.
  - `address`: MAC address of the Bluetooth interface of the device type `String`. `power`: Signal power of the discovered device `String`.

7.3. Communication Protocol

The communication between the user application and the verifier application is performed using four fixed messages corresponding to the data structures described above: see Figure 7. We use the BLE communication protocol to transmit data between the user and the verifier. The verifier's terminal is acting as a beacon. The user application is able to search for a device whose advertisement name includes the string `covidpass-terminal`. When the user selects one of the available verifier's devices, the first message is sent by the user device. To ensure the reliability of the communication, a simple acknowledgment of the sent chunks has been created.

Figure 7. Implemented communication model.

The first message carries the string `STARTVERIFY`. In response to receiving this message, the verifier's terminal responds with a message of the data type `NonceRequest`. The message thus carries a *nonce* challenge, a list of attributes to reveal, and the verifier's certificate. The verifier's certificate is used to verify the identity of the verifier, verify what attributes are to be disclosed to the verifier, and whether the verifier has the right to query those attributes. The certificate is verified upon receipt of the `NonceRequest` message, and the requested attributes are compared against the allowed attributes. Once the user decides to proceed with the verification process, the cryptographic proof is generated and sent to the verifier along with the revealed attributes. The sending of this message is signaled by including the string `VERIFYREQUEST` at the beginning of the message before the proof in the form of the data structure `VERIFYREQUEST`. When the message is received by the verifier

device, the proof is verified. Based on the result of this verification, the final user–verifier communication message is produced. The final message may take the form `VERIFY_OK` if the authentication is successful. If the verification fails, a message `VERIFY_FAILED` or `BAD_VERIFYREQUEST` is sent. The first message indicates that the verification failed. The second message indicates that a communication error occurred and the user is prompted to resend the third message of the protocol.

7.4. Application Structure

The Android mobile application is built upon the Model–View–ViewModel (MVVM) architectural pattern and implemented using modern technologies such as the Kotlin programming language, local Room database, and Jetpack Compose framework for rendering the user interface. The application is composed of user interface elements, database access classes, and support classes providing communication and cryptographic core functions. The interrelation of these elements is illustrated in Figure 8.

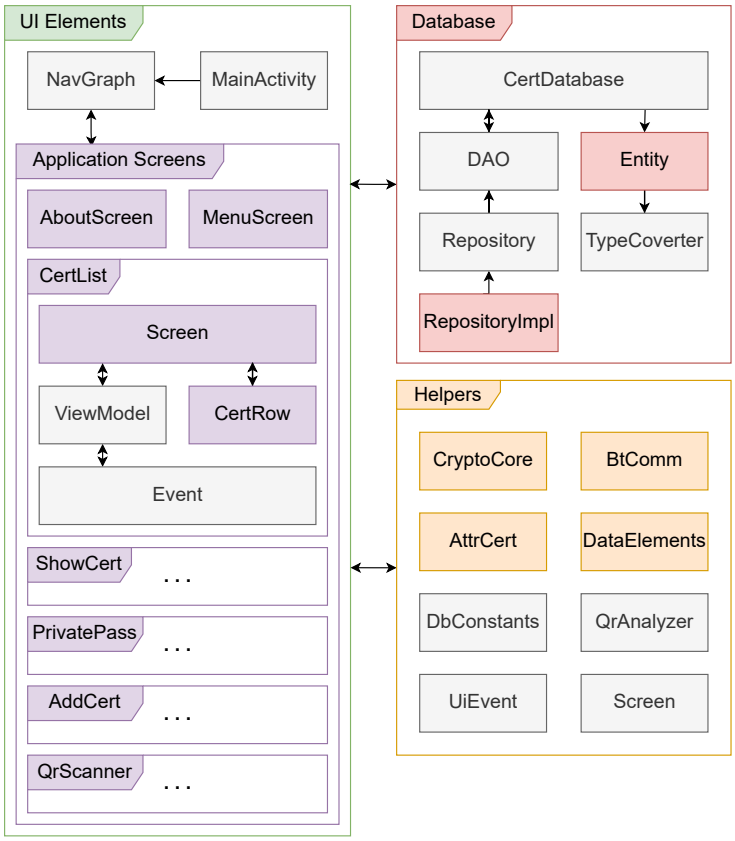


Figure 8. Android user application class diagram.

The user interface elements include the following classes:

- **Models**: Classes containing data structures that hold the data of different parts of the application.
- **Views**: Classes defining the graphical environment using the Jetpack Compose framework. These classes may include basic logic for controlling the rendering of individual elements.
- **View Models**: Classes containing models that affect the displayed values of the graphical interface of Android application activities.

Additionally, the application includes a persistent data model consisting of classes that offer basic operations over the database store and the definition of individual data entities, such as certificate data items. The last component of the application comprises supporting classes, mainly implementing the cryptographic core using the MCL library,

followed by the implementation of the communication stack through BLE. The BLE Stack is based on the standard functions provided by the Android operating system Application Programming Interface (API).

7.5. Application Database

User digital vaccine certificates are stored in a class representing the `Cert` entity containing the following items: certificate identifier, creation date, email, status, certificate σ value, and the values of their attributes and corresponding σ values. The implementation also includes a `CertRepository` class that allows basic certificate operations such as inserting, deleting, and reading from the Room database. The certificates can be inserted into the database using a one-time access token that allows the certificate to be downloaded from the I/RA's web server. The token can be scanned as a QR code for increased user-friendliness.

8. Experimental Results

In this section, we present our performance tests on various smartphone platforms and one smart card platform. First, we provide benchmark results of our cryptographic core. Second, we evaluate the overall performance of the verification phase including the BLE communication overhead.

8.1. Cryptographic Core Performance

To measure the time consumption of our algorithms, we developed a test application in the Java programming language that does not perform any communication overhead between system entities (i.e., issuer, user, and verifier). Each algorithm of the cryptographic core runs in this test application. Therefore, saving and loading of individual parameters was not performed. The application implements the cryptographic core listed in Section 7.1. The application uses the `measureTimeMillis()` function to measure the elapsed time of each algorithm. This feature allows us to measure Central Processing Unit (CPU) time with millisecond accuracy. Our experimental results are presented in Table 1 and Figure 9. The measurements confirmed the expectation that the time requirement of the algorithm increases with the number of hidden attributes. To make the experimental results more indicative, we performed all tests with all hidden attributes except the first attribute that was revealed. The most computationally demanding algorithm is the `Verify` algorithm. However, even when revealing a theoretical number of 500 attributes (which is a highly inflated value) from a cryptographic credential, the required time reaches less than 300 ms. Therefore, the measurements show that the cryptographic core does not represent any performance issue on current smartphone platforms.

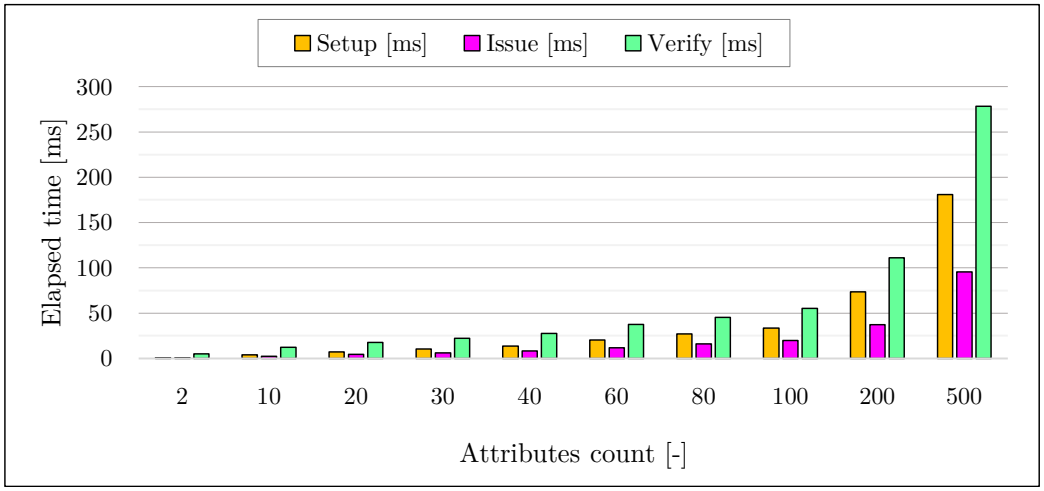


Figure 9. Performance of the cryptographic core depending on the number of hidden attributes.

Table 1. The time consumed to perform each phase of the cryptographic core on various devices.

Device	Attributes	2	10	20	30	40	60	80	100	200	500
Google Pixel 4a	Setup [ms]	1	4	7	10	12	18	25	31	68	162
	Issue [ms]	1	2	4	5	7	10	14	17	22	55
	Verify [ms]	5	11	16	20	25	34	41	50	103	262
Samsung S21 FE	Setup [ms]	1	2	4	6	8	12	16	19	42	106
	Issue [ms]	1	2	3	5	6	9	11	14	32	82
	Verify [ms]	3	7	10	13	16	22	26	32	65	163
Honor 8X	Setup [ms]	1	6	11	16	21	31	41	51	110	275
	Issue [ms]	1	3	6	9	12	17	24	28	58	150
	Verify [ms]	8	19	27	34	42	57	69	84	165	410
Average	Setup [ms]	1.0	4.0	7.3	10.7	13.7	20.3	27.3	33.7	73.3	181.0
	Issue [ms]	1.0	2.3	4.3	6.3	8.3	12.0	16.3	19.7	37.3	95.7
	Verify [ms]	5.3	12.3	17.7	22.3	27.7	37.7	45.3	55.3	110.0	278.3

8.2. Overall Performance Including BLE Communication Overhead

The second part of the performed experiments focused on testing all components of our privacy-preserving digital vaccine certificate application. Within the testing infrastructure, test certificates with a different number of attributes were created by the I/RA. These certificates were downloaded to the smartphone application using a secured web service. Then, the system performance was measured during user authentication to the VER's terminal. To run the application, we used a Raspberry Pi 4B single-board computer and a lower-performance Raspberry Pi Zero 2 W. Both of these devices have a built-in Bluetooth 5 interface which supports communication via the energy-efficient BLE protocol. The results show that the time duration of each protocol phase (i.e., `Scanning`, `Nonce`, `Verify`) is nearly identical for both computers. For representative results, all measurements were performed on a Raspberry Pi Zero 2 W: see Figure 10. Note that the `Scanning` protocol phase is responsible for finding and connecting the smartphone application to the BLE server, the `Nonce` phase performs the sending of an authentication challenge from the Verifier and the computation of the cryptographic proof of the user, and the `Verify` phase then checks the correctness of the cryptographic proof on the VER's side.

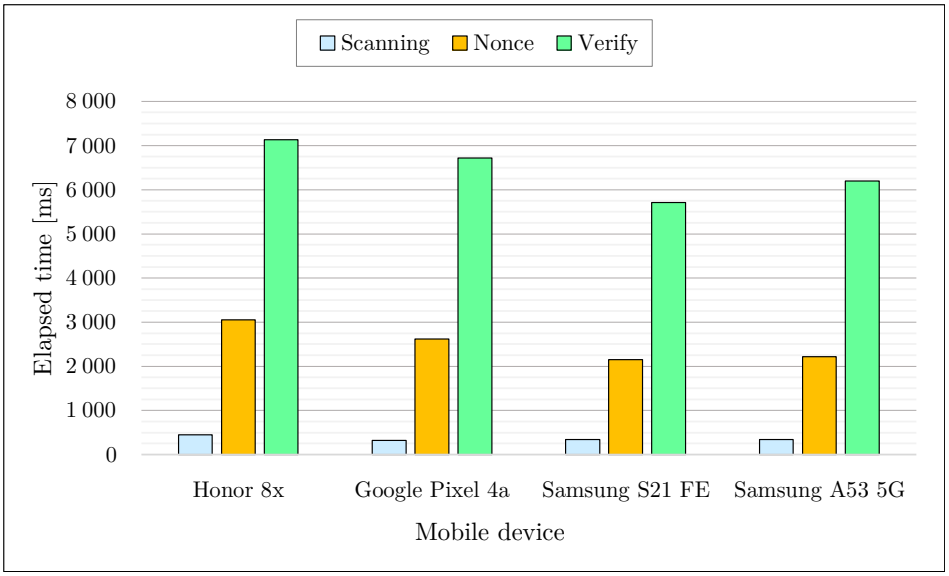


Figure 10. Performance of the implemented application when verifying a sample privacy-preserving digital vaccine certificate according to Table 2. The measurement results are the average of the elapsed time for 5 measurements.

Each certificate issued under the `Issue` algorithm contains one or more attributes and the appropriate attribute signature for each attribute. In the data structure representing the certificate, the attributes are stored as pairs containing a key and a value. The user application generates a `VerifyRequest` containing the revealed certificate entries and the digital signatures of the attributes. This request is serialized to the JSON format, and images, such as user photos, are converted to text using Base64 encoding. For testing purposes, a certificate containing attributes representing a possible certificate proving vaccination was used. The specific details included in the certificate are listed in Table 2. In total, there were 19 attributes. Additional attributes added for performance measurement purposes were added to the end of the certificate and contained 32-byte values for simplicity.

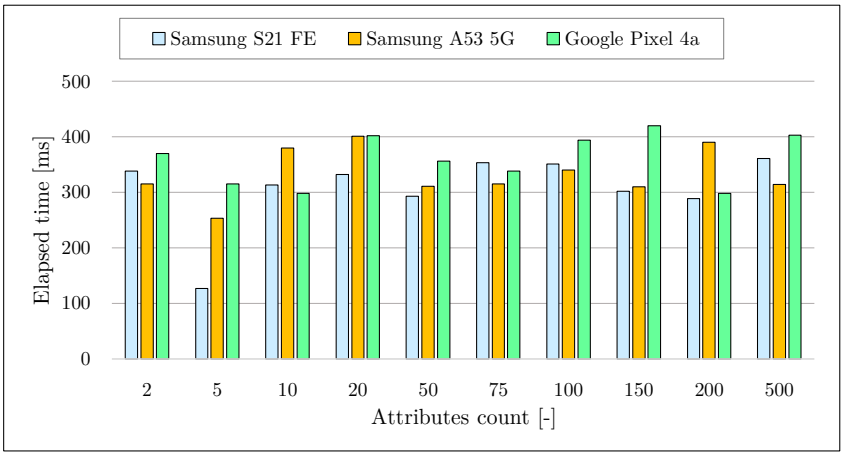
Table 2. Data fields and sample values within the test certificate. The field size includes keys and JSON delimiters.

Certificate Field	Field Key	Sample Value	Size [B]
First name	`firstname`	John	17
Surname	`surname`	Doe	14
Day of birth	`brithdate_day`	25	19
Month of birth	`brithdate_month`	8	20
Year of birth	`birthdate_year`	1990	22
Photo	`photo`	(Base64 data)	12, 600
Unique issuer ID	`unique_id`	12345678	21
Vaccination day	`vaccination_day`	10	21
Vaccination month	`vaccination_month`	10	20
Vaccation year	`vaccation_year`	2022	24
Order of dose	`dose`	2	9
Total doses	`total_dose`	2	15
Completed vaccination	`completed_vaccination`	True	29
Vaccine	`vaccine`	SARS CoV-2 Sample Vaccine	36
Product	`product`	Product A	20
Manufacturer	`manufacturer`	Manufacturer A	30
Issuer	`issuer`	BUT Brno, Czech Republic	34
State of EU	`state_eu`	CZ	14

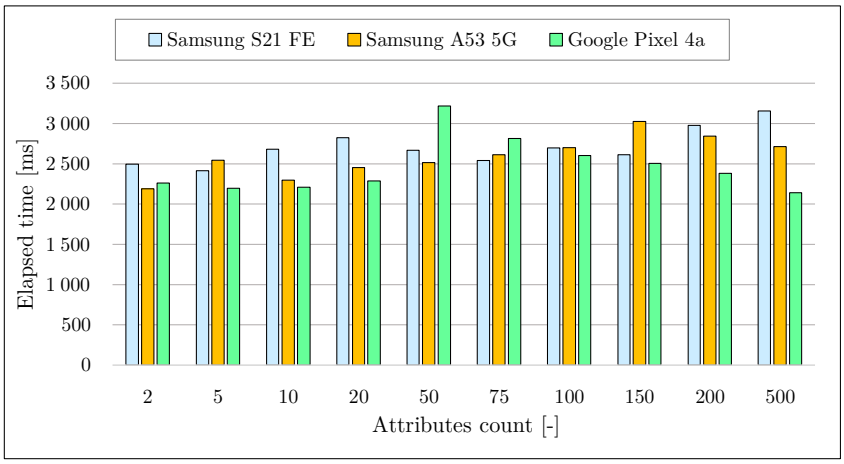
In addition, we also evaluated the computational complexity of our application in the case of revealing a theoretical number of 500 attributes (which is a highly inflated value) from a cryptographic credential: see Figure 11. The required verification time, including the BLE communication overhead, takes up to 3 s for most of the tested smartphones. The measurements therefore show that our application can be practical even in the case of a high number of attributes contained in the cryptographic credentials.

8.3. Performance on Smart Cards

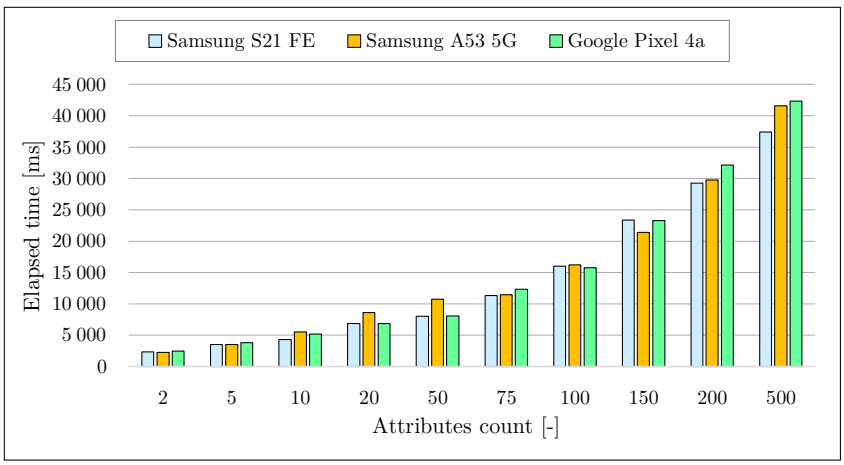
To evaluate the performance of our privacy-preserving digital vaccine certificate on the smart card platform, we used the MULTOS-based smart card implementation from our previous work in [21,27]. The `Show-Verify` algorithms were implemented using Barreto–Naehrig (BN) BN254 elliptic curve from the MCL library [25]. Only standard Multos API and free public development environment (Eclipse IDE for C/C++ Developers, SmartDeck 3.0.1, MUtil 2.8) were used. In the case of using smart cards for storing and presenting certificates, the verification time (in the worst case, i.e., 20 stored attributes, all hidden) can be expected to be up to around 3 s as shown in Figure 12. Since smart cards do not support BLE communication, it would be necessary to use Personal Computer/Smart Card (PC/SC) and contact/contactless communication interface.



(**a**) Scanning for the Verifier terminal.



(**b**) Nonce.



(**c**) Verify.

Figure 11. Application performance depends on the number of attributes. The size of all attributes was 32 B. Subfigure (**a**) shows needed time for detecting the BLE verifier terminal, subfigure (**b**) shows needed time for computing cryptographic proof on the user's side, and subfigure (**c**) shows needed time for verification of the cryptographic proof on the verifier' side.

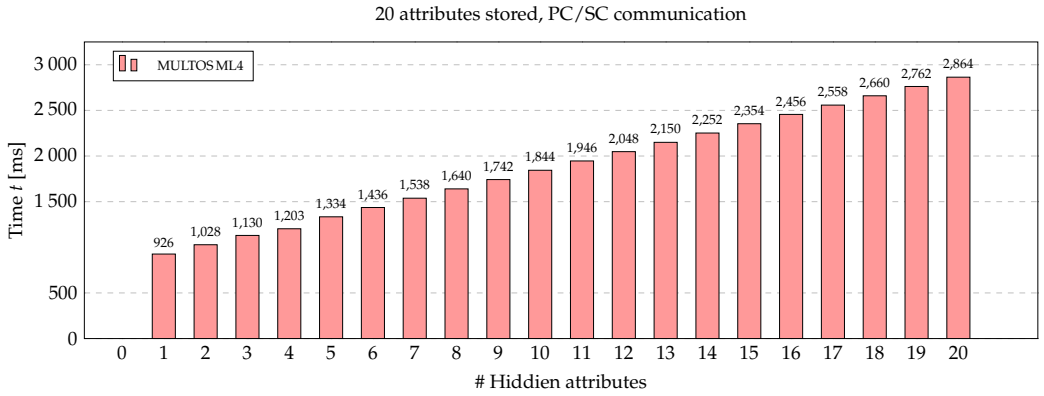


Figure 12. Application performance depending on the number of attributes on smart card platform

9. Discussion

The experimental results show that our solution is feasible to implement on current handheld devices, especially on Android smartphones and smartwatches. Applying cryptographic attribute-based credentials in digital vaccine certificates brings benefits in higher protection of citizens' privacy and is in line with current European legislations such as GDPR, the upcoming eIDAS 2.0, and its new tools that it envisages led by European digital identity (https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-digital-identity_en, accessed on 1 September 2023). This identity is intended to allow citizens to prove their identity to access online services, share digital documents, or simply prove specific personal characteristics such as age without revealing their identity or other personal information. We show that even in scenarios where certificates have a theoretical maximum of 500 attributes (which is a highly inflated value), the processing time remains under 300 ms on current smartphones. Consequently, our application can be practical even when dealing with certificates containing a high number of attributes.

European Health Insurance Cards are free cards that grant EU citizens access to medically necessary, state-provided healthcare while temporarily staying in any of the 27 EU countries, Iceland, Liechtenstein, Norway, Switzerland, and the United Kingdom, under the same conditions and at the same price. Therefore, European Health Insurance Smart Cards can be used to store information about the citizen vaccination status in the form of our privacy-preserving digital vaccine certifications. Our experimental results show that deploying the implementation of our solution on a smart card platform within a real-world environment is possible. Even with a digital vaccine certificate containing 20 attributes, the verification of completed vaccination would take approximately 3 s.

In our work, we consider various devices as secure storage of digital certificates (including smartphones/smartwatches and smart cards). Indeed, different devices bring different pros and cons, especially considering the security, time efficiency, and acquisition cost. Some works such as, for example [28], point out the problematic implementation of attribute-based credentials into real-world applications due to privacy and security risks. For instance, the identity disclosing risk allows a dishonest participant to start the conversation by requesting the entire set of attributes, resulting in a total loss of privacy and anonymity. This risk is caused, besides other reasons, by the use of smart cards. Smart cards do not have a user interface that allows users to directly control the attributes that are revealed and, if possible, stop the authentication protocol. Another disadvantage of smart cards is their computing power. As we have shown in our tests, the calculation of cryptographic proof is up to 1000× slower on a smart card than on smartphones, and it is limited by a small number of attributes. In this work, we assumed these risks mitigated these shortcomings and demonstrated a practical implementation of attribute-based credentials in a real-world scenario. To do this, we developed an Android mobile application allowing users present and control the revealed attributes from the digital certificate. Even if smartphones and smartwatches are not considered tamper-resistant

devices like smart cards, they provide sufficient security protection for user data thanks to the separation of applications, processes, and data (use of application sandbox) on the operating system level. That is why they are used in a number of security applications, such as bank or e-government applications. It is true that the acquisition costs of a smartphone are higher than the acquisition costs of a smart card, but if we take into account that most citizens already have a smartphone, this shortcoming is negligible.

Although our solution provides high security and privacy protection (see the security analysis in Section 6), we see also other possibilities to continue our work. This is primarily a further increase in the protection of citizen privacy and the decentralization of solutions for the support of cross-border Information and Communication Technologies (ICT) systems. Namely, our solution protects user privacy by disclosing only some information from the digital vaccine certificates. Furthermore, the disclosure of this information is also directly confirmed by the user (i.e., the certificate holder). All transmitted data are encrypted with end-to-end encryption (between the user and the verifier). Encrypted communication and randomized cryptographic proofs make it impossible for an eavesdropper to access data and profile users. Most of the disclosed attributes from the certificate have a general character and, therefore, do not directly lead to user profiling by the verifier (i.e., the service provider). However, the attribute of the user photo has a unique character and has to be always disclosed to allow a visual identification of the certificate holder. Thus, it allows the verifier to track the users (albeit anonymous users) within its services. To eliminate this risk, technologies such as Fuzzy extractors [29] deserve to be investigated. On the other hand, the blockchain technology (i.e., a public verifiable open ledger) and smart contracts (i.e., programmable logic programs integrated into the blockchain) have emerged as promising tools for devising decentralized models to securely settle arbitrary resource transactions such as digital vaccine certificates. This technology holds the potential to eliminate centralized entities involved in certificate management throughout their lifecycle, while ensuring the immutability, traceability, and availability of the involved assets. For instance, storing credential revocation schemes on the ledger would ensure high availability and maintain an immutable record of revoked certificates. Finally, our solution assumes the personal identification of certificate holders based on the presented photo (i.e., revealed attribute) from the digital certificate. However, the identification of the certificate holder itself is up to the inspecting person and is therefore subjective in nature. In order to guarantee the defined accuracy of the correct identification of certificate holders, it would be interesting to incorporate some of the biometric authentication methods into our solution.

10. Conclusions

This work presents a novel solution for the privacy-preserving EU Digital Vaccine Certificates. Our solution solves the privacy and security shortcomings of the current solution such as ineffective certificate holder identification and a high violation of user privacy with the disclosure of sensitive information. Furthermore, our solution is in line with current EU legislation, i.e., GDPR and eIDAS 2.0 regulations. The core of our proposal is built on our novel attribute-based credential scheme, which can be easily implemented on various handheld devices, especially on Android smartphones and smartwatches. However, due to the lightweight nature of our scheme, it can also be implemented on constrained devices such as smart cards. Our experimental results show that even if we use certificates with a theoretical number of 500 attributes (which is a highly inflated value), the required time would reach less than 300 ms on current smartphones. However, implementation of our solution on a smart card platform is also possible. In the case of a certificate containing 20 attributes, the verification of completed vaccination would take approximately 3 s. This would allow European Health Insurance Cards to be used as a repository for privacy-protecting digital vaccine certifications.

Author Contributions: Conceptualization, P.D.; methodology, P.D. and S.R.; software, P.D. and P.I.; validation, P.D., S.R., P.I., L.M. and C.A.-T.; formal analysis, P.D. and S.R.; investigation, P.D., S.R., P.I., L.M. and C.A.-T.; resources, P.D.; data curation, P.D. and P.I.; writing—original draft preparation, P.D., S.R., P.I. and L.M.; writing—review and editing, P.D., S.R., P.I., L.M. and C.A.-T.; visualization, P.D. and P.I.; supervision, P.D.; project administration, P.D.; funding acquisition, P.D., L.M. and C.A.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of the Interior of the Czech Republic under grant VJ03030014 (Development of International Collaboration in Cryptography and Cybersecurity Research). It is also supported by the PECT "Cuidem el que ens uneix" project, Operation Sensòrica [PR15-020174], within the frame of the RIS3CAT and ERDF Catalonia Operational Programme 2014-2020, co-financed by the Catalan Government and the Provincial Council of Tarragona; and by the Grant PID2021-125962OB-C32 "SECURING/DATA" funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Androulaki, E.; Circiumaru, I.; Vico, J.D.; Prada, M.; Sorniotti, A.; Stoecklin, M.; Vukolic, M.; Wallace, M. IBM Digital Health Pass Whitepaper: A Privacy-Respectful Platform for Proving Health Status; *Cryptol. ePrint Arch.* **2021**, *preprint*.
2. Karopoulos, G.; Hernandez-Ramos, J.L.; Kouliaridis, V.; Kambourakis, G. A survey on digital certificates approaches for the covid-19 pandemic. *IEEE Access* **2021**, *9*, 138003–138025. [CrossRef]
3. Mbunge, E.; Fashoto, S.; Batani, J. *COVID-19 Digital Vaccination Certificates and Digital Technologies: Lessons from Digital Contact Tracing Apps*; Available at SSRN 3805803; SSRN: Rochester, NY, USA , 2021.
4. Kissi, J.; Kusi Achampong, E.; Kumasenu Mensah, N.; Annobil, C.; Naa Lamptey, J. Moving towards Digitising COVID-19 Vaccination Certificate: A Systematic Review of Literature. *Vaccines* **2022**, *10*, 2040. [CrossRef] [PubMed]
5. EU Digital COVID Certificate. Available online: https://commission.europa.eu/strategy-and-policy/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_en (accessed on 25 March 2023).
6. Validation Applications čTečka and Tečka. Available online: https://covid.gov.cz/en/situations/vaccination/validation-applications-ctecka-and-tecka (accessed on 25 March 2023).
7. EHealth Network. Available online: https://health.ec.europa.eu/system/files/2021-06/covid-certificate_json_specification_en_0.pdf (accessed on 25 March 2023).
8. Halpin, H. A Critique of EU Digital COVID-19 Certificates: Do Vaccine Passports Endanger Privacy? In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–8.
9. CoronaCheck App and Printed Corona Admission Ticket Privacy Statement. Available online: https://coronacheck.nl/en/privacy (accessed on 17 December 2021).
10. COOV APP. Available online: https://ncv.kdca.go.kr/coov (accessed on 17 December 2021).
11. Pols, F. Technologies for Transparency. Available online: https://infrablockchain.com/documents/InfraBlockchain_Technical_White_Paper_Version_2_4_ENG_202008.pdf (accessed on 17 December 2021).
12. Digitální Certifikát EU COVID. 2021. Available online: https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_cs (accessed on 17 December 2021).
13. Bichsel, P.; Binding, C.; Camenisch, J.; Groß, T.; Heydt-Benjamin, T.; Sommer, D.; Zaverucha, G. *Specification of the Identity Mixer Cryptographic Library Version 2.3.0**; Technical Report; IBM: Armonk, NY, USA, 2010.
14. COVID Credentials Initiative. Available online: https://www.covidcreds.org/ (accessed on 25 March 2023).
15. Halpin, H. Vision: A critique of immunity passports and w3c decentralized identifiers. In Proceedings of the Security Standardisation Research: 6th International Conference, SSR 2020, London, UK, 30 November–1 December 2020; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2020; pp. 148–168.
16. De Vasconcelos Barros, M.; Schardong, F.; Felipe Custódio, R. Leveraging self-sovereign identity, blockchain, and zero-knowledge proof to build a privacy-preserving vaccination pass. In *Blockchain, and Zero-Knowledge Proof to Build a Privacy-Preserving Vaccination Pass*; SSRN: Rochester, NY, USA, 2022.
17. Eisenstadt, M.; Ramachandran, M.; Chowdhury, N.; Third, A.; Domingue, J. COVID-19 antibody test/vaccination certification: There's an app for that. *IEEE Open J. Eng. Med. Biol.* **2020**, *1*, 148–155. [CrossRef] [PubMed]
18. Barati, M.; Buchanan, W.J.; Lo, O.; Rana, O. A privacy-preserving distributed platform for COVID-19 vaccine passports. In Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion, Leicester, UK, 6–9 December 2021; pp. 1–6.

19. Hasan, H.R.; Salah, K.; Jayaraman, R.; Arshad, J.; Yaqoob, I.; Omar, M.; Ellahham, S. Blockchain-based solution for COVID-19 digital medical passports and immunity certificates. *IEEE Access* **2020**, *8*, 222093–222108. [CrossRef] [PubMed]
20. Kobbaey, T.; Bilquise, G.; Alqatawna, J.; Dashti, O. A Blockchain-based Vaccination Model for COVID-19 and Other Infectious Diseases. In Proceedings of the 2022 8th International Conference on Information Technology Trends (ITT), Dubai, United Arab Emirates, 25–26 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 189–195.
21. Camenisch, J.; Drijvers, M.; Dzurenda, P.; Hajny, J. Fast keyed-verification anonymous credentials on standard smart cards. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Lisbon, Portugal, 25–27 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 286–298.
22. Gayoso Martínez, V.; Hernández Encinas, L.; Sánchez Ávila, C. A survey of the elliptic curve integrated encryption scheme. *J. Comput. Sci. Eng.* **2010**, *2* , 7–13 .
23. Boneh, D.; Boyen, X. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptol.* **2008**, *21*, 149–177. [CrossRef]
24. Smart, N.P. The exact security of ECIES in the generic group model. In Proceedings of the Cryptography and Coding: 8th IMA International Conference Cirencester, UK, 17–19 December 2001; Proceedings 8; Springer: Berlin/Heidelberg, Germany, 2001; pp. 73–84.
25. Shigeo, M. Mcl Library. Available online: https://github.com/herumi/mcl (accessed on 25 March 2023).
26. Dzurenda, P. Cryptographic Protection of Digital Identity. Master's Thesis, Brno University of Technology, Brno, Czech Republic, 2 September 2019.
27. Casanova-Marqués, R.; Dzurenda, P.; Hajny, J. Implementation of Revocable Keyed-Verification Anonymous Credentials on Java Card. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–8.
28. Dzurenda, P.; Casanova-Marqués, R.; Malina, L.; Hajny, J. Real-world Deployment of Privacy-Enhancing Authentication System using Attribute-based Credentials. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–9.
29. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139. [CrossRef]