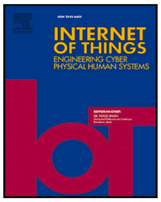
Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

Maximizing privacy and security of collaborative indoor positioning using zero-knowledge proofs

Raúl Casanova-Marqués [a,b,*], Joaquín Torres-Sospedra [c,**], Jan Hajny [a], Michael Gould [b]

[a] *Brno University of Technology, FEEC, Department of Telecommunications, Technicka 12, Brno, 616 00, Czech Republic*
[b] *Universitat Jaume I, Institute of New Imaging Technologies, Avda. Vicente Sos Baynat S/N, Castellón, 12071, Spain*
[c] *Universidade do Minho, Algoritmi Research Centre, Campus Azurém, Guimarães, 4800-058, Portugal*

ARTICLE INFO

Keywords:
Attribute-based credentials
Decentralized authentication
Privacy
Anonymity
Collaborative indoor positioning systems
Bluetooth low energy
Wearables

ABSTRACT

The increasing popularity of wearable-based Collaborative Indoor Positioning Systems (CIPSs) has led to the development of new methods for improving positioning accuracy. However, these systems often rely on protocols, such as iBeacon, that lack sufficient privacy protection. In addition, they depend on centralized entities for the authentication and verification processes. To address the limitations of existing protocols, this paper presents a groundbreaking contribution to the field of wearable-based CIPSs. We propose a decentralized Attribute-based Authentication (ABA) protocol that offers superior levels of privacy protection, untraceability, and unlinkability of user actions. Unlike existing protocols that rely on centralized entities, our approach leverages decentralized mechanisms for authentication and verification, ensuring the privacy of user location data exchange. Through extensive experimentation across multiple platforms, our results demonstrate the practicality and feasibility of the proposed protocol for real-world deployment. Overall, this work opens up new avenues for secure and privacy-preserving wearable-based CIPSs, with potential implications for the rapidly growing field of Internet of Things (IoT) applications.

1. Introduction

Undoubtedly, Location-based Services (LBSs) are present in our daily lives. They can suggest optimal routes to reach a place, enable autonomous industrial vehicles [1,2], or even track lost or stolen objects or elders at home [3,4]. Recently, wearable-based Collaborative Indoor Positioning Systems (CIPSs) have gained prominence, spurred by the desire to overcome the disadvantages of traditional approaches such as Wi-Fi or Bluetooth Low Energy (BLE) fingerprinting [5]. That is, an expensive infrastructure of beacons and servers, as well as limited positioning accuracy of around a few meters [6]. These collaborative systems enable users to exchange information, for instance in terms of BLE advertising packets such as iBeacon, which can be used to compute the relative distances between them using the Received Signal Strength (RSS) values gathered. This alternative is simple to set up and provides a reasonable trade-off between power consumption and performance [7]. However, the existing BLE protocols, such as iBeacon, are not currently prepared to provide sufficient privacy protection.

Data privacy is both a regulatory must and an increasing consumer expectation [8,9]. A few years ago, people were concerned about the privacy of their personal information; now, privacy is a requirement. Every user's data must be subjected to the

[*] Corresponding author at: Brno University of Technology, FEEC, Department of Telecommunications, Technicka 12, Brno, 616 00, Czech Republic.
[**] Corresponding author at: Universidade do Minho, Algoritmi Research Centre, Campus Azurém, Guimarães, 4800-058, Portugal.
E-mail addresses: casanova@vut.cz (R. Casanova-Marqués), info@jtorr.es (J. Torres-Sospedra).

https://doi.org/10.1016/j.iot.2023.100801
Received 2 March 2023; Received in revised form 13 April 2023; Accepted 26 April 2023
Available online 6 May 2023

most stringent protection and security analysis. This concern for privacy and the requirement that a user cannot be easily or unambiguously identified compel us to review the authentication mechanisms currently in use. Given this predisposition, it is imperative to establish authentication techniques that allow users to apply their privacy requirements. Currently, authentication schemes based on smart cards or biometrics are traceable and linkable [10]. These schemes acquire excessive information from users to verify that they have legitimate credentials to utilize a certain service or get access to a restricted area.

Systems of such an invasive nature have prompted the development of alternatives that respect the privacy of their users more. These suggestions are primarily concerned with the use of zero-knowledge proofs for the anonymous authentication and verification of users. In particular, Attribute-based Authentication (ABA) schemes have gained popularity since they enable users to anonymously and selectively demonstrate their ownership of personal attributes. Personal characteristics such as the legal age, citizenship, the validity of a transportation ticket, and the SARS-CoV-2 test result. Nevertheless, these schemes are typically centralized models in which the issuer and verifier are the same entity. This design constitutes a “single point of failure” in the system’s verification procedures, jeopardizing its security and availability.

Existing CIPSs are also based on centralized models [5]. Adopting a decentralized architecture is a potential countermeasure to avoid the drawbacks of centralized systems. However, the usage of decentralized paradigms poses significant security and privacy concerns due to the constant communication among unknown devices and the absence of secure communication protocols. In addition, the information transferred through BLE, utilizing protocols such as iBeacon, is sent in plain text and without authentication. Since malicious users are capable of impersonating genuine users, injecting fake information, and performing eavesdropping [11], every piece of information exchanged must be appropriately authenticated and protected.

In light of the identified problems, in this work we pursue the decentralization of CIPSs while providing, through anonymous ABA schemes, a secure and privacy-friendly system for location data exchange.

1.1. Related work

In the field of research, numerous methods have been presented to safeguard the privacy of user location information in the context of Location-based Services. These solutions aim to protect sensitive information from being disclosed to unauthorized parties, such as third-party advertisers or malicious individuals, while still allowing LBS applications to provide personalized and relevant services to users. This is becoming increasingly important as the use of LBS continues to grow and more data is generated through these services.

An illustration of Location-based Services can be found in the realm of the Internet of Vehicles (IoV). IoV applications are often dependent on the collection and processing of sensitive information. This information may contain driving habits, personal information, and location data, making privacy a critical issue that must be addressed to ensure the protection of users’ information from unauthorized disclosure or misuse. Malandrino et al. [12] proposed a method to verify and infer the positions of vehicles in vehicular networks while preserving their privacy. The authors used the technique of anonymous beaconing, which broadcasts the positions of vehicles without revealing their identities, to achieve this goal. Liu et al. [13] presented a new privacy-preserving trust evaluation scheme named LPPTE, which aimed to enhance the fusion of data from different sources in cooperative vehicular safety applications. The authors pointed out that conventional trust evaluation schemes were not appropriate for these applications because of privacy issues, and introduced a lightweight alternative that maintained privacy while enabling the evaluation of trust among sources. The scheme was thoroughly explained, and its performance was assessed through simulations. Huang et al. [14] explored the privacy challenges in LBSs in the IoV. The authors proposed a privacy-preserving scheme that aims to protect the privacy of users while still providing accurate location information. Xi et al. [15] discussed a privacy-enhancing technology for the IoV. The authors proposed a Zero-knowledge Proof (ZKP)-based anonymous mutual authentication scheme called ZAMA to provide secure communication between vehicles and other entities in the IoV while preserving privacy. ZAMA uses ZKPs to verify the identity of participants without revealing any personal information, thereby improving the security and privacy of the IoV. The aforementioned publications suffer from several security concerns, including the transmission of data in plaintext, reliance on centralized servers, linkability and traceability of user actions, and the lack of revocation mechanisms.

The utilization of LBSs extends to Global Positioning System (GPS) navigation, location-based advertising, and location-based social networking. As LBS continue to evolve, preserving privacy and security is becoming increasingly important. Several research articles explore methods and techniques for ensuring privacy in LBS applications, such as privacy-preserving schemes and algorithms, differential privacy, and blockchain technology. These privacy-preserving solutions are not only relevant to existing LBS applications, but also have the potential to enhance privacy and security in emerging applications, such as Coronavirus Disease 2019 (COVID-19) contact-tracing solutions.

Peng et al. [16] presented a study on a privacy-preserving scheme for LBSs that addressed the issue of trajectory privacy. The authors proposed a collaborative approach for preserving the privacy of a user’s trajectory by using a combination of cryptographic methods and data perturbation techniques. The study evaluated the performance of the proposed scheme and compared it to existing methods in terms of privacy protection, computation overhead, and communication costs. Jarvinen et al. [17] introduced PILOT, a pioneering Indoor Positioning System (IPS) that addresses privacy concerns by using several advanced techniques such as secure multi-party computation of distance metrics, quantization of RSS values, the k-Nearest Neighbors (k-NN) algorithm, and oblivious array access. They reported a running time below 1 s and provided a comprehensive evaluation of their solution. Gupta and Shanker [18] focused on improving the performance of LBSs through data caching. The authors suggested a new cache management policy called OMCPR that uses spatial k-anonymity to balance the trade-off between preserving user privacy and efficiently using cached data. The results indicated that OMCPR outperformed other policies in terms of both privacy protection and performance enhancement for LBSs. Shubina et al. [19] delved into the delicate equilibrium between location accuracy and privacy in wearable networks, exploring the challenges that arise when seeking to achieve high accuracy while ensuring user privacy. The study proposed several potential solutions to this conundrum and evaluated their performance. Additionally, the implications associated with implementing these solutions were discussed in depth. Kim et al. [20] introduced a survey paper that examines the use of Differential Privacy (DP) techniques in LBSs. The authors provided an overview of the existing privacy-preserving methods for location data and discussed their applicability to real-world LBSs. Barsocchi et al. [21] presented a new reference architecture for indoor positioning and discussed the challenges encountered during its installation and operation in real-world scenarios. In particular, they explored the database infrastructure and security procedures required to ensure data isolation, anonymization, and preservation in accordance with current legislation. Jiang et al. [22] presented an overview of the opportunities, challenges, and potential applications of DP in the Industrial Internet of Things (IIoT). The authors highlighted the importance of DP in preserving the privacy of users in IIoT, while enabling efficient data processing and analysis. Shubina et al. [23] conducted a qualitative comparison of current COVID-19 contact-tracing solutions in development, analyzing factors such as positioning technology, measurement, architecture, detection accuracy, energy efficiency, and privacy level. The authors emphasized the importance of privacy as a critical factor in these solutions, while also highlighting the specific strengths and characteristics of each solution. Yang et al. [24] presented a privacy-preserving solution for indoor navigation systems by incorporating location-based oblivious sharing. This allows for shared access to location information while protecting users' privacy by obscuring their exact locations. Li et al. [25] explored the integration of LBSs with blockchain technology. The authors proposed a novel method to enhance the security and trust in location data management using a blockchain-based system named SAGIN. This method aimed to improve the efficiency of location data management while preserving the privacy and security of users' data. Hu et al. [26] introduced PriHorus, a novel privacy-preserving IPS based on RSS and partial homomorphic encryption. Unlike prior work such as PILOT, PriHorus relied on maximum likelihood estimation instead of the classical k-NN algorithm. Guo et al. [27] presented FedPos, a novel federated transfer learning framework for indoor positioning based on Channel State Information (CSI) from a single Access Point (AP). Although their model showed high transferability, it is currently limited to devices capable of measuring CSI data, which remains unavailable in wearable devices. The primary issue with most of the aforementioned publications is their utilization of a centralized server architecture to support their proposed LBSs, which may introduce security and privacy vulnerabilities and increase the risk of unauthorized data access or breaches. Moreover, some of these papers suggest mechanisms that may lead to traceable or linkable information, posing potential threats to user privacy.

Location-based Services span a broader range of applications, including Collaborative Indoor Positioning Systems. Unfortunately, privacy in CIPS has received far too little attention. The systematic review conducted by Pascacio et al. [5] served as a starting point for our analysis of the various CIPSs already in use. The review selected and assessed a total of 84 papers that were published between 2006 and 2020. None of the examined works accounted for user privacy or CIPSs security. We performed a literature search using scientific databases, including the most prominent computer science journals and conferences: IEEE Xplore, ACM Digital Library, Elsevier, ScienceDirect, and Springer Link. We were only able to choose two publications that directly or indirectly address the absence of security and privacy in these systems. Zidek et al. [28] designed the Bellrock scheme, which combines an ecosystem of standard beacons with user-specific beacons. Bellrock offers access control to conventional beacons and anonymity to user-based beacons, utilizing three methods, i.e., random, synchronized, and encrypted, to produce pseudo-anonymous identifiers that may be unmasked by a server. Yin et al. [29] suggested a federated localization framework called FedLoc. The framework seeks to provide precise location services cooperatively without jeopardizing user privacy, particularly sensitive information pertaining to their geographic trajectories. The above proposals have a significant drawback in that they rely on a central server, making the system a centralized environment. This dependence on a central server poses various risks, including the possibility of a Denial of Service (DoS) attack or even the risk of spoofing, among others.

The landscape of collaborative LBSs has recently seen several proposals introduced. Fraile and Koulamas [30] proposed an indoor positioning system based on mobile devices that uses BLE signals to estimate the location of mobile devices in indoor environments. Delgado et al. [31] suggested a system for connected robots that employs a distributed architecture with multiple robots collaborating to accomplish mission-critical tasks. Pascacio et al. [32] put forward a neural network-based approach to improve indoor positioning accuracy using BLE RSS lateration. Finally, Wong and Lee [33] introduced an indoor navigation and information sharing system for emergency response situations that uses Building Information Modeling (BIM) and multi-user networking to facilitate collaboration among emergency responders. All of these proposals could potentially expose user data and undermine system reliability without adequate privacy protocols. Therefore, developing robust privacy protocols is crucial to ensure the privacy and security of users.

1.2. Contributions

To address the limitations of existing literature, our paper presents a new and innovative decentralized privacy-preserving user authentication mechanism for CIPSs. Our solution leverages attribute-based authentication, zero-knowledge proofs, and a revocation scheme based on lifetime to provide advanced security features that are specifically tailored to the needs of CIPSs. In particular, our contributions include:

1. We propose a novel BLE-based authentication scheme that offers several key benefits, including anonymized location data sharing, decentralized authentication, and offline revocation. Our security analysis demonstrates the robustness and effectiveness of our scheme.
2. We conduct extensive evaluations of our protocol in various environments with different restrictions, demonstrating its versatility and effectiveness in real-world scenarios.

3. We define a standardized BLE advertising packet format that simplifies the implementation of our protocol on unsupported platforms, enabling a wider range of devices to benefit from our innovation.
4. We present comprehensive comparative results of our protocol's execution on commonly used devices, showcasing its superior performance and efficiency.
5. We provide a thorough comparative analysis with other existing location-based schemes, providing insights into the benefits and drawbacks of each approach.

Our contributions represent a significant step forward in the field of decentralized authentication and highlight the potential for BLE-based authentication to revolutionize the way CIPSs operate in a secure and decentralized manner. The rest of this paper is organized as follows. Section 2 outlines the cryptographic preliminaries used to construct our protocol. Section 3 defines the cryptographic scheme and the entities involved. Section 4 furnishes a comprehensive and overarching discussion of the security analysis, while a more formal and rigorous analysis of the security properties is presented in Appendix. Section 5 introduces the use cases. Section 6 describes the development of the protocol on different wearable devices. Section 7 displays the benchmarks of the protocol execution. Section 8 discusses the comparative analysis of the proposed protocol with other existing protocols in the field. Finally, Section 9 reports the conclusions.

2. Cryptographic preliminaries

In this section, we recall the fundamental building blocks of our cryptographic scheme: the Diffie–Hellman (DH) key exchange protocol [34], the weak Boneh–Boyen (wBB) signature scheme [35], and the Sigma protocols [36].

2.1. Notation

The symbol ":" means "such that" and $|x|$ is the bit-length of x. The symbol $\mathcal{H}$ denotes a secure hash function. We write $a \in_R A$ when a is sampled uniformly at random from A. Let $\mathtt{GroupSetup}\ (1^\kappa)$ be an efficient algorithm that generates a group $\mathbb{G} = \langle g \rangle$ of prime order q, such that $|q| = \kappa$. Let $\mathbf{e}$ denote a bilinear map $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

2.2. Weak Boneh–Boyen signature

The weak Boneh–Boyen [35] signature scheme is a short pairing-based signature scheme that is demonstrably secure under the q-Strong Diffie–Hellman assumption in the standard model. The scheme permits fast message signing and is readily combinable with zero-knowledge proofs such that the knowledge of signed messages (and signatures themselves) may be proved anonymously, unlinkably, and untraceably. The following is a brief description of the wBB signature scheme:

- $(params, pk, sk) \leftarrow \mathtt{KeyGen}\ (1^\kappa)$: on the input of the system security parameter κ, the algorithm generates a bilinear group $params = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$, computes $pk = g_2^{sk}$ where $sk \in_r \mathbb{Z}_q$, and outputs $(params, pk)$ as the public key and sk as the private key.
- $(\sigma) \leftarrow \mathtt{Sign}\ (params, sk, m)$: on the input of the message $m \in \mathbb{Z}_q$, the system security parameters $params$ and the secret key sk, the algorithm outputs the signature of the message $\sigma = g_1^{\frac{1}{sk+m}}$.
- $(1/0) \leftarrow \mathtt{Verify}\ (params, pk, m, \sigma)$: on the input of the system security parameters $params$, the public key pk, a signature σ, and a message m, the algorithm returns 1 if and only if $\mathbf{e}(\sigma, pk) \cdot \mathbf{e}(\sigma^m, g_2) = \mathbf{e}(g_1, g_2)$ holds, i.e., the signature is valid, and 0 otherwise.

2.3. Sigma protocols

Σ-protocols [36] may be used to demonstrate the knowledge of secrets and the completeness of construction without disclosing more information. We employ the protocols outlined in [37] to demonstrate the knowledge of a discrete logarithm (the protocol $PK\{x : c = g^x\}$). These protocols may be converted into full zero-knowledge protocols [38]; hence, it is possible to demonstrate that they do not expose any further information than intended. The work by Fiat and Shamir [39] enables its non-interactive execution with computational security.

3. Cryptographic scheme

We propose an innovative decentralized attribute-based authentication protocol. This section discusses the entities involved, as well as the cryptographic design of the algorithms comprising the protocol.

To aid in the clarity and readability of this paper, we provide a table of symbols used throughout our cryptographic protocol. Table 1 defines each symbol and its associated meaning, allowing for a comprehensive understanding of the protocol's components.

Table 1
Table of symbols.

Symbol	Definition
$q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2$	Parameters for the selected pairing-friendly elliptic curve.
m_{ID}	Attribute with the user's identifier.
m_r	Attribute for revocation based on the week and year.
$sk_I, \mathcal{X}_0$	Key pair of the issuer (private and public keys).
$i_r, \mathcal{I}_r$	Shared key pair among system users (private and public keys).
$\sigma, \sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}}$	Cryptographic credential issued to the user.
$\hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}}$	Cryptographic credential randomized by the user.
ρ	Random number used to randomize the cryptographic credential.
ρ_κ, ρ_{ID}	Random numbers used to compute the protocol commitment and responses.
t_κ	Cryptographic commitment computed by the user.
e	Challenge used in the cryptographic protocol.
s_κ, s_{ID}	Responses obtained during the execution of the cryptographic protocol.
τ	Random number used to randomize the transaction identifier.
$\mathcal{R}$	Transaction identifier.
ψ	Alias of $\hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}}$.
π	Alias of e, s_κ, s_{ID}.
λ	Information transmitted or received in plaintext.
ξ	Information transmitted or received in ciphertext.
t'_κ	Cryptographic commitment reconstructed by the verifier.

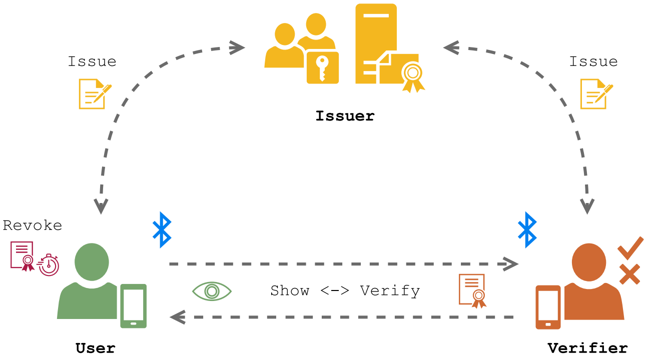


Fig. 1. Entities and algorithms constituting the proposed protocol.

3.1. Entities

The following entities comprise the system model presented in Fig. 1:

- *Issuer*: is responsible for issuing the personal attribute m_{ID} gathered in a cryptographic credential using the `Issue` algorithm. The personal attribute m_{ID} is the user identifier obtained during the system registration. The cryptographic credential is signed by the private key of the issuer sk_I. In our design, the issuer is also responsible for revoking invalid users from the system. This is done to simplify interactions with other entities. The revocation attribute m_r is additionally aggregated to the cryptographic credential.
- *User*: acquires the unique credential, including the attributes issued by the issuer, to gain access to the system or service. Using the `Show` algorithm, the user then anonymously demonstrates ownership of the attributes to the verifier. The user may additionally transmit information to the verifier in either plaintext or encrypted format.
- *Verifier*: utilizes the `Verify` algorithm to confirm the user's possession of the attributes. If the ownership of the attributes is successfully validated and the user's access has not been revoked, the verifier accepts the received information. If not, the information will be rejected.

3.2. Protocol specification

Following is a description of the algorithms, including their input and output parameters:

- $(params) \leftarrow$ `Setup` (1^κ): the algorithm receives the security parameter κ as input and generates the public system parameters. These parameters are a bilinear group with parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$ that satisfy $|q| = \kappa$.

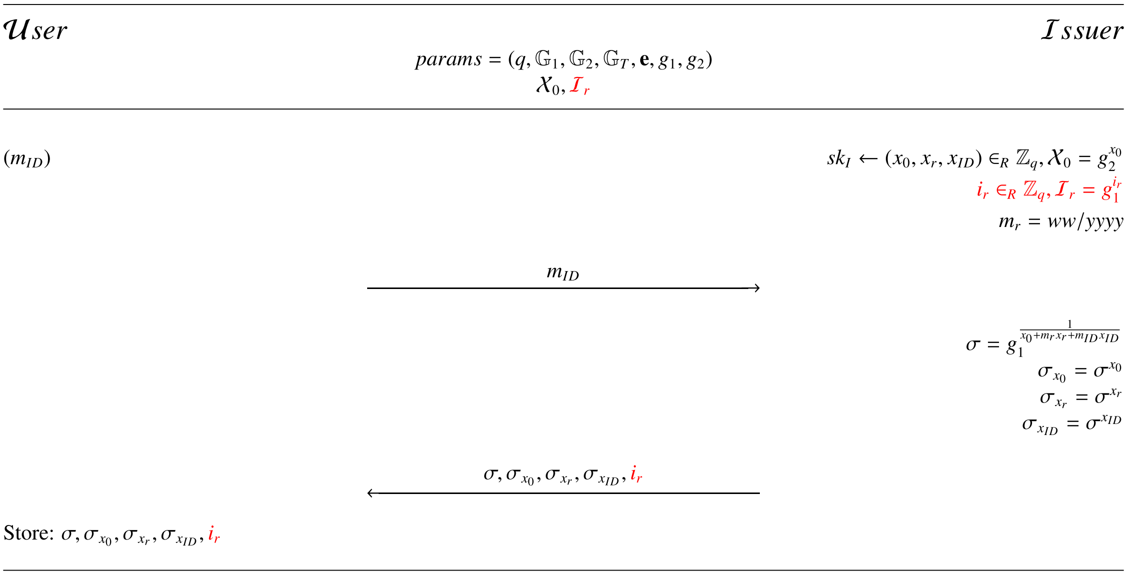

$\mathcal{U}ser$
$\mathcal{I}ssuer$
$params = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$
$\mathcal{X}_0, \mathcal{I}_r$
(m_{ID})
$sk_I \leftarrow (x_0, x_r, x_{ID}) \in_R \mathbb{Z}_q, \mathcal{X}_0 = g_2^{x_0}$
$i_r \in_R \mathbb{Z}_q, \mathcal{I}_r = g_1^{i_r}$
$m_r = ww/yyyy$
m_{ID}
$\sigma = g_1^{\frac{1}{x_0+m_r x_r+m_{ID} x_{ID}}}$
$\sigma_{x_0} = \sigma^{x_0}$
$\sigma_{x_r} = \sigma^{x_r}$
$\sigma_{x_{ID}} = \sigma^{x_{ID}}$
$\sigma, \sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}}, i_r$
Store: $\sigma, \sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}}, i_r$

Fig. 2. Definition of the `Issue` algorithm.

- $(sk_I, \mathcal{X}_0, i_r, \mathcal{I}_r) \leftarrow$ `KeyGen` $(params)$: the algorithm randomly selects the private keys $sk_I \leftarrow (x_0, x_r, x_{ID}) \in_R \mathbb{Z}_q$ and computes the issuer's public key $\mathcal{X}_0 = g_2^{x_0}$, based on the system parameters $params$. In addition, the algorithm also randomly generates the shared secret key $i_r \in_R \mathbb{Z}_q$, and calculates the shared public key $\mathcal{I}_r = g_1^{i_r}$, which will be utilized by system users. The algorithm outputs the issuer keys $(sk_I, \mathcal{X}_0, i_r, \mathcal{I}_r)$. The issuer runs the `KeyGen` algorithm.
- $(\sigma, \sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}}, i_r) \leftarrow$ `Issue` $(params, sk_I, m_r, m_{ID})$: the algorithm inputs the private keys of the issuer $sk_I \leftarrow (x_0, x_r, x_{ID}) \in_R \mathbb{Z}_q$, the revocation attribute m_r, and the user attribute m_{ID}. The revocation attribute m_r is set to $ww/yyyy$ in our implementation, which means that m_r is the current year and the week of that year. The algorithm is run between the user and the issuer and is shown in Fig. 2. First, the user sends its attribute m_{ID} to the issuer. Next, the issuer signs the attributes m_r and m_{ID} with the secret keys as $\sigma = g_1^{\frac{1}{x_0+m_r x_r+m_{ID} x_{ID}}}$ and computes auxiliary values $\sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}}$, where $\sigma_{x_0} = \sigma^{x_0}$, $\sigma_{x_r} = \sigma^{x_r}$, and $\sigma_{x_{ID}} = \sigma^{x_{ID}}$. The algorithm outputs the cryptographic credential σ and auxiliary values $(\sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}})$ to the user. The algorithm also securely provides the user with the shared secret key i_r. This key must remain secure on the device.
- $(\mathcal{R}, \psi, \pi, \lambda$ or $\xi) \leftarrow$ `Show` $(params, m_r, m_{ID}, \sigma, \sigma_{x_0}, \sigma_{x_r}, \sigma_{x_{ID}}, \mathcal{I}_r, \lambda, time\,stamp)$: the algorithm receives the timestamp from the verifier, the revocation attribute m_r, and the user attribute m_{ID} as inputs. To prevent replay attacks, the user is required to verify that the received timestamp is no more than two seconds earlier or later than the current time. In our implementation, we assume all users have an Network Time Protocol (NTP)-synchronized date and time on their devices. The algorithm outputs the transaction identifier $\mathcal{R}$, the randomized user credentials ψ, the cryptographic proof of possession of the attributes π, and the data to be transferred in an encrypted format ξ. Nonetheless, if the user executes the algorithm in a public environment, data is transmitted in plaintext λ, i.e., unencrypted. Thus, the output of the algorithm is λ and not ξ. This characteristic is described in full in Section 5. Users execute the `Show` algorithm. Fig. 3 depicts a comprehensive explanation of the `Show` algorithm. The user begins by randomizing their credentials. The user then calculates the commitment t_κ and uses the hash of t_κ as the symmetric key to encrypt the data that needs to be sent. Lastly, the user computes a proof of knowledge for all the attributes in the credential. Note that when we require encrypted data transmission, we utilize the red-highlighted formulae. The $\mathcal{R}$ value is the transaction identifier, but it is also required for the verifier to construct the decryption key $\mathcal{H}(t'_\kappa)$. If the information is to be sent as plaintext λ, we can omit the operations marked in red and send the data straight.
- $(0/1) \leftarrow$ `Verify` $(params, timestamp, \mathcal{X}_0, i_r, \mathcal{R}, \psi, \pi, \lambda$ or $\xi)$: the algorithm inputs the timestamp previously generated for the user, the shared secret key i_r, the transaction identifier $\mathcal{R}$, the randomized user credentials ψ, the cryptographic proof of possession of the attributes π, and the user data in an encrypted format ξ. Nonetheless, if the verifier executes the algorithm in a public environment, data is received in plaintext λ, i.e., unencrypted. Thus, the input of the algorithm is λ and not ξ. This characteristic is described in full in Section 5. Verifiers execute the `Verify` algorithm. Fig. 3 depicts a comprehensive explanation of the `Verify` algorithm. The verifier begins by recalculating the commitment t'_κ. The verifier then uses the hash of t'_κ as the symmetric key to decrypt the user data. In addition, the verifier also uses t'_κ to create the cryptographic hash $\mathcal{H}(t'_\kappa, \hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}}, timestamp, \lambda)$ and checks that the value of e received from the user matches. Lastly, the verifier computes a bilinear pairing to ensure that the randomized credentials correspond to a genuine user of the system and were emitted by the issuer. Note that when we require encrypted data reception, we utilize the red-highlighted formulae. Additionally, the shared secret key i_r must remain secure on the device to prevent non-system users from accessing the data. If the information is to be received as plaintext λ, we can omit the operations marked in red to facilitate the calculation.

$\mathcal{U}ser$ $\mathcal{V}erifier$

$$params = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$$
$$\mathcal{X}_0, \mathcal{I}_r$$

$\lambda \leftarrow (data\ to\ transmit)$ $m_r = ww/yyyy$

$$\xleftarrow{\quad timestamp \quad}$$

$\tau \in_R \mathbb{Z}_q$
$\mathcal{R} = g_1^{\tau}$
$\rho, \rho_{\kappa}, \rho_{ID} \in_R \mathbb{Z}_q$
$\hat{\sigma} = \sigma^{\rho}, \hat{\sigma}_{x_0} = \sigma_{x_0}^{\rho}$
$\hat{\sigma}_{x_r} = \sigma_{x_r}^{\rho}, \hat{\sigma}_{x_{ID}} = \sigma_{x_{ID}}^{\rho}$
$t_{\kappa} = g_1^{\rho_{\kappa}} \sigma_{x_{ID}}^{\rho_{ID}\rho} \mathcal{I}_r^{\tau}$
$\xi = Enc_{\mathcal{H}(t_{\kappa})}(\lambda)$
$e = \mathcal{H}(t_{\kappa}, \hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}}, timestamp, \lambda)$
$s_{\kappa} = \rho_{\kappa} + e\rho$
$s_{ID} = \rho_{ID} - em_{ID}$
$\psi = (\hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}})$
$\pi = (e, s_{\kappa}, s_{ID})$

$$\xrightarrow{\quad \mathcal{R}, \psi, \pi, \lambda\ or\ \xi \quad}$$

$t'_{\kappa} = g_1^{s_{\kappa}} \hat{\sigma}_{x_0}^{-e} \hat{\sigma}_{x_r}^{-em_r} \hat{\sigma}_{x_{ID}}^{s_{ID}} \mathcal{R}^{i_r}$
$\lambda = Dec_{\mathcal{H}(t'_{\kappa})}(\xi)$
$e \overset{?}{=} \mathcal{H}(t'_{\kappa}, \hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}}, timestamp, \lambda)$
$\mathbf{e}(\hat{\sigma}_{x_0}, g_2) = \mathbf{e}(\hat{\sigma}, \mathcal{X}_0)$

Fig. 3. Definition of the `Show` and `Verify` algorithms.

- `Revoke`: the algorithm has no input parameters or output. In addition, it is not directly executed by any entity in the system. The revocation model is based on the expiration of an epoch. In our implementation, the revocation attribute m_r indicates the week of the current year. Thus, when the week number changes, the credentials of all users are automatically revoked. The users must request reissuance from the issuer once their credentials have expired. Note that the issuer's shared keys are regenerated weekly to prevent revoked users who fail to renew their credentials from continuing to read data. Therefore, every week the i_r and $\mathcal{I}_r$ keys are invalidated, and new shared keys are produced.

The flowcharts presented in Fig. 4 provide a high-level definition of the `Show` and `Verify` algorithms and are a valuable tool for facilitating a comprehensive understanding of the protocols, their underlying mechanisms, and the sequence of steps involved in their execution. By visually presenting the key components of each algorithm and the relationships between them, the flowcharts enable readers to quickly grasp the fundamental concepts of our cryptographic protocol.

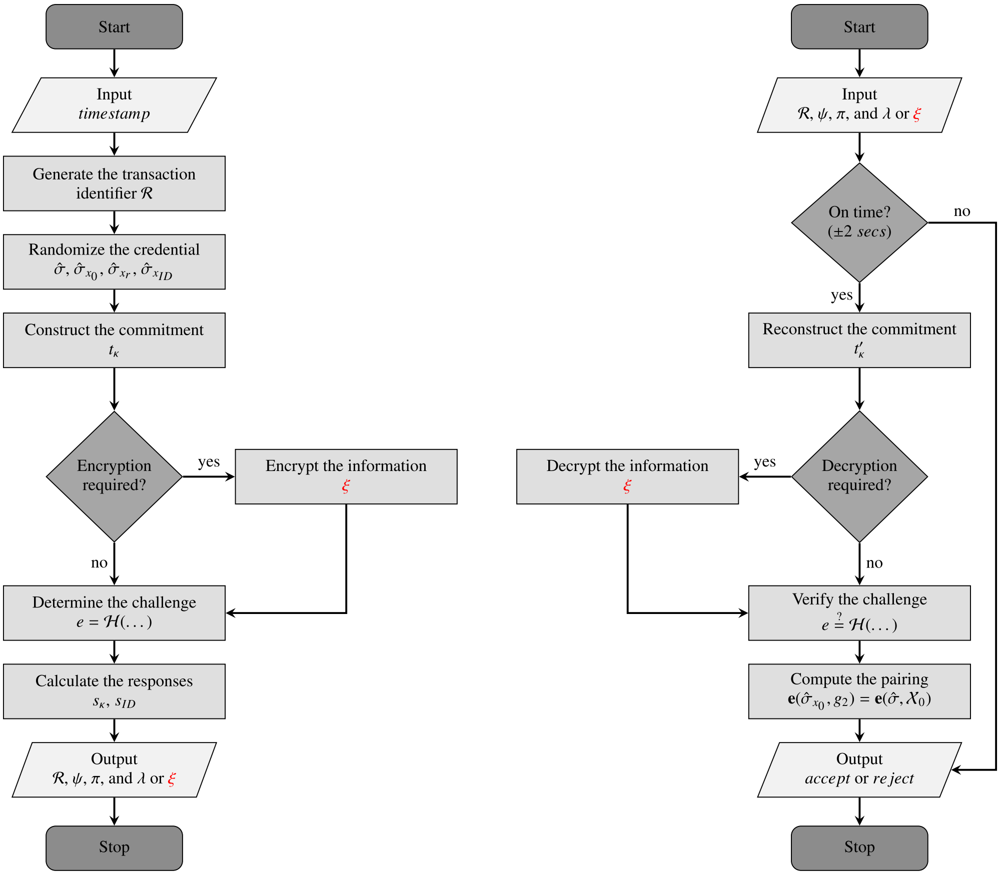
4. Security and privacy discussion

This section provides an examination of the security and privacy requirements of our novel decentralized ABA protocol for CIPSs. The protocol relies on a robust authentication mechanism to ensure secure interactions among users. Our discussion substantiates the critical security, privacy, and functionality properties that the protocol provides. This analysis demonstrates that the proposed protocol satisfies these security and privacy requirements, establishing it as a dependable solution for securing CIPSs. For a formal and more in-depth analysis, please refer to Appendix.

4.1. Required properties

We examine the essential security, privacy, and functionality properties, including anonymity, unlinkability, and untraceability; confidentiality and integrity; correctness; key-parameter consistency; as well as unforgeability, completeness, soundness, and zero-knowledge, that the protocol must provide to guarantee the security and privacy of CIPSs.

Anonymity, *unlinkability*, and *untraceability* are essential properties to ensure the user's privacy. The anonymity property ensures that a party's identity is hidden or kept confidential during a protocol's execution, preventing it from being linked to any exchanged


Start
Input
$timestamp$
Generate the transaction identifier $\mathcal{R}$
Randomize the credential
$\hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}}$
Construct the commitment
t_κ
Encryption required?
yes
Encrypt the information
ξ
no
Determine the challenge
$e = \mathcal{H}(\dots)$
Calculate the responses
s_κ, s_{ID}
Output
$\mathcal{R}, \psi, \pi$, and λ or ξ
Stop
Start
Input
$\mathcal{R}, \psi, \pi$, and λ or ξ
On time? (± 2 $secs$)
no
yes
Reconstruct the commitment
t'_κ
Decryption required?
yes
Decrypt the information
ξ
no
Verify the challenge
$e \overset{?}{=} \mathcal{H}(\dots)$
Compute the pairing
$\mathbf{e}(\hat{\sigma}_{x_0}, g_2) = \mathbf{e}(\hat{\sigma}, X_0)$
Output
$accept$ or $reject$
Stop

(a) High-level Show algorithm.

(b) High-level Verify algorithm.

Fig. 4. High-level definition of the Show and Verify algorithms.

information. The unlinkability property guarantees that different protocol executions cannot be linked to the same party or identity, thereby preventing any correlation of the party's actions or information across multiple protocol executions. Lastly, the untraceability property ensures that a party's actions or information exchanged during a protocol execution cannot be traced back to the party, avoiding any identification of the party's actions through any means, including network monitoring or traffic analysis. The properties of anonymity, unlinkability, and untraceability are achieved in our protocol through the utilization of standard zero-knowledge protocols. The randomized user credential in each protocol execution ensures the prevention of user identification or tracing, which further enhances the security and privacy of the protocol. These properties have been formally proven in Proposition 5 of our security and privacy analysis.

Confidentiality and *integrity* properties are fundamental requirements in secure communication protocols. Confidentiality ensures that sensitive information exchanged during a protocol's execution remains confidential and protected from unauthorized disclosure, and only authorized parties can access or read it. Integrity guarantees that the information remains trustworthy throughout the protocol execution and is not tampered with or altered in any way. Our protocol achieves confidentiality and integrity by using standard cryptographic primitives such as Advanced Encryption Standard (AES) and Secure Hash Algorithm 3 (SHA-3), which are widely used and trusted in the field. Additionally, our protocol offers flexibility to the user by providing the option to encrypt the information exchanged during the protocol execution, ensuring its confidentiality and protection from unauthorized access. Nevertheless, the user can also decide to share the information in plaintext, making it available to everyone without compromising its integrity. These properties have been formally proven in Proposition 6 of our security and privacy analysis.

Correctness is of the utmost importance, as it ensures that the information exchanged during a protocol's execution is accurate and in line with the intended specification. This property guarantees that the protocol achieves its desired goal while preventing errors or misunderstandings in the exchange of information. Our protocol guarantees correctness through the use of advanced cryptographic techniques such as commitment reconstruction, hash functions, and pairings. These techniques help to ensure that the information exchanged during the protocol execution is accurate and reflects the state of the system. This is accomplished by verifying the integrity of the data to ensure that it has not been tampered with or altered in any way. This property has been formally proven in Proposition 2 of our security and privacy analysis.

Key-parameter consistency is crucial to ensure the security of the protocol, as it guarantees that the keys and parameters used during a protocol's execution are valid and consistent. This property prevents the protocol from relying on faulty or invalid keys or parameters that could be exploited by an attacker. Our protocol ensures key-parameter consistency by generating keys and parameters from truly random sources, which makes them both unpredictable and secure. We also validate their authenticity and consistency before using them in the protocol to ensure that only legitimate and properly generated keys and parameters are used. This approach helps prevent attacks that exploit weaknesses or vulnerabilities in the keys and parameters used in the protocol. This property has been formally proven in Proposition 4 of our security and privacy analysis.

Unforgeability, *completeness*, *soundness*, and *zero-knowledge* are indispensable to the security and privacy of our protocol. The unforgeability property ensures that only the authorized entity can produce valid signatures and prevents adversaries from forging signatures without access to the signer's private key. The completeness property is essential to ensure that a protocol execution is comprehensive and does not leave any loopholes or opportunities for attacks, ultimately guaranteeing the protocol's overall security and reliability. This property ensures that all valid inputs are accepted and that the protocol produces a valid output, leaving no room for ambiguity or incomplete execution. The soundness property ensures that a protocol execution provides verifiable and unambiguous evidence that cannot be tampered with or disputed. It guarantees that any invalid inputs are rejected and further strengthens the protocol's security and integrity. Finally, the zero-knowledge property ensures that parties can prove their knowledge of secrets without revealing any information beyond what is strictly necessary. Our protocol satisfies those properties through the use of standard signature and zero-knowledge protocols. Specifically, the protocol leverages well-established techniques such as the weak Boneh–Boyen digital signature, the Fiat-Shamir transform, and the Schnorr identification scheme to achieve these goals. This ensures that the protocol is both secure and efficient while also providing the necessary guarantees to protect the privacy of users in the system. These properties have been formally proven in Propositions 1, 2, and 3 of our security and privacy analysis.

5. Use cases for the proposed scheme

This section presents two use cases for the decentralized attribute-based authentication protocol. In addition, we introduce the privacy-enhanced mode. In this mode, users and verifiers may select whether to actively participate in the system or not.

5.1. Public environments

In this first scenario, we assume users are present in an environment with public access. For instance, shopping malls, universities, and hospitals. Workers, as well as customers, students, and patients, have unrestricted access to these buildings. In this scenario, the data can be transmitted authenticated but unencrypted, i.e., in plaintext. Users can download the mobile application and decide whether to register with the system. An unregistered user will be able to obtain positioning data that has been authenticated by valid users of the system, but will be unable to transmit data because they lack valid credentials. In contrast, registered users will also be able to transmit positioning information. Executing the `Show` and `Verify` algorithms, depicted in Fig. 3, enables the transmission of authenticated information in plaintext, i.e., unencrypted. These algorithms must be executed without performing the red calculations. For a high-level overview of the processes involved, please refer to the accompanying flowcharts of the `Show` algorithm in Fig. 4(a) and the `Verify` algorithm in Fig. 4(b).

Fig. 5 depicts various types of users and verifiers participating in a public environment, i.e., positioning them in a shopping center. The green figures represent users who have registered with the system and are permitted to transmit positioning information. The blue figures represent users who are not registered with the system and can therefore only receive and validate the information from the green users, meaning they can only act as verifiers requesting positioning information. Verifiers, or users who have requested positioning data to improve their location, are represented by the orange figures. Finally, users and verifiers operating in privacy-enhanced mode are represented by the gray and red figures, respectively. This mode is described in detail in Section 5.3.

5.2. Private environments

In this second scenario, we assume users are present in an environment with non-public access. For instance, military facilities, critical infrastructures, or private corporations. Access to these buildings is restricted, and only employees have permission to enter. In this scenario, data should be transmitted authenticated and encrypted. Users are already registered within the system and must only download the mobile application. All users will possess valid credentials, allowing them to freely transmit positioning data. Executing the `Show` and `Verify` algorithms, depicted in Fig. 3, enables the transmission of authenticated and encrypted information. These algorithms must be executed, including the red calculations. For a high-level overview of the processes involved, please refer to the accompanying flowcharts of the `Show` algorithm in Fig. 4(a) and the `Verify` algorithm in Fig. 4(b).

This use case is similar to the one shown in Fig. 5, but all users must be registered and part of the system, so the blue figures are eliminated.

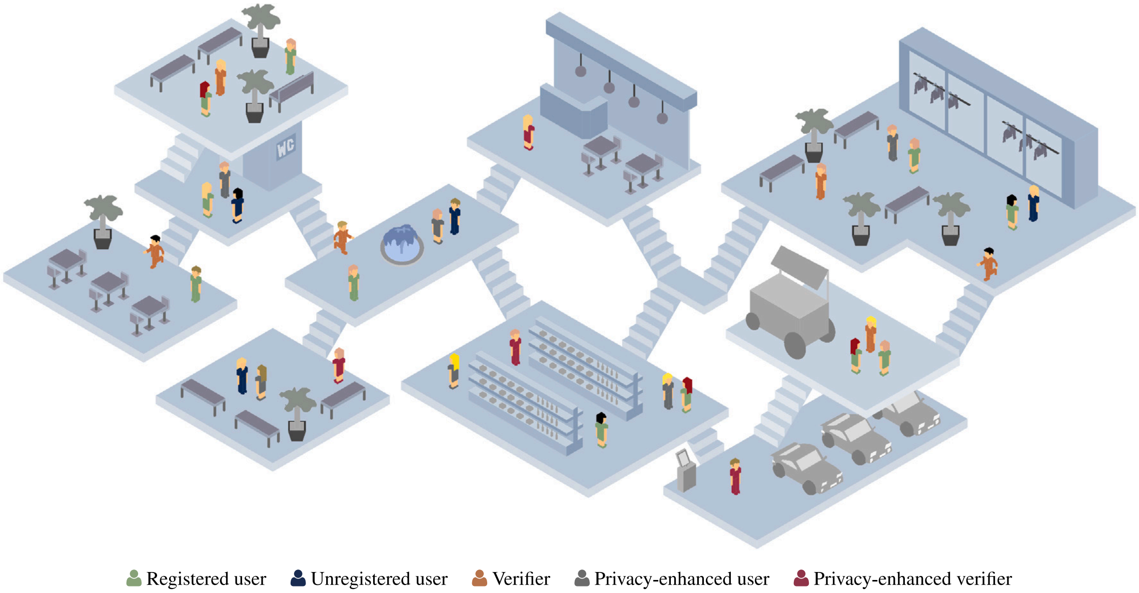


Fig. 5. Practical use case of the decentralized attribute-based authentication protocol.

5.3. Privacy-enhanced mode

This mode of operation allows users and verifiers to enhance their privacy by blocking the transmission of information but permitting its reception. It is useful for improving the privacy of users in environments with few devices and where their identities are easily discernible.

Users participating in the system respond to verifier requests and send their positioning data in plaintext or encrypted. However, when they are using the privacy-enhanced mode, users do not respond to verifiers' requests and therefore do not interact with the system.

Verifiers participating in the system request positioning data from nearby users. When they receive the information, they can validate it and use it to refine their location. However, when they are using the privacy-enhanced mode, verifiers do not request information from users. They use the information received from users but requested by other verifiers. Therefore, they can validate and utilize the information received from users to improve their location while remaining hidden.

6. Implementation details

This section describes the development process and summarizes the main key points considered during the design of the application. Table 2 outlines the devices we employed and their hardware and software specifications. The device selection process was intentionally agnostic and unbiased, with devices chosen at random to ensure a diverse sample. This strategy ensured that both legacy and modern devices were included in our implementation, providing compatibility with a broad range of hardware and reinforcing the protocol's viability in real-world scenarios. The application was designed for several platforms, including single-board computers, smartphones, smartwatches, and microcontrollers. Such heterogeneous device types are common in Internet of Things (IoT) ecosystems and are representative of the different use cases that our protocol can support. While smartphones and smartwatches are ideal for obtaining positioning data actively by users, single-board computers and microcontrollers are better suited for industrial settings. They can enable autonomous vehicle positioning or enhance the location sensing capabilities of other devices, further extending the reach and flexibility of our solution.

We divided the implementation of the application into three components: *(i)* the *Libre Collaborative Indoor Positioning (LibreCIP)* library for core functionality; *(ii)* the *device wrappers* to ensure compatibility with the selected IoT devices (see Table 2); and *(iii)* the BLE integration for transmitting and receiving data using Bluetooth Low Energy.

6.1. LibreCIP

The cornerstone of the application is our *LibreCIP* library, a purpose-built software package that offers the complete implementation of the protocol. *LibreCIP* includes a highly optimized suite of advanced cryptographic functions such as user authentication and verification, data encryption and decryption, etc. In addition, it contains compression and decompression routines that enable efficient data transmission and storage while preserving the security and privacy of user information. The library is designed from scratch to offer unparalleled performance and security on a wide range of devices. It is written in the C programming language and

Table 2
Hardware and software specifications of the devices.

Device	CPU	OS	RAM
Single-board computers			
Raspberry Pi 4 Model B	ARM Cortex-A72	Raspberry Pi OS	4 GB
Smartphones			
Samsung Galaxy S21+ 5G	Exynos 2100	Android 11	8 GB
Samsung Galaxy S20 FE	Exynos 990	Android 10	6 GB
Samsung Galaxy A52	SDM720G	Android 11	6 GB
Samsung Galaxy A32	MTK D720 Dual + Hexa	Android 11	4 GB
Samsung Galaxy S8	Exynos 8895	Android 9	4 GB
iPhone 11	A13 Bionic	iOS 16.2	4 GB
iPhone XS Max	A12 Bionic	iOS 16.2	4 GB
PinePhone Pro	Rockchip RK3399S 64 bit SoC	Arch Linux ARM	4 GB
Smartwatches			
Huawei Watch 2	Snapdragon 2100	Android Wear 2	768 MB
Apple Watch Series 5	Apple S5 (64-bit dual-core)	watchOS 9.2	1 GB
PineTime	ARM Cortex-M4F	RIOT 2022.10	64 kB
Microcontrollers			
Arduino Nano 33 BLE	NINA-b3 (nRF52840)	RIOT 2022.10	256 kB
Arduino Nano 33 IoT	ATSAMD21	RIOT 2022.10	32 kB

Table 3
Comparison of the zlib, xz, and lz4 compression algorithms.

Compression algorithm	Data size [B]	Compressed data size [B]	Compression ratio [%]
`zlib v1.2.13`	294	297	1.010
`xz v5.4.0`	294	299	1.017
`lz4 v1.9.4`	294	284	0.966

relies on several third-party libraries. Specifically, we utilized `mcl` [40] and `relic-toolkit` [41] for elliptic curve cryptographic support; `lz4` [42] for providing data compression and decompression support; and `crypto`, the native cryptographic library of RIOT [43], for data encryption and decryption, as well as for cryptographic hash algorithm support. Note that we avoided the use of costly and difficult-to-compile libraries on devices with limited resources to enable code portability and easy integration into wearable devices. Consequently, we exclusively utilized libraries that are compatible with the RIOT operating system [44]. The library used to implement the cryptographic core varies based on the device. We used `mcl` for Apple, Android, Raspberry Pi, and PinePhone Pro devices since it offers faster execution speeds. In contrast, we utilized the `relic-toolkit` for the PineTime and microcontrollers. The cryptographic backend can be selected during compilation.

The protocol was designed and implemented using elliptic curve cryptography. Specifically, we utilized the Barreto-Naehrig 254-bit (BN254) curve supplied by the `mcl` and the `relic-toolkit` libraries. Uncompressed points occupy 64 bytes and compressed points 33 bytes with this curve size. A scalar integer has a size of 32 bytes. We explored transmitting the points in their compressed form to reduce the size of the data.

Likewise, we examined data compression to further reduce the size of the information to be transmitted. We evaluated a number of open-source compression algorithms to select the one with the highest compression ratio for our protocol. Table 3 compares various compression algorithms, including `zlib` [45], `xz` [46], and `lz4` [42]. However, when working with such a small amount of data, there is no discernible difference between the benchmarked data compression algorithms. In addition, data size may grow owing to the inclusion of compression headers. Therefore, we chose `lz4` because the RIOT operating system natively supports it and because it is the only algorithm that reduces the size of the original data. I.e., the compressed data size is lower than the original data size.

Lastly, we used the RIOT `crypto` library to encrypt and decrypt data and calculate cryptographic hashes. Specifically, we employed the AES algorithm in Cipher Block Chaining (CBC) mode with 128-bit keys for encrypting and decrypting and the SHA-3 function with 256-bit digests.

6.2. Device wrappers

The wrappers for each device are written in their respective native languages. They consist of the User Interfaces (UIs) of the devices specified in Table 2 as well as the BLE routines and libraries required for communication. The UIs are designed based on the device type. For Android, iOS, Wear OS, and watchOS devices, we created Graphical User Interfaces (GUIs). In contrast, we developed Command Line Interfaces (CLIs) for the Raspberry Pi and PinePhone Pro. We did not create UIs for the microcontrollers or the PineTime.

We used the Java SE Development Kit 17.0.6 to create the Android and Wear OS apps. The Android Native Development Kit (NDK) enables us to develop portions of our application in native code using languages such as C and C++. This permits us

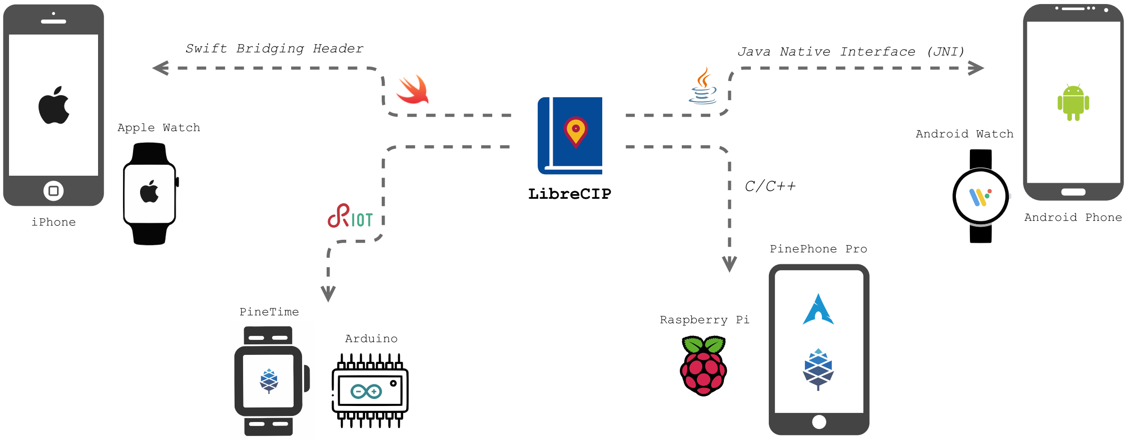


Fig. 6. Interoperability between *LibreCIP* and different devices.

to invoke the *LibreCIP* library functions via Java Native Interface (JNI). We used Swift to create the apps for iOS and watchOS. This language permits the execution of C-written functions through a bridge. This enables us to invoke *LibreCIP* library functions. The PinePhone Pro and Raspberry Pi applications were developed in C, so they can use the *LibreCIP* library directly without additional layers. The PinePhone Pro utilizes the Arch Linux ARM operating system with the *sxmo* desktop. We merely created a command-line program to assess its performance, as it is not a device ready for end users. The Raspberry Pi application is also a command-line application. We used RIOT for the microcontroller's applications. These devices lack display and output peripherals; therefore, we pondered creating applications that would always respond to user requests to improve their location accuracy and occasionally request their location to refine the precision of the device's own location. The applications are written in C and run when the devices are powered on. Although the PineTime device has a display, we did not design a GUI; hence, we also considered running the same type of application. Fig. 6 depicts the different device wrappers and the technologies used by each to call *LibreCIP* library functions.

6.3. Bluetooth low energy integration

Our protocol transfers data using BLE advertising packets. Bluetooth 4.2 [47] and earlier versions specified a 31-byte payload size for a single BLE advertising packet. Bluetooth 5.0 [48], on the other hand, introduced a significant enhancement by expanding the capacity of advertising packets. The advertising payload may include up to 254 bytes with Low-Energy Advertising Extensions. Since it is impossible to fit all data into 31 bytes, we contemplated utilizing Bluetooth 5.0. Unfortunately, Bluetooth 5.0 and the Low-Energy Advertising Extension are not supported by all actual devices. Therefore, we designed a packet format that is compatible with Bluetooth 4.2 and previous versions. This packet structure permits the transmission of information by fragmenting it into smaller packets and chaining them together, enabling other users to identify the number and sequence of packets to be received. Fig. 7 depicts the BLE packet structure we designed for transmitting our protocol data.

The first packet comprises a 7-byte header and 24 bytes of data. The `Packet Id` field corresponds to the first three bytes of the application's UUID hash value, i.e., $\mathcal{H}(\texttt{UUID}) = \texttt{0x414...f5e}$. The `Packet Next Id` field corresponds to the first three bytes of the next packet's hash value, i.e., $\mathcal{H}(\texttt{Packet}_1) = \texttt{0x141...5ef}$. The `Packet Count` field indicates the total number of packets. The `Encryption` field specifies whether the data is in plaintext or encrypted format. Intermediate packets maintain the `Packet Id` and `Packet Next Id` fields but lack the `Packet Count` and `Encryption` fields. The `Packet Id` field in these packets is equal to the first three bytes of the packet's hash. The final packet keeps the `Packet Id` field but omits the `Packet Next Id` field. The `Packet Id` field in this packet is also equal to the first three bytes of the packet's hash.

7. Experimental results

This section presents the results of the implementation of our protocol on different types of devices. To ascertain the feasibility of our approach and evaluate the performance and speed of the algorithms, we conducted several experiments, benchmarking the entire protocol. Fig. 8 illustrates the results of the execution. The figure depicts the benchmarks in milliseconds and includes the protocol run times and BLE communication overhead.

To interpret these results, it is important to consider the limitations of the study. For example, the times were taken in a laboratory, away from environments crowded with transmitting devices, which could impact the generalizability of the results. Nevertheless, these results suggest that the protocol may be a promising option for protecting privacy in current CIPSs. Future studies should seek to replicate these results in larger and more diverse environments to fully understand their performance.

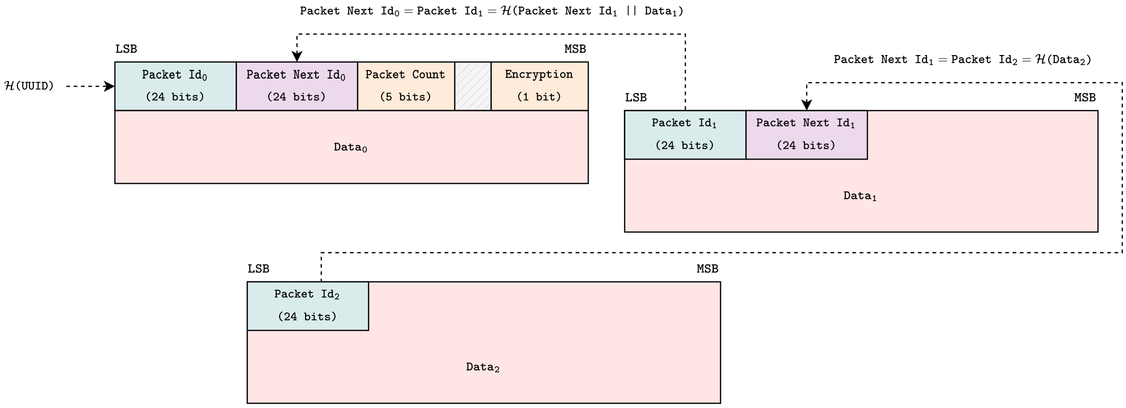

Packet Next Id$_0$ = Packet Id$_1$ = $\mathcal{H}$(Packet Next Id$_1$ || Data$_1$)
LSB
MSB
$\mathcal{H}$(UUID)
Packet Id$_0$ (24 bits)
Packet Next Id$_0$ (24 bits)
Packet Count (5 bits)
Encryption (1 bit)
Data$_0$
Packet Next Id$_1$ = Packet Id$_2$ = $\mathcal{H}$(Data$_2$)
LSB
MSB
Packet Id$_1$ (24 bits)
Packet Next Id$_1$ (24 bits)
Data$_1$
LSB
MSB
Packet Id$_2$ (24 bits)
Data$_2$

Fig. 7. Structure of the BLE advertising packets.

It is clear from the data that the smartphones, particularly the Samsung Galaxy S21+ 5G and iPhone 11, have the fastest computation times, with values around 1.3 ms and 2.9 ms for the Show and Verify algorithms, respectively. On the other hand, the Arduino Nano 33 BLE and Arduino Nano 33 IoT, which are microcontroller boards designed for IoT applications, have much slower computation times, with values in the range of 197.345 ms to 329.624 ms. This difference in performance is largely because microcontroller boards have less processing power and memory resources compared to smartphones and smartwatches, which are equipped with more advanced and powerful hardware.

It is notable that all popularly used smartphones and smartwatches maintain Show and Verify algorithm processing times below 15 ms. The quickest device executes the Show algorithm in 1.287 ms, while the slowest does it in 9.154 ms. In contrast, the fastest execution of the Verify algorithm requires 2.896 ms, and the slowest one requires 12.591 ms. The Raspberry Pi 4 benchmarks are omitted from the figure due to their negligibility. This device's execution time is on the order of μs. Show runs in 0.013 ms, whereas Verify executes in 0.015 ms. The communication overhead is 26.371 ms.

It is also worth noting that the Huawei Watch 2, Samsung Galaxy S8, PineTime, and Arduino Nano devices have the slowest transmission times, compared to the top-performing smartphones, with values over 30 ms. This can be attributed to the fact that these devices are equipped with older versions of Bluetooth technology or, as in the case of the PineTime, are not primarily designed for BLE communication.

The error bars in Fig. 8 represent the variability in the measurements obtained from multiple executions of our application on different devices. We observe that the smartphones in our study exhibit larger errors than the microcontrollers. This can be attributed to the fact that smartphones run full-fledged operating systems with multiple concurrent processes and services, which introduces more variability in the results. In contrast, the microcontrollers run a single application without sharing execution power, resulting in less variability and hence smaller errors that are almost imperceptible. Therefore, the difference in the complexity of the execution environment can explain the observed variability in our results.

Finally, during the development phase, we employed the Energy Profiler tool in Android Studio to evaluate the effect on energy consumption for Android applications. In addition, Xcode presents a comparable solution for iOS and watchOS devices through its "Instruments" tool. These tools offer developers a comprehensive analysis of application performance, memory utilization, and energy consumption, providing them with the necessary information to identify any potential problems and enhance the performance of their applications. To our satisfaction, the results indicated that there was no adverse effect on energy utilization.

8. Discussion

The findings presented in the preceding section demonstrate the efficiency of the proposed protocol across a range of devices, including wearables and low-power devices. This section provides a qualitative comparison of the proposed solution with other existing protocols in the field, highlighting its unique features and advantages.

Table 4 provides a comparison of key features between our proposed protocol and two existing solutions, Bellrock and FedLoc. Our protocol outperforms both Bellrock [28] and FedLoc [29] in terms of ensuring data authenticity, providing anonymity protection, supporting revocation, adopting serverless authentication architecture, achieving beacon-independent localization, supporting wearables and IoT devices, and ensuring scalability. Our protocol also preserves privacy, which is an essential characteristic for any secure and privacy-preserving localization system.

The protocol we propose excels at ensuring data authenticity by implementing a robust authentication mechanism that verifies the identity of users before accepting any location data. This supplementary security measure plays a critical role in safeguarding both the precision and integrity of the localization system by thwarting any attempts to generate counterfeit or tampered positioning data. On the other hand, both Bellrock and FedLoc lack a mechanism to authenticate the location data, making them vulnerable to tampering and impersonation attacks.

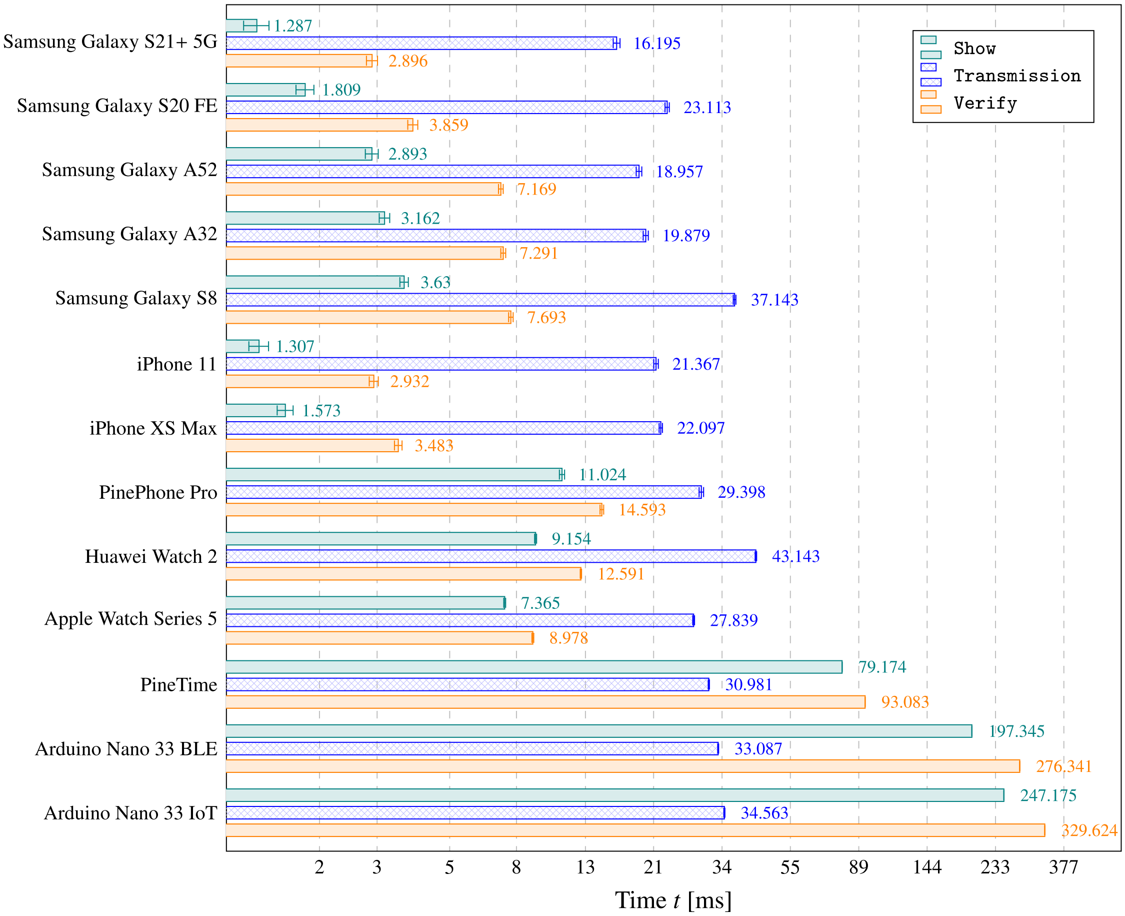


Fig. 8. Speed comparison of the Show and Verify algorithms, and the transmission overhead.

To protect user anonymity, the cryptographic scheme we suggest employs randomized user credentials with every transmission, thereby ensuring that the identities of users remain concealed. Bellrock similarly presents itself as a strong privacy-preserving option by leveraging various techniques to generate pseudo-anonymous identifiers that shield user identities. Conversely, FedLoc neglects the need for anonymity protection and, therefore, poses potential risks to user identity disclosure. The significance of anonymity in the context of localization systems cannot be overstated, and in this respect, our protocol and Bellrock have a clear advantage over FedLoc. On the other hand, it is worth noting that all three schemes share the common goal of preserving the privacy of positioning data. This is a crucial feature for any secure and privacy-preserving localization system, as users' location data is often sensitive and must be protected against unauthorized access. All solutions provide measures to ensure that users' positioning data remains private, including encryption and secure communication protocols. By preserving privacy, these schemes enable users to benefit from the advantages of LBSs without sacrificing their privacy.

By adopting a serverless authentication architecture, the cryptographic protocol we introduce stands out from both Bellrock and FedLoc. Both Bellrock and FedLoc rely on a centralized server, which could be a potential vulnerability. In contrast, our protocol does not require a centralized server, reducing the risk of a single point of failure and making it more resistant to attacks, thus offering a more secure and reliable solution.

The installation and maintenance of numerous beacons for accurate localization can be a complex and expensive process in beacon-dependent solutions. Unlike beacon-dependent solutions, the protocol we advance and FedLoc offer a beacon-independent approach, eliminating the need for numerous physical beacons. This not only simplifies the infrastructure requirements but also increases the flexibility of deployment across various environments. Moreover, both our cryptographic scheme and FedLoc exhibit scalability, enabling effortless system expansion to accommodate more users and devices. However, the beacon-dependent approach of Bellrock could potentially limit its scalability and cost-effectiveness in larger deployments. In addition, the suggested protocol supports a wide range of devices and platforms, making it highly versatile and adaptable to different user needs. In contrast, Bellrock only supports Android smartphones, which may limit its applicability in certain contexts. FedLoc, on the other hand, also supports some IoT devices, but not as many as our approach. The ability to support numerous devices and platforms is a crucial characteristic for LBSs, as it allows for greater user adoption and flexibility. Therefore, our protocol's wider support for devices and platforms gives it an advantage over both Bellrock and FedLoc in terms of usability and accessibility.

Table 4
Comparison between our proposed protocol and two existing solutions, Bellrock and FedLoc.

Characteristics	Our protocol	Bellrock [28]	FedLoc [29]
Ensures data authenticity	Yes	No	No
Provides anonymity protection	Yes	Yes	No
Preserves privacy	Yes	Yes	Yes
Supports revocation	Yes	No	No
Adopts serverless authentication architecture	Yes	No	No
Achieves beacon-independent localization	Yes	No	Yes
Supports wearables and IoT devices	Yes	Partial	Yes
Ensures scalability	Yes	No	Yes

Revocation is a crucial aspect of any secure localization protocol, as it enables the system to remove invalid or malicious users that may compromise the integrity of the data. This is a critical feature for ensuring the long-term security and reliability of the system and provides an added layer of protection against potential attacks. Both Bellrock and FedLoc do not consider user revocation, leaving the system vulnerable to potential threats from compromised or malicious users. Our proposed scheme is the only one among the three that takes user revocation into account, allowing for the removal of any users who may pose a threat to the system's security.

Overall, the cryptographic scheme we put forth stands out as a comprehensive and advanced solution for LBSs, offering a range of robust security measures, a scalable architecture, broad device and platform support, and user revocation capabilities. These characteristics make our protocol an ideal choice for various applications where security and reliability are paramount.

9. Conclusion

In this paper, we have presented a novel approach to addressing the privacy and security concerns in CIPSs. By examining Attribute-based Authentication as a solution, we designed and implemented a decentralized scheme that utilizes encrypted and anonymized location information transmitted via BLE advertising. The authentication scheme provides robust privacy protection in a fully decentralized environment, with no reliance on centralized data sources. Our results show efficient performance on a variety of devices, including single-board computers, smartphones, smartwatches, and microcontrollers. Our implementation has proven to be suitable for real-time scenarios, with durations well under 350 ms even on the slowest devices. To the best of our knowledge, this is the first fully decentralized ABA scheme running over BLE, offering a promising solution for protecting user privacy in CIPSs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge funding from the European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreements No. 813278 (A-WEAR: A network for dynamic wearable applications with privacy constraints, http://www.a-wear.eu/) and No. 101023072 (ORIENTATE: Low-cost Reliable Indoor Positioning in Smart Factories, http://orientate.dsi.uminho.pt/). This paper is an extended version of work [49] by Casanova-Marqués, R., Pascacio, P., Hajny, J., Torres-Sospedra, J., 2021. Anonymous Attribute-based Credentials in Collaborative Indoor Positioning Systems, in: Proceedings of the 18th International Conference on Security and Cryptography, SCITEPRESS - Science and Technology Publications, Online Streaming, pp. 791–797. https://doi.org/10.5220/0010582507910797

Appendix. Formal security and privacy analysis

We demonstrate the security of our decentralized Attribute-based Authentication protocol by providing a detailed security and privacy analysis of the `Show` and `Verify` algorithms. The authentication mechanism relies on a combination of provably secure cryptographic primitives, including the widely-used DH key exchange [34], the efficient and secure wBB signature scheme [35], and the robust Sigma protocols [36]. The formal analysis of the security, privacy, and functionality properties comprises six propositions, each of which is supported by a corresponding proof to establish the fulfillment of said properties.

Proposition 1. *The proposed decentralized attribute-based authentication protocol is existentially* ***unforgeable*** *under chosen-message attacks in the random oracle model assuming that the discrete logarithm problem is hard.*

Proof. This is based on the fact that our proposal is built on the wBB signature and its unforgeability is proven in [35]. □

Proposition 2. *The proposed decentralized attribute-based authentication protocol is **complete**, **correct** and **sound**. I.e., valid authentications will always be verified correctly, and invalid ones will always fail verification.*

Proof. The *completeness* property is satisfied when the verifier correctly reconstructs the commitment t'_κ.

$$t'_\kappa = g_1^{s_\kappa} \hat{\sigma}_{x_0}^{-e} \hat{\sigma}_{x_r}^{-em_r} \hat{\sigma}_{x_{ID}}^{s_{ID}} \mathcal{R}^{i_r} = g_1^{\rho_\kappa + e\rho} \sigma_{x_0}^{-e\rho} \sigma_{x_r}^{-e\rho m_r} \sigma_{x_{ID}}^{\rho_{ID}\rho - e\rho m_{ID}} \mathcal{R}^{i_r} = g_1^{\rho_\kappa} \sigma_{x_{ID}}^{\rho_{ID}\rho} \mathcal{R}^{i_r} g_1^{e\rho} \sigma^{-e\rho x_0} \sigma^{-e\rho m_r x_r} \sigma^{-e\rho m_{ID} x_{ID}}$$

$$= g_1^{\rho_\kappa} \sigma_{x_{ID}}^{\rho_{ID}\rho} \mathcal{R}^{i_r} g_1^{e\rho} g_1^{\frac{-e\rho(x_0 + m_r x_r + m_{ID} x_{ID})}{x_0 + m_r x_r + m_{ID} x_{ID}}} = g_1^{\rho_\kappa} \sigma_{x_{ID}}^{\rho_{ID}\rho} \mathcal{R}^{i_r} g_1^{e\rho - e\rho} = g_1^{\rho_\kappa} \sigma_{x_{ID}}^{\rho_{ID}\rho} \mathcal{R}^{i_r} = t_\kappa$$

Three conditions must be met to prove the *correctness* of the protocol:

1. the verifier can reconstruct the decryption key $\mathcal{H}(t'_\kappa)$. This condition is always fulfilled due to the completeness property.
2. the challenge e computed by the verifier is identical to the one the user calculated.
3. the pairing computation determines that $\mathbf{e}(\hat{\sigma}_{x_0}, g_2) = \mathbf{e}(\hat{\sigma}, \mathcal{X}_0)$ in $\mathbb{G}_T$.

$$\mathbf{e}(\hat{\sigma}_{x_0}, g_2) = \mathbf{e}(\hat{\sigma}, \mathcal{X}_0) \rightarrow \mathbf{e}(\sigma_{x_0}^{\rho}, g_2) = \mathbf{e}(\sigma^{\rho}, \mathcal{X}_0) \rightarrow \mathbf{e}(\sigma^{\rho x_0}, g_2) = \mathbf{e}(\sigma^{\rho}, \mathcal{X}_0) \rightarrow$$

$$\mathbf{e}(g_1^{\frac{\rho x_0}{x_0 + m_r x_r + m_{ID} x_{ID}}}, g_2) = \mathbf{e}(g_1^{\frac{\rho}{x_0 + m_r x_r + m_{ID} x_{ID}}}, g_2^{x_0}) \rightarrow \mathbf{e}(g_1, g_2)^{\frac{\rho x_0}{x_0 + m_r x_r + m_{ID} x_{ID}}} = \mathbf{e}(g_1, g_2)^{\frac{\rho x_0}{x_0 + m_r x_r + m_{ID} x_{ID}}}$$

The proposed authentication technique is built on the foundation of zero-knowledge proofs, and its *soundness* is demonstrated through the construction of a knowledge extractor. This extractor employs the well-established rewinding technique described in [37]. □

Proposition 3. *The proposed decentralized attribute-based authentication protocol is **zero-knowledge**. I.e., there exists a simulator $\mathbb{S}$ that can efficiently generate a protocol transcript that is indistinguishable from a real protocol transcript.*

Proof. The *zero-knowledge* property is demonstrated by creating the zero-knowledge simulator $\mathbb{S}$. Assume that the simulator can program the random oracle $\mathcal{H}$ such that when it receives $\tilde{t_\kappa}$, $\tilde{\sigma}$, $\tilde{\sigma}_{x_0}$, $\tilde{\sigma}_{x_r}$, $\tilde{\sigma}_{x_{ID}}$, $timestamp$, and λ as inputs, it outputs $\tilde{e}$. The simulator then executes the following:

1. randomly selects the transaction identifier $\tilde{\mathcal{R}} \in_R \mathbb{G}_1$,
2. randomly selects the shared secret key $\tilde{i_r} \in_R \mathbb{Z}_q$,
3. randomly selects the credentials $\tilde{\sigma}, \tilde{\sigma}_{x_0}, \tilde{\sigma}_{x_r}, \tilde{\sigma}_{x_{ID}} \in_R \mathbb{G}_1$,
4. randomly selects the responses $\tilde{s_\kappa}, s_{\tilde{ID}} \in_R \mathbb{Z}_q$,
5. randomly selects the challenge $\tilde{e} \in_R \mathbb{Z}_q$,
6. computes the commitment $\tilde{t_\kappa} = g_1^{\tilde{s_\kappa}} \tilde{\sigma}_{x_0}^{-\tilde{e}} \tilde{\sigma}_{x_r}^{-\tilde{e} m_r} \tilde{\sigma}_{x_{ID}}^{s_{\tilde{ID}}} \tilde{\mathcal{R}}^{\tilde{i_r}}$.

All pairs are randomly and uniformly selected from the same sets; therefore, the output of the simulator is computationally indistinguishable from the real protocol transcript, i.e., $\tilde{\mathcal{R}} \cong \mathcal{R}$, $(\tilde{\sigma}, \tilde{\sigma}_{x_0}, \tilde{\sigma}_{x_r}, \tilde{\sigma}_{x_{ID}}) \cong (\hat{\sigma}, \hat{\sigma}_{x_0}, \hat{\sigma}_{x_r}, \hat{\sigma}_{x_{ID}})$, and $(\tilde{e}, \tilde{s_\kappa}, s_{\tilde{ID}}) \cong (e, s_\kappa, s_{ID})$. □

Proposition 4. *The proposed decentralized attribute-based authentication protocol is **key-parameter consistent**. I.e., every public key has a distinct private key.*

Proof. The `KeyGen` algorithm outputs $sk_I \leftarrow (x_0, x_r, x_{ID}) \in_R \mathbb{Z}_q$, $pk_I \leftarrow (\mathcal{X}_0 = g_2^{x_0})$, $i_r \in_R \mathbb{Z}_q$, and $\mathcal{I}_r = g_1^{i_r}$. Since we are working in a prime-order group, each element has a unique discrete logarithm. Every public key corresponds to a unique secret key. □

Proposition 5. *The proposed decentralized attribute-based authentication protocol provides **anonymity**, **unlinkability**, and **untraceability**.*

Proof. Due to the proof of knowledge protocol's *zero-knowledge* property, the proof π is always *anonymous*, *unlinkable*, and *untraceable*. The distribution of $\hat{\sigma}$, $\hat{\sigma}_{x_0}$, $\hat{\sigma}_{x_r}$, and $\hat{\sigma}_{x_{ID}}$ is uniform and random in $\mathbb{Z}_q$ since ρ is selected uniformly and randomly from $\mathbb{Z}_q$. Consequently, the disclosed values are indistinguishable from random elements. □

Proposition 6. *The proposed decentralized attribute-based authentication protocol provides **confidentiality** and **integrity**. I.e., the information to be transmitted is not disclosed to unauthorized parties and is tamper-resistant.*

Proof. The *confidentiality* property is achieved by encrypting the data to be transmitted λ with the key $\mathcal{H}(t_\kappa)$. Using $\mathcal{H}(t'_\kappa)$, which is calculated utilizing the random and shareable user values ψ, verifiers can decrypt the information. Additionally, $\mathcal{R}$ and i_r are required for the computation. The value i_r is a secret key shared exclusively among system users. Therefore, an attacker will be unable to read the information since it does not possess the shared secret key. Note that in unrestricted environments, the information is not encrypted and the $\mathcal{R}^{i_r}$ computation is not conducted; hence, there is no *confidentiality*. However, the information remains immutable. The *integrity* property is accomplished by calculating the hash e. If λ or ξ are altered, $e \overset{?}{=} \mathcal{H}(\dots, \lambda)$ will yield a different value, and the verification will fail. □

References

[1] Yu Xianjia, Li Qingqing, Jorge Pena Queralta, Jukka Heikkonen, Tomi Westerlund, Applications of UWB networks and positioning to autonomous robots and industrial systems, in: 2021 10th Mediterranean Conference on Embedded Computing (MECO), IEEE, Budva, Montenegro, 2021, pp. 1–6, http://dx.doi.org/10.1109/MECO52532.2021.9460266.

[2] Ivo Silva, Cristiano Pendao, Joaquin Torres-Sospedra, Adriano Moreira, TrackInFactory: A tight coupling particle filter for industrial vehicle tracking in indoor environments, IEEE Trans. Syst. Man Cybern.: Syst. 52 (7) (2022) 4151–4162, http://dx.doi.org/10.1109/TSMC.2021.3091987.

[3] Lingli Zhao, Xiaoqin Yu, Design and development of anti-theft tracking app based on geofence, in: 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), IEEE, Dalian, China, 2021, pp. 738–741, http://dx.doi.org/10.1109/IPEC51340.2021.9421332.

[4] Tanweer Alam, Abdirahman Ahmed Hadi, Rayyan Qari Shahabuddin Najam, Designing and implementing the people tracking system in the crowded environment using mobile application for smart cities, Int. J. Syst. Assur. Eng. Manage. 13 (1) (2022) 11–33, http://dx.doi.org/10.1007/s13198-021-01277-7.

[5] Pavel Pascacio, Sven Casteleyn, Joaquín Torres-Sospedra, Elena Simona Lohan, Jari Nurmi, Collaborative indoor positioning systems: A systematic review, Sensors 21 (3) (2021) 1002, http://dx.doi.org/10.3390/s21031002.

[6] Rainer Mautz, Indoor positioning technologies, in: Geodätisch-Geophysikalische Arbeiten in der Schweiz, ETH Zurich, 2012, pp. 1–129, http://dx.doi.org/10.3929/ETHZ-A-007313554.

[7] Jian Yang, Christian Poellabauer, Pramita Mitra, Cynthia Neubecker, Beyond beaconing: Emerging applications and challenges of BLE, Ad Hoc Netw. 97 (2020) 102015, http://dx.doi.org/10.1016/j.adhoc.2019.102015.

[8] Vivian Genaro Motti, Kelly Caine, Users' privacy concerns about wearables, in: Financial Cryptography and Data Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 231–244, http://dx.doi.org/10.1007/978-3-662-48051-9_17.

[9] Sandra Seubert, Carlos Becker, The democratic impact of strengthening European fundamental rights in the digital age: The example of privacy protection, Ger. Law J. 22 (1) (2021) 31–44, http://dx.doi.org/10.1017/glj.2020.101.

[10] Roger A. Grimes, Hacking Multifactor Authentication, first ed., Wiley, ISBN: 978-1-119-67235-7, 2020, http://dx.doi.org/10.1002/9781119672357.

[11] Mouna S. Chebli, Heba Mohammad, Khalifa Al Amer, An overview of wireless indoor positioning systems: Techniques, security, and countermeasures, in: Internet and Distributed Computing Systems, Springer International Publishing, Cham, 2019, pp. 223–233, http://dx.doi.org/10.1007/978-3-030-34914-1_22.

[12] Francesco Malandrino, Carlo Borgiattino, Claudio Casetti, Carla-Fabiana Chiasserini, Marco Fiore, Roberto Sadao, Verification and inference of positions in vehicular networks through anonymous beaconing, IEEE Trans. Mob. Comput. 13 (10) (2014) 2415–2428, http://dx.doi.org/10.1109/TMC.2013.2297925.

[13] Zhiquan Liu, Jianfeng Ma, Jian Weng, Feiran Huang, Yongdong Wu, Linfeng Wei, Yuxian Li, LPPTE: A lightweight privacy-preserving trust evaluation scheme for facilitating distributed data fusion in cooperative vehicular safety applications, Inf. Fusion 73 (2021) 144–156, http://dx.doi.org/10.1016/j.inffus.2021.03.003.

[14] Jiaqi Huang, Yi Qian, Rose Qingyang Hu, A privacy-preserving scheme for location-based services in the internet of vehicles, J. Commun. Inf. Netw. 6 (4) (2021) 385–395, http://dx.doi.org/10.23919/JCIN.2021.9663103.

[15] Ning Xi, Weihui Li, Lv Jing, Jianfeng Ma, ZAMA: A ZKP-based anonymous mutual authentication scheme for the IoV, IEEE Internet Things J. 9 (22) (2022) 22903–22913, http://dx.doi.org/10.1109/JIOT.2022.3186921.

[16] Tao Peng, Qin Liu, Dacheng Meng, Guojun Wang, Collaborative trajectory privacy preserving scheme in location-based services, Inf. Sci. 387 (2017) 165–179, http://dx.doi.org/10.1016/j.ins.2016.08.010.

[17] Kimmo Jarvinen, Helena Leppakoski, Elena-Simona Lohan, Philipp Richter, Thomas Schneider, Oleksandr Tkachenko, Zheng Yang, PILOT: Practical privacy-preserving indoor localization using outsourcing, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, Stockholm, Sweden, 2019, pp. 448–463, http://dx.doi.org/10.1109/EuroSP.2019.00040.

[18] Ajay K. Gupta, Udai Shanker, OMCPR: Optimal mobility aware cache data pre-fetching and replacement policy using spatial K-anonymity for LBS, Wirel. Pers. Commun. 114 (2) (2020) 949–973, http://dx.doi.org/10.1007/s11277-020-07402-2.

[19] Viktoriia Shubina, Aleksandr Ometov, Sergey Andreev, Dragos Niculescu, Elena Simona Lohan, Privacy versus location accuracy in opportunistic wearable networks, in: 2020 International Conference on Localization and GNSS (ICL-GNSS), IEEE, Tampere, Finland, 2020, pp. 1–6, http://dx.doi.org/10.1109/ICL-GNSS49876.2020.9115424.

[20] Jong Wook Kim, Kennedy Edemacu, Jong Seon Kim, Yon Dohn Chung, Beakcheol Jang, A survey of differential privacy-based techniques and their applicability to location-based services, Comput. Secur. 111 (2021) 102464, http://dx.doi.org/10.1016/j.cose.2021.102464.

[21] Paolo Barsocchi, Antonello Calabrò, Antonino Crivello, Said Daoudagh, Francesco Furfari, Michele Girolami, Eda Marchetti, COVID-19 and privacy: Enhancing indoor localization architectures towards effective social distancing, Array 9 (2021) 100051, http://dx.doi.org/10.1016/j.array.2020.100051.

[22] Bin Jiang, Jianqiang Li, Guanghui Yue, Houbing Song, Differential privacy for industrial internet of things: Opportunities, applications, and challenges, IEEE Internet Things J. 8 (13) (2021) 10430–10451, http://dx.doi.org/10.1109/JIOT.2021.3057419.

[23] Viktoriia Shubina, Aleksandr Ometov, Anahid Basiri, Elena Simona Lohan, Effectiveness modelling of digital contact-tracing solutions for tackling the COVID-19 pandemic, J. Navig. 74 (4) (2021) 853–886, http://dx.doi.org/10.1017/S0373463321000175.

[24] Huijie Yang, Pandi Vijayakumar, Jian Shen, Brij B. Gupta, A location-based privacy-preserving oblivious sharing scheme for indoor navigation, Future Gener. Comput. Syst. 137 (2022) 42–52, http://dx.doi.org/10.1016/j.future.2022.06.016.

[25] Bohan Li, Ruochen Liang, Wei Zhou, Hailian Yin, Han Gao, Ken Cai, LBS meets blockchain: An efficient method with security preserving trust in SAGIN, IEEE Internet Things J. 9 (8) (2022) 5932–5942, http://dx.doi.org/10.1109/JIOT.2021.3064357.

[26] Zhihua Hu, Yunzhi Li, Guosong Jiang, Rui Zhang, Mande Xie, PriHorus: Privacy-preserving RSS-based indoor positioning, in: ICC 2022 - IEEE International Conference on Communications, IEEE, Seoul, Korea, Republic of, 2022, pp. 5627–5632, http://dx.doi.org/10.1109/ICC45855.2022.9839103.

[27] Jingtao Guo, Ivan Wang-Hei Ho, Yun Hou, Zijian Li, FedPos: A federated transfer learning framework for CSI-based Wi-Fi indoor positioning, IEEE Syst. J. (2023) 1–12, http://dx.doi.org/10.1109/JSYST.2022.3230425.

[28] Augustin Zidek, Shyam Tailor, Robert Harle, Bellrock: Anonymous proximity beacons from personal devices, in: 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, Athens, 2018, pp. 1–10, http://dx.doi.org/10.1109/PERCOM.2018.8444603.

[29] Feng Yin, Zhidi Lin, Qinglei Kong, Yue Xu, Deshi Li, Sergios Theodoridis, Shuguang Robert Cui, FedLoc: Federated learning framework for data-driven cooperative localization and location data processing, IEEE Open J. Signal Process. 1 (2020) 187–215, http://dx.doi.org/10.1109/OJSP.2020.3036276.

[30] Lidia Pocero Fraile, Christos Koulamas, Design and evaluation of an indoor positioning system based on mobile devices, in: 2022 11th Mediterranean Conference on Embedded Computing (MECO), IEEE, Budva, Montenegro, 2022, pp. 1–6, http://dx.doi.org/10.1109/MECO55406.2022.9797091.

[31] Carmen Delgado, Lanfranco Zanzi, Xi Li, Xavier Costa-Perez, OROS: Orchestrating ROS-driven collaborative connected robots in mission-critical operations, in: 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, Belfast, United Kingdom, 2022, pp. 147–156, http://dx.doi.org/10.1109/WoWMoM54355.2022.00026.

[32] Pavel Pascacio, Joaquin Torres-Sospedra, Sven Casteleyn, Elena Simona Lohan, A collaborative approach using neural networks for BLE-RSS lateration-based indoor positioning, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, Padua, Italy, 2022, pp. 01–09, http://dx.doi.org/10.1109/IJCNN55064.2022.9892484.

[33] Mun On Wong, Sanghoon Lee, Indoor navigation and information sharing for collaborative fire emergency response with BIM and multi-user networking, Autom. Constr. 148 (2023) 104781, http://dx.doi.org/10.1016/j.autcon.2023.104781.

[34] Whitfield Diffie, Martin Hellman, New directions in cryptography, IEEE Trans. Inf. Theory 22 (6) (1976) 644–654, http://dx.doi.org/10.1109/TIT.1976.1055638.

[35] Dan Boneh, Xavier Boyen, Short signatures without random oracles and the SDH assumption in bilinear groups, J. Cryptol. 21 (2) (2008) 149–177, http://dx.doi.org/10.1007/s00145-007-9005-7.

[36] Ronald John Fitzgerald Cramer, Modular Design of Secure Yet Practical Cryptographic Protocols (Ph.D. thesis), Universiteit van Amsterdam, ISBN: 978-90-74795-64-7, 1996.

[37] Jan Camenisch, Markus Stadler, Proof Systems for General Statements About Discrete Logarithms, Technical Report, ETH Zurich, 1997, pp. 1–13, http://dx.doi.org/10.3929/ETHZ-A-006651937.

[38] Ronald Cramer, Ivan Damgård, Philip MacKenzie, Efficient zero-knowledge proofs of knowledge without intractability assumptions, in: Public Key Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 354–372, http://dx.doi.org/10.1007/978-3-540-46588-1_24.

[39] Amos Fiat, Adi Shamir, How to prove yourself: Practical solutions to identification and signature problems, in: Advances in Cryptology – CRYPTO' 86, Springer Berlin Heidelberg, Berlin, Heidelberg, 1987, pp. 186–194, http://dx.doi.org/10.1007/3-540-47721-7_12.

[40] Mitsunari Shigeo, MCL library, 2023, https://github.com/herumi/mcl.

[41] Diego de Freitas Aranha, Conrado Porto Lopes Gouvêa, Tobias Markmann, Riad S. Wahby, K. Liao, RELIC is an efficient library for cryptography, 2023, https://github.com/relic-toolkit/relic.

[42] Yann Collet, Lossless compression algorithm, 2023, https://lz4.github.io/lz4/.

[43] RIOT Community, RIOT OS: The friendly operating system for the internet of things, 2023, https://riot-os.org/.

[44] Emmanuel Baccelli, Cenk Gundogan, Oliver Hahm, Peter Kietzmann, Martine S. Lenders, Hauke Petersen, Kaspar Schleiser, Thomas C. Schmidt, Matthias Wahlisch, RIOT: An open source operating system for low-end embedded devices in the IoT, IEEE Internet Things J. 5 (6) (2018) 4428–4440, http://dx.doi.org/10.1109/JIOT.2018.2815038.

[45] Greg Roelofs, A massively spiffy yet delicately unobtrusive compression library, 2022, https://www.zlib.net.

[46] Tukaani Developers, Tukaani project, 2023, https://tukaani.org/xz/.

[47] Bluetooth Special Interest Group, Bluetooth Core Specification V4.2, Bluetooth Special Interest Group, 2014, URL https://www.bluetooth.com/specifications/specs/core-specification-4-2/.

[48] Bluetooth Special Interest Group, Bluetooth Core Specification V5.0, Bluetooth Special Interest Group, 2016, URL https://www.bluetooth.com/specifications/specs/core-specification-5-0/.

[49] Raúl Casanova-Marqués, Pavel Pascacio, Jan Hajny, Joaquín Torres-Sospedra, Anonymous attribute-based credentials in collaborative indoor positioning systems, in: Proceedings of the 18th International Conference on Security and Cryptography, SCITEPRESS - Science and Technology Publications, 2021, pp. 791–797, http://dx.doi.org/10.5220/0010582507910797, Online Streaming.