Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 420 – 435

doi:10.3906/elk-2105-197

Estonian Internet voting with anonymous credentials

Isa SERTKAYA[1*], **Peter ROENNE**[2], **Peter Y. A. RYAN**[2]
[1]MCS Labs & BCLabs, TÜBİTAK BİLGEM UEKAE,Gebze, Turkey
[2]SnT & University of Luxembourg, Luxembourg

Received: 20.05.2021 • **Accepted/Published Online:** 25.12.2021 • **Final Version:** 04.02.2022

Abstract: The Estonian Internet voting (EIV) scheme is a unique example of a long-term nation-wide, legally binding electronic voting deployment. The EIV scheme is used in parallel with standard paper-based election day voting, of course invalidating an already cast i-vote. This necessarily requires careful authentication of the eligible voters and makes the Estonian identity card solution a crucial part of the scheme, however, note that Parsovs has recently drawn attention to the security flaws found in Estonian ID-cards. In this study, we propose an e-voting scheme EIV-AC that integrates the EIV scheme with anonymous credentials based on self-sovereign identity. In addition to the EIV scheme's security properties, the EIV-AC scheme further supports *participation privacy*, i.e. whether or not an eligible voter has participated in an election is kept hidden – also from the election authorities.

Key words: Electronic voting, e-voting, Estonian Internet voting, privacy, anonymous credentials, self-sovereign identity, distributed ledger technologies

1. Introduction

Bernhard et al. summarize the important requirements of secure elections, list the open questions for electronic voting (e-voting) research and then survey the current standings of previously proposed e-voting systems [1]. A fundamental question included in this list is "*Is there a sufficiently secure way to distribute credentials for Internet voting?*". Voter authentication is the most crucial phase for any election system since it is the keystone for assuring

- *eligibility verifiability* that is anyone should be able to check that each vote in the election outcome is cast by an eligible voter and at most one vote per voter is included in the tallying [2],
- *participation privacy*, i.e. the fact of whether or not an eligible voter has participated in an election should not be disclosed without the voter's intention [3],

and hence for resisting especially to

- *ballot stuffing attack*, that tries to submit votes for voters who have not participated in the election or for maliciously forged voter credentials,
- *abstention attack* in which an attacker, who has the list of the participants, tries to refrain voters from voting [4].

*Correspondence: isa.sertkaya@tubitak.gov.tr

Voter authentication is typically handled either with internal (e.g., JCJ-Civitas [4], Australian iVote [5]) or external (e.g., Estonian Internet Voting (EIV) [6]) authentication. Internal voter authentication-based e-voting systems require a registration phase in which voters are first required to prove their identity to the election registrars, register themselves and then get registration credentials in form of public-private key pairs or private credentials to prove voter eligibility during the voting phase. Naturally, this registration phase increases the need for trust in the elections' central authorities and may cause ballot stuffing or abstention attacks if the necessary precautions are not carefully taken or the registrars maliciously collude. Our motivation in this work is to propose a different approach for securely distributing credentials for e-voting schemes by utilizing anonymous credential-based self-sovereign identity as an external authentication. We are going to give an instantiation based on the Estonian Internet voting (EIV) [6].

The Estonian Internet voting (EIV) scheme has been used in national-wide elections since 2005. EIV is organized by the Estonian State Electoral Office (EO) along with the Information System Authority within a one-week period before the actual election day. Voters can cast multiple votes by using EIV and/or cast votes on the election day. In EIV, the last cast vote is tallied, but will be overruled by a cast paper vote. Since its first use in 2005, the usage rate of EIV has increased over time. For instance, in the 2019 European Parliament elections approximately 46.7% of all the votes were cast via the EIV scheme[1].

EIV scheme has been a subject of e-voting research and discussions since its first deployment. During the 2011 elections, a proof of concept vote manipulation attack was mounted [7]. As a counter-measure, optional individual verification is integrated into EIV scheme [8]. In 2014, [9] analyzed the security of EIV scheme based on various aspects and found that the scheme has serious architectural limitations and procedural gaps. Consequently, the EIV IVXV framework was deployed in 2017 [6]. In 2019, the Minister of Foreign Trade and Information Technology called together a committee that produced a list of open action items to potentially work on. Based on this list, [10] draws attention to the electronic identity and summarizes possible improvements.

The Estonian Internet voting scheme relies heavily on the electronic identity (eID) infrastructure. Currently, three main eID solutions are in use and only two of them (ID-card and mID) are used for EIV.

- ID-card: is a compulsory identity document issued by the Police and Border Guard Board to all Estonian citizens and the citizens of the European Union permanently residing in Estonia.
- Mobile-ID (mID): relies on the mobile phone SIM card as the key storage and cryptographic processor. It can be activated by signing an mID service agreement with mobile operator or can be activated on the Police and Border Guard Board website.
- Smart-ID (sID): is a mobile application that works as an identification solution for anyone that does not have a SIM card in their smart device but needs to securely prove their online identity.

Among these identity solutions, ID-card is the most widely used; 67% of the Estonian residents use the ID-card regularly[2]. Recently, [11] has shown several security flaws in the ID-card manufacturing process. These flaws include certificates with duplicate RSA public keys, private key generation outside the ID-card and certificates with corrupted RSA public keys. As a result, [11] suggests looking for fault-tolerant identity designs.

EIV scheme does not satisfy participation privacy against the central authorities, since the signing key pair, that is included in the ID-card, is used for vote signing while forming the ballot. An honest-but-curious

[1] See Statistics about Internet voting in Estonia for details.
[2] https://e-estonia.com/solutions/e-identity/id-card

or a malicious adversary who gains access to the signed-encrypted ballots, can deduce the total list of eligible voters who participated in the election. Since voting in Estonian elections is voluntary[3], instead of actually identifying the participants, building an e-voting scheme on both eligibility verifiability and participation privacy requirements would comply more with the privacy protection and data minimization acts, such as EU General Data Protection Regulation.

Based on the facts given above, in this manuscript, we focus on updating the EIV scheme by integrating it with an anonymous credentials-based self-sovereign identity solution. In contrast to the previous centralized, federated and user-centric identity management solutions, self-sovereign identity (SSI) aims to provide individuals' control over their identity, security, privacy and a single management platform for both real world and digital identities. Most of the SSI solutions use a combination of distributed ledger technologies and anonymous credentials to create immutable identity records. Some examples of privacy-enhanced digital identity frameworks that have been proposed are Identity Mixer, U-Prove, Privacy-ABCs, FIDO, U-port, Sovrin, [12].

A credential is a set of attributes belonging to an entity, such as national identity number, name, birth date/place, residency, signed by an issuing authority. Traditional credential presenting schemes do not support selective disclosure (presenting only a subset of the attributes) or proving ownership of attributes that satisfy some predicates without actually revealing them. For instance, an ID-card can serve as a credential to authenticate oneself. But when it is used for proving an age predicate fulfillment, it would also reveal all the other attributes such as name, surname, etc. Anonymous credentials provide a privacy-preserving tool for proving identity attributes and support various types of zero-knowledge proofs for the attributes.

Related Work. To the best of our knowledge, this study is the first that proposes an integration of EIV scheme with anonymous credential-based self-sovereign identity.

An anonymous credential scheme is in fact a digital signature that enables credential issuance such that one is able to efficiently prove that one owns a digital signature on a particular set of attributes by some zero-knowledge protocols. Anonymous credentials were first introduced by Chaum [13] and later enhanced by Brands [14], Camenisch and Lysyanskaya [15, 16] and Camenisch *et al.* [17]. Currently, Microsoft's U-Prove is based on [14], IBM's IdeMix is based on [15] and Sovrin Foundation's SSI solution is based on [15, 17]. Furthermore, there are several organizations working towards anonymous credentials-based decentralized identity schemes and standardization such as the World Wide Web Consortium Verifiable Credentials Working Group, Decentralized Identity Foundation.

Contributions. In this manuscript, we propose an e-voting scheme EIV-AC that is an integration of the Estonian Internet voting scheme with anonymous credentials-based self-sovereign identity solution. As anonymous credential scheme, the BBS+ signature, given in [17], is chosen. This version of the BBS+ signature is efficient, built on a Type-3 pairings setting, supports selective disclosure proofs and can be further enriched with zero knowledge range and set membership proofs.

The anonymous credential scheme provides privacy-preserving eligibility proof and an election-specific signing key pair for each voter. This election-specific key is constructed by the master secret included in the anonymous credential and the unique election tag published by the election organizer EO, so that all the votes cast by one voter will be linked. As in the current EIV scheme, each vote is encrypted by the public key published by EO. Then, the ballot is formed with this encrypted vote, the voter's signature and corresponding election-specific signature verification key along with zero-knowledge proofs of knowledge for voter's eligibility and correct signing key formation. After that, this ballot is sent to the vote collector for vote storing. Individual

[3]See Constitution: The Constitution of the Republic of Estonia with the latest amendment on 16 May 2015, § 56.

vote verification, the tallying and auditing processes are carried out as in the EIV IVXV framework.

We claim that the EIV-AC scheme satisfies eligibility verifiability and participation privacy, that is the central election authorities and auditors will only be certain that the voter is eligible without actually identifying her. Further, even if the voter casts many votes, only one will be included in the tallying process. Furthermore, we discuss various deployment strategies for EIV-AC scheme.

Organization. In Section 2, we describe our notation and recall the necessary, the underlying cryptographic primitives and anonymous credentials. Our proposed protocol is given in Section 3. In Section 4, we first give details of zero-knowledge proof of knowledge utilized in our protocol and informal security analysis, and then discuss possible implementation framework and deployment strategies. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. Notations

In the sequel, κ is the security parameter and "$\leftarrow$" and "$\leftarrow_{\$}$" denote assigning an output value to a specific variable and to a uniformly distributed variable of a set, respectively. $\{0,1\}^*$ and $\{0,1\}^\kappa$ respectively denote an arbitrary length bit-string and a bit string of length κ.

The notation $[a]P$ corresponds to a scalar multiplication, in an additive group, of a generator $P \in \mathbb{G}$ of order p by a scalar $a \in \mathbb{Z}_p$, i.e. $P + P + \cdots + P$, a times. We will use the following primitives

- Cryptographically secure hash-function $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\kappa$,
- Digital signature scheme $\mathsf{Sig} = (\mathsf{G_{sig}}, \mathsf{K_{sig}}, \mathsf{S_{sig}}, \mathsf{V_{sig}})$ where
  - Setup $(\mathsf{pp_{sig}} \leftarrow \mathsf{G_{sig}}(\kappa))$: Given a security parameter κ, yields public parameters as $\mathsf{pp_{sig}}$,
  - Keygen $((\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{K_{sig}}(\mathsf{pp_{sig}}))$: Given public parameters $\mathsf{pp_{sig}}$, generates a private and public signing key pair,
  - Sign $(\sigma \leftarrow \mathsf{S_{sig}}(\mathsf{sk}, m))$: Given private signing key sk and a message m, signature σ for the message m is generated,
  - Verify $(\{0,1\} \leftarrow \mathsf{V_{sig}}(\mathsf{pk}, m, \sigma))$: Given public key pk, a message m and signature σ, outputs 1 if the signature σ is valid, otherwise outputs 0.
- Public key encryption scheme $\mathsf{Pec} = (\mathsf{G_{pec}}, \mathsf{K_{pec}}, \mathsf{E_{pec}}, \mathsf{D_{pec}})$ where
  - Setup $(\mathsf{pp_{pec}} \leftarrow \mathsf{G_{pec}}(\kappa))$: Given a security parameter κ, outputs public parameters as $\mathsf{pp_{pec}}$,
  - Keygen $((\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{K_{pec}}(\mathsf{pp_{pec}}))$: Given public parameters $\mathsf{pp_{pec}}$, generates a private and public encryption key pair,
  - $\mathsf{Encrypt}$ $(c \leftarrow \mathsf{E_{pec}}(\mathsf{pk}, m))$: Given public key pk and a message m, encrypts the message m and outputs ciphertext c,
  - $\mathsf{Decrypt}$ $(m \leftarrow \mathsf{D_{pec}}(\mathsf{sk}, c))$: Given private key sk and a ciphertext c, decrypts c to m.

- The zero-knowledge proofs of knowledge of discrete logarithms and statements will be given as formalized by [18]. For instance, $PK\{(a_1, a_2, \ldots, a_n) \mid \phi(a_1, a_2, \ldots, a_n)\}$ denotes a prover convinces a verifier of knowledge of values $(a_1, a_2, \ldots, a_n)$ that satisfies a predicate $\phi(a_1, a_2, \ldots, a_n)$.

Definition 1 *[19] Let $\mathbb{G}_1, \mathbb{G}_2$ (additively written) and $\mathbb{G}_T$ (multiplicatively written) be groups of prime order p. A* pairing *e is defined as a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ having the following properties:*

- ***bilinearity**: for all $A \in \mathbb{G}_1, B \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, we have*

$$e([a]A, [b]B) = e(A, B)^{ab},$$

- ***nondegenerecy**: for $A \neq 0_{\mathbb{G}_1}, B \neq 0_{\mathbb{G}_2}$, $e(A, B) \neq 1_{\mathbb{G}_T}$, where $0_{\mathbb{G}_1}$ (resp. $0_{\mathbb{G}_2}$ and $1_{\mathbb{G}_T}$) is the identity element of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$ and $\mathbb{G}_T$).*

Then, a bilinear environment is a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P, Q, e)$ where $r, \mathbb{G}_1, \mathbb{G}_2$, $\mathbb{G}_T$, and e are defined as above, and P (resp. Q) is a generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$).

2.2. Anonymous credentials

As illustrated in Figure 1, a basic anonymous credential-based self-sovereign identity scheme has users U, issuers I and verifiers V as involved entities. Users are holders that receive issued credentials. Issuers grant and verify the credentials of the users. Verifiers verify credentials of the users presented in plain, partially or fully hidden form. For the sake of simplicity, all of the attributes except the first (named as master secret and denoted as

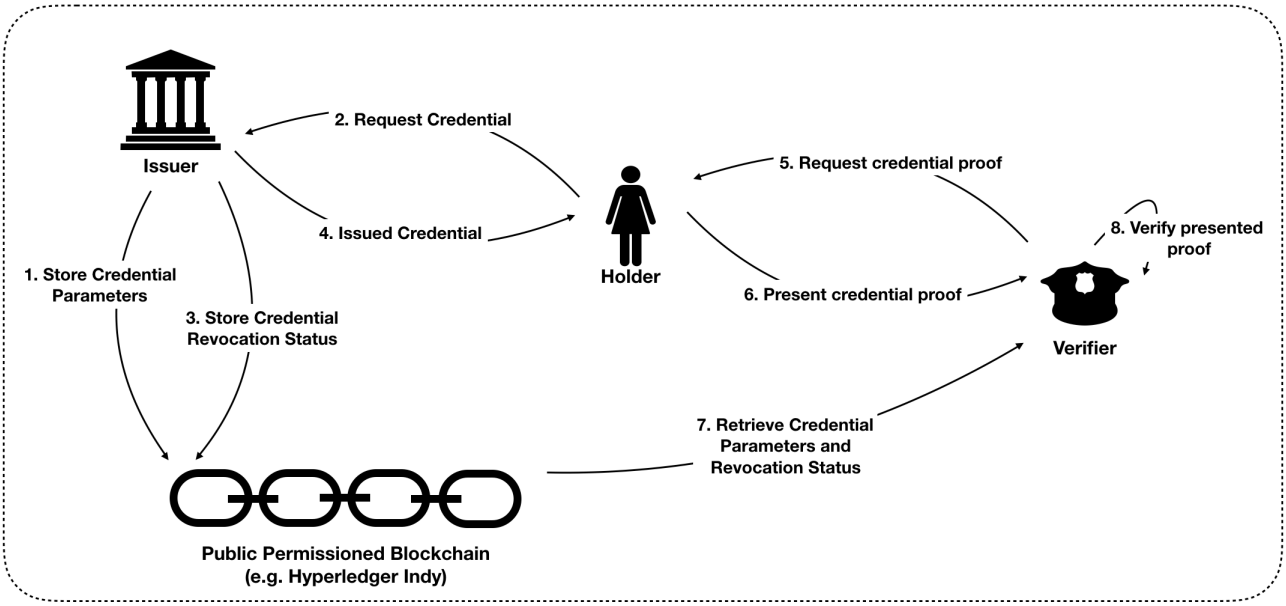

Figure 1. Anonymous credential based self-sovereign identity Data Model

m_1) will be created by the central authority and proof presentation will be described as selective disclosure of attribute predicates. Each of the holder's credentials contains a master secret as a special attribute. This master secret is used to prove that multiple anonymous credentials are linked to one and only one holder. The

issuance of a credential is done based on a commitment of this master secret to ensure that the same master secret is used in all of the user's credentials, without revealing its value to the issuer. Therefore, the master secret is the one secret that the holder holds very dearly, as it is essentially the keystone of her entire identity set.

Distributed ledger technology is used as a verifiable data registry to store public parameters such as credential formations, the underlying cryptographic primitives' public keys and of course if necessary revocation status of the anonymous credentials. An issuer may revoke any credential such that a holder and any verifiers can know within a reasonable amount of time that credential was revoked. For an efficient revocation process, Merkle hash trees or cryptographic accumulators are utilized. Naturally, this scheme can be extended to allow multiple issuers and more advanced proof predicates such as inequality, range, and set membership proofs. We omit such extensions here, but for a more comprehensive protocol reader may refer to [20, 21].

Following [17, 21], anonymous credentials will be given based on the BBS+ signature scheme. We assume that credentials will be parametrized by ℓ attributes and system parameters consisting of a security parameter κ, bilinear environment $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P, Q, e)$ along with the generators $P_0, P_1, \ldots, P_\ell$ of $\mathbb{G}_1$ are constructed.

Any prover, who has a BBS+ signature (A, a, b) on attributes $\{m_i\}_{1 \le i \le \ell}$, can prove knowledge of the signature while selectively disclosing some $\{m_i\}_{i \in \mathcal{D}}$ with $\mathcal{D} \subseteq \{2, \ldots, \ell\}$, by randomizing the signature and presenting the necessary proof. Setup, credential issuance and credential proof presentation with selective disclosure can be outlined as follows.

- Issuer setup: Issuer I chooses $\mathsf{sk_I} \leftarrow_\$ \mathbb{Z}_p^*$ as her private key and $\mathsf{pk_I} \leftarrow [\mathsf{sk_I}]Q$. Optionally, I can also attach a proof of knowledge of her private key.
- Credential issuance:
  1. User U establishes a secure connection with I, proves her real world identity and gets a nonce $n_1 \leftarrow_\$ \{0,1\}^\kappa$, which will be used for proof freshness.
  2. U chooses a master secret $m_1 \leftarrow_\$ \mathbb{Z}_p$, a masking value $b' \leftarrow_\$ \mathbb{Z}_p$, and computes a Pedersen commitment $C_1 \leftarrow [b']P_0 + [m_1]P_1$ along with a nonce n_1 based proof
  $$\pi_1 \leftarrow PK\{(m_1, b') : C_1 = [b']P_0 + [m_1]P_1\},$$
  as described in Section 4.1. Then U sends (C_1, π_1) to I.
  3. I first verifies the proof π_1 as described in Section 4.1, then creates U's attributes as $m_2, m_3, \ldots, m_\ell$, chooses $a, b'' \leftarrow_\$ \mathbb{Z}_p$ and computes
  $$A = [(a + \mathsf{sk_I})^{-1}](P + [b'']P_0 + C_1 + \sum_{i=2}^{\ell} [m_i]P_i).$$
  4. I then sends $\{\{m_i\}_{2 \le i \le \ell}, (A, a, b'')\}$ to U.
  5. U first assures the correctness of the attributes $\{m_i\}_{2 \le i \le \ell}$, next sets $b \leftarrow b' + b''$, computes
  $$B \leftarrow P + [b]P_0 + \sum_{i=1}^{\ell} [m_i]P_i,$$
  and validates the signature by checking $e(A, \mathsf{pk_I} + [a]Q) = e(B, Q)$.

6. If the equations above hold, U stores her issued credential as

$$\alpha_{\mathsf{U}} \leftarrow \{\{m_i\}_{1\leq i\leq \ell}, A, B, a, b\}.$$

- Credential presentation by selective disclosure:

1. U gets the attribute disclosure set $\mathcal{D} \subseteq \{2, 3, \ldots, \ell\}$ and a nonce $n_2 \leftarrow_{\$} \{0,1\}^{\kappa}$ from a verifier V and checks that her credential α_{U} fulfills the requirement.

2. U first randomizes her credential signature and by using n_2, creates a proof

$$\pi_2 \leftarrow PK\{(\{m_i\}_{i\notin\mathcal{D}}, a, r_2, r_3, \bar{b}) : \bar{A} - D = [-a]A' + [r_2]P_0 \wedge$$
$$P + \sum_{i\in\mathcal{D}}[m_i]P_i = [r_3]D + [-\bar{b}]P_0 + \sum_{i\notin\mathcal{D}}[m_i]P_i\}.$$

as given in Section 4.1.

3. U finally sends $(\{m_i\}_{i\in\mathcal{D}}, A', \bar{A}, D, \pi_2)$ to V .

4. V accepts the presented proof only if $A' \neq 0_{\mathbb{G}_1}$ (A' is not the identity element of $\mathbb{G}_1$), $e(A', \mathsf{pk_I}) = e(\bar{A}, Q)$ and the verification given in Section 4.1 for π_2 holds.

3. EIV-AC scheme

Similar to the EIV IVXV framework, the Election Organizer (EO), the Vote Collector (VC), the I-Ballot Box Processor (IBBP) and the Tallier (T) are the core entities. Additionally, the Certification Authority (CA), the Time-marking Service (TMS), the Registration Service (RS), the Data Auditors (DA) and eligible Voters ($\mathcal{V}$) interact with the scheme. These core entities and their interactions are illustrated in Figure 2. As it can be

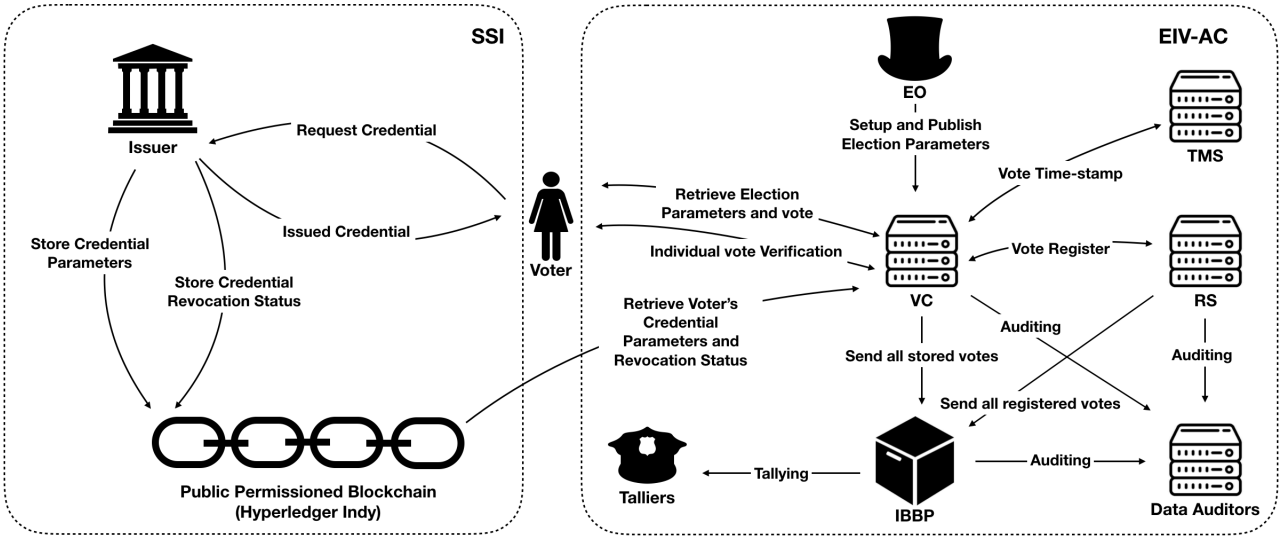

Figure 2. Core entities and their interactions within the EIV-AC scheme.

deduced from Figure 2, self-sovereign identity is utilized as an external authenticator, its underlying permissioned blockchain is governed by the national identity issuing/management body (in Estonian case, this would naturally be Estonian Police and Border Guard Board). Here, the permissioned blockchain is in fact publicly readable blockchain, therefore EIV-AC scheme's vote collecting entity VC can read the necessary metadata for credential verification.

We assume that each eligible voter holds her national ID as a unique anonymous credential issued by the central authority (such as Police and Border Guard Board) and necessary public parameters are stored on a distributed ledger.

$$\forall \mathsf{v} \in \mathcal{V},\ \alpha_{\mathsf{v}} \leftarrow \{\{m_i\}_{1 \le i \le \ell}, A, B, a, b\}.$$

Pre-Voting Phase. For each election, the election organizer EO follows the steps given below as setup.

- Approves the election configuration, constructs the candidates slate $\mathcal{C}_{\mathsf{elc}}$.
- Creates a unique election identifier $\mathsf{t}_{\mathsf{elc}} \leftarrow_{\$} \mathbb{Z}_p^*$.
- Determines a public key encryption scheme Pec and generates an election key pair that is going to be used for encrypting and decrypting the votes,

  $$(\mathsf{sk}_{\mathsf{elc}}, \mathsf{pk}_{\mathsf{elc}}) \leftarrow \mathsf{K}_{\mathsf{pec}}(\kappa).$$

  Any public key encryption scheme satisfying indistinguisability under chosen plaintext attack property can be used, e.g., threshold ELGamal encryption [22] as in EIV.
- Chooses a digital signature scheme Sig, that is going to be used by voters to sign their votes and generates its public parameters. Any digital signature scheme, that satisfies existentially unforgeability against adaptive chosen message attacks, can be utilized. Here, we abuse the notation and create signing key pairs based on the BBS+ signature's pairing environment.
- Performs the role of T - to protect the election private key and to tabulate the voting result.
- Delegates the handling of the online voting phase to VC and the handling of the postvoting/pretabulation offline phase to IBBP where both can be independent organizations.
- Determines a selective disclosure set $\mathcal{D}_{\mathsf{elc}}$, on which anonymous credential proof presentations will be built on, in a privacy preserving manner.

  As explained in Section 2.2, the m_i's, $1 < i \le \ell$, of the anonymous credential, are the holder's attributes including name, surname, birth date, birthplace, nationality, permanent address, etc. For instance, in the parliament elections, $\mathcal{D}_{\mathsf{elc}}$ would be comprised of nationality and birth date, while in the local government council elections, $\mathcal{D}_{\mathsf{elc}}$ would be comprised of nationality, birth date, and permanent address.
- While securely storing the private key $\mathsf{sk}_{\mathsf{elc}}$, publishes the public parameter $\mathsf{pp}_{\mathsf{elc}}$ as

  $$\mathsf{pp}_{\mathsf{elc}} \leftarrow (\mathsf{t}_{\mathsf{elc}}, \mathcal{C}_{\mathsf{elc}}, \mathsf{pk}_{\mathsf{elc}}, \mathcal{D}_{\mathsf{elc}}).$$

- Provides voting application VoteApp and individual verification application VerApp to the voters.

Voting Phase. An eligible voter $\mathsf{v} \in \mathcal{V}$ uses VoteApp to establish a secure connection with VC, gets the candidate list $\mathcal{C}_{\mathsf{elc}}$ and a nonce $n_3 \leftarrow_{\$} \{0,1\}^{\kappa}$. Then v creates the double envelope for her candidate $\mathsf{c_v} \in \mathcal{C}_{\mathsf{elc}}$ as follows.

C_1. The inner envelope is the encrypted choice $\varepsilon_{\mathsf{v}} \leftarrow \mathcal{E}_{\mathsf{pec}}(\mathsf{pk_{elc}}, (\mathsf{c_v}, \mathsf{r_v}))$ with a random bit string $\mathsf{r_v} \in \{0,1\}^{\kappa}$.

C_2. v creates the election specific signing key $\mathsf{sk_v} \leftarrow (m_1 + \mathsf{t_{elc}})^{-1}$ and the corresponding public key $\mathsf{pk_v} \leftarrow [(m_1 + \mathsf{t_{elc}})^{-1}]Q$, and signs the encrypted ballot as

$$\sigma_{\mathsf{v}} = \mathsf{S_{sig}}(\mathsf{sk_v}, \varepsilon_{\mathsf{v}})$$

C_3. v randomizes her credential signature, creates a proof π_3 for having a credential α_{v} satisfying $\mathcal{D}_{\mathsf{elc}}$ and correctly formed signing public key $\mathsf{pk_v} \leftarrow [(m_1 + \mathsf{t_{elc}})^{-1}]Q$:

$$\begin{aligned}\pi_3 \leftarrow PK\{(\{m_i\}_{i \notin \mathcal{D}_{\mathsf{elc}}}, a, r_2, r_3, \bar{b}) :\\ \bar{A} - D = [-a]A' + [r_2]P_0 \quad \wedge\\ P + \sum_{i \in \mathcal{D}_{\mathsf{elc}}} [m_i]P_i = [r_3]D + [-\bar{b}]P_0 + \sum_{i \notin \mathcal{D}_{\mathsf{elc}}} [m_i]P_i \quad \wedge\\ \mathsf{pk_v} = [(m_1 + \mathsf{t_{elc}})^{-1}]Q\}.\end{aligned}$$

where the index i runs over the indexes of attributes that are required to be revealed by the EO, as given in Section 4.1.

C_4. the outer envelope is formed as $\mathsf{b_v} \leftarrow (\varepsilon_{\mathsf{v}}, \sigma_{\mathsf{v}}, \mathsf{pk_v}, (\{m_i\}_{i \in \mathcal{D}_{\mathsf{elc}}}, A', \bar{A}, D, \pi_3))$ and then it is sent to VC.

C_5. VC responds with a unique identifier $\mathsf{b_{id}}$ and the RS confirmation $\mathsf{reg_{b_{id}}}$.

C_6. VoteApp verifies the digitally signed $\mathsf{reg_{bid}}$ with respect to $\mathsf{H(b_v)}$.

C_7. VoteApp presents the vote identifier $\mathsf{b_{id}}$ and $\mathsf{r_v}$ in a QR code in order to be captured by VerApp.

Vote Storing Phase. In order to store a vote, VC needs to verify and register the vote.

S_1. VC accepts presented credential proof if and only if the credential is not revoked, $A' \neq 0_{\mathbb{G}_1}$, $e(A', \mathsf{pk_I}) = e(\bar{A}, Q)$ and the verification, given in Section 4.1, of π_3 holds.

S_2. Then VC validates the signature σ_{v} included in $\mathsf{b_v}$ by checking

$$1 \leftarrow \mathsf{V_{sig}}(\mathsf{pk_v}, \varepsilon_{\mathsf{v}}, \sigma_{\mathsf{v}}).$$

S_3. VC generates a unique random vote identifier $\mathsf{b_{id}}$.

S_4. VC acquires a time-mark $\mathsf{ts_{b_{id}}} \leftarrow \mathsf{S_{sig}}(\mathsf{sk_{TMS}}, (\mathsf{H(b_v)}, \mathsf{utc_{b_{id}}}))$, from the TMS to show that the data $\mathsf{H(b_v)}$ existed at the time $\mathsf{utc_{b_{id}}}$.

S_5. VC sends a registration request $\mathsf{req_{b_{id}}} \leftarrow \mathsf{S_{sig}}(\mathsf{sk_{VC}}, (\mathsf{b_{id}}, \mathsf{H(b_v)}))$, to RS.

S_6. RS validates the signature $\mathsf{req_{b_{id}}}$ and responds to VC with a signed confirmation $\mathsf{reg_{b_{id}}} \leftarrow \mathsf{S_{sig}}(\mathsf{sk_{RS}}, \mathsf{H(req_{b_{id}})})$.

S_7. VC sends the $\mathsf{b_{id}}$ and the confirmation $\mathsf{reg_{b_{id}}}$ to the voter's VoteApp.

Any voter can cast vote multiple times. In this phase, all votes are stored without removal. Whenever a vote storing process is completed successfully,

- VC stores $(\mathsf{b_v}, \mathsf{b_{id}}, \mathsf{ts_{b_{id}}}, \mathsf{reg_{b_{id}}})$ into the ballot set $\mathcal{B}_{\mathsf{VC}}$,
- RS stores $(\mathsf{req_{b_{id}}}, \mathsf{reg_{b_{id}}})$ into the ballot registry set $\mathcal{B}_{\mathsf{RS}}$.

Individual Vote Verifying Phase. Any voter, who wants to check her vote is recorded as cast, optionally uses VerApp running on a mobile device.

V_1. VerApp captures the vote identifier $\mathsf{b_{id}}$ and randomness $\mathsf{r_v}$ from VoteApp through a QR code.

V_2. VerApp establishes an authenticated secure channel with VC and sends $\mathsf{b_{id}}$ to the VC.

V_3. VC responds to VerApp with $\mathsf{b_v}$ and $\mathsf{reg_{b_{id}}}$ corresponding to the $\mathsf{b_{id}}$. In case of an unknown $\mathsf{b_{id}}$ or exceeded verification time frame, an error is returned.

V_4. VerApp verifies both $\mathsf{b_v}$ and $\mathsf{reg_{b_{id}}}$.

V_5. VerApp uses $\mathcal{C}_{\mathsf{elc}}$ and the randomness $\mathsf{r_v}$ to find a $\mathsf{c'} \in \mathcal{C}_{\mathsf{elc}}$ such that

$$\mathcal{E}_{\mathsf{pec}}(\mathsf{pk_{pec}}, (\mathsf{c'}, \mathsf{r_v})) = \varepsilon_{\mathsf{v}}\,.$$

V_6. The result of this process -either the $\mathsf{c'}$ or an error message- is displayed to the voter who has to decide if the result reflects her will, i.e. $\mathsf{c'} = \mathsf{c_v}$.

Pre-Tabulation Phase. Following the completion of the online voting phase, digitally signed ballot set $\mathcal{B}_{\mathsf{VC}}$ and registration query and response set $\mathcal{B}_{\mathsf{RS}}$ are respectively transferred from VC and RS to IBBP. Then IBBP,

- verifies all double envelopes, checks eligibility and verifies registry confirmations,
- compares $\mathcal{B}_{\mathsf{VC}}$ and $\mathcal{B}_{\mathsf{RS}}$, based on the voters' signature verification keys $\{\mathsf{pk_v}\}_{\mathsf{v}\in\mathcal{V}}$, composes new set $\mathcal{B}_{\mathsf{IBBP}}$ by including only the last cast vote from each voter along with its registry confirmation,
- transmits the $\{\mathsf{pk_v}\}_{\mathsf{v}\in\mathcal{V}}$ set to EO and receives the ballot revocation list (based on the corresponding paper vote existence),
- removes those ballots from $\mathcal{B}_{\mathsf{IBBP}}$ and extracts the encrypted vote set $\mathcal{B}_{\mathsf{elc}} \leftarrow \{\varepsilon_{\mathsf{v}}\}$ that is going to be tabulated.

Tabulation. EO decrypts the nonrevoked votes included in $\mathcal{B}_{\mathsf{elc}}$ with the election's private key $\mathsf{sk_{elc}}$ and publishes the election result along with a proof that assures all the procedures were correctly computed. Naturally, election auditing is carried out as in the IVXV framework.

4. Analysis and discussion

4.1. Zero knowledge proofs of knowledge

Details of π_1. User U uses π_1 for proving the knowledge of the master secret m_1 to the issuer I.

Recall that

$$\pi_1 \leftarrow PK\{(m_1, b') : C_1 = [b']P_0 + [m_1]P_1\}.$$

1. Prover chooses $r_{m_1}, r_{b'} \leftarrow_{\$} \mathbb{Z}_p$ and computes

$$T_1 \leftarrow [r_{b'}]P_0 + [r_{m_1}]P_1.$$

2. Using the nonce value n_1, prover computes

$$c_1 \leftarrow \mathsf{H}(n_1, T_1, C_1), \quad s_{m_1} \leftarrow r_{m_1} - c_1 m_1, \quad s_{b'} \leftarrow r_{b'} - c_1 b'.$$

3. Then, prover sends $(c_1, T_1, s_{m_1}, s_{b'})$ as π_1 to the verifier.

4. Verifier checks $c_1 \stackrel{?}{=} \mathsf{H}(n_1, T_1, C_1)$ with nonce n_1 and

$$[s_{b'}]P_0 + [s_{m_1}]P_1 = T_1 - c_1 C_1,$$

if both verifications hold, the verifier accepts the proof.

Details of π_2. User U uses π_2 for selectively disclosing some of her credentials to the verifier V and proving that she owns a signature of issuer I on them without revealing all of the attributes and the signature itself.

In order to randomize the signature,

- U chooses $r_1 \leftarrow_{\$} \mathbb{Z}_p^*$, sets $A' \leftarrow [r_1]A$ and computes $\bar{A} \leftarrow [r_1]B + [-a]A'$.
- Next, U chooses $r_2 \leftarrow_{\$} \mathbb{Z}_p$, sets $D \leftarrow [r_1]B + [-r_2]P_0$, $r_3 \leftarrow r_1^{-1}$ and $\bar{b} \leftarrow b - r_2 r_3$.

Then,

$$\pi_2 \leftarrow PK\{(\{m_i\}_{i \notin \mathcal{D}}, a, r_2, r_3, \bar{b}) : \bar{A} - D = [-a]A' + [r_2]P_0 \wedge$$
$$P + \sum_{i \in \mathcal{D}} [m_i]P_i = [r_3]D + [-\bar{b}]P_0 + \sum_{i \notin \mathcal{D}} [m_i]P_i\}.$$

is created and verified as follows.

1. Prover chooses

$$r_{m_i} \leftarrow_{\$} \mathbb{Z}_p \text{ for } i \notin \mathcal{D} \text{ and } r_a, r_{r_2}, r_{r_3}, r_{\bar{b}} \leftarrow_{\$} \mathbb{Z}_p,$$

then computes

$$T_1 \leftarrow [r_a]A' + [r_{r_2}]P_0,$$
$$T_2 \leftarrow [r_{r_3}]D + [r_{\bar{b}}]P_0 + \sum_{i \notin \mathcal{D}} [r_{m_i}]P_i.$$

2. Based on the nonce value n_2, prover computes

$$c_2 \leftarrow \mathsf{H}(n_2, A', \bar{A}, D, T_1, T_2)$$
$$s_{m_i} \leftarrow r_{m_i} - c_2 m_i \quad \text{for all } i \notin \mathcal{D}$$
$$s_a \leftarrow r_a - c_2 a$$
$$s_{r_2} \leftarrow r_{r_2} - c_2 r_2$$
$$s_{r_3} \leftarrow r_{r_3} - c_2 r_3$$
$$s_{\bar{b}} \leftarrow r_{\bar{b}} - c_2 \bar{b}$$

3. Finally, prover sends $(c_2, T_1, T_2, \{s_{m_i}\}_{i \notin \mathcal{D}}, s_a, s_{r_2}, s_{r_3}, s_{\bar{b}})$ as π_2.

4. To verify π_2, verifier first checks

$$c_2 \stackrel{?}{=} \mathsf{H}(n_2, A', \bar{A}, D, T_1, T_2),$$

with nonce n_2 and then the following hold.

$$[s_a]A' + s_{r_2}P_0 = T_1 - [c_2](\bar{A} - D)$$
$$[s_{r_3}]D + [s_{\bar{b}}]P_0 + \sum_{i \notin \mathcal{D}} [s_{m_i}]P_i = T_2 - [c_2](P + \sum_{i \in \mathcal{D}} [m_i]P_i)$$

Lemma 1 *[17] The construction of π_2 forms a zero-knowledge proof of knowledge of a BBS+ signature for signatures with $A \neq 1_{\mathbb{G}_1}$.*

Details of π_3. User U uses π_3 for selectively disclosing some of her credentials to the verifier VC, her election signing key is properly formed with the election's tag $\mathsf{t_{elc}}$ and proving that she owns a signature of issuer I on them without revealing all of the attributes and the signature itself.

In a similar way, the signature is randomized as follows.

- v chooses $r_1 \leftarrow_\$ \mathbb{Z}_p^*$, sets $A' \leftarrow [r_1]A$ and computes $\bar{A} \leftarrow [r_1]B + [-a]A'$.
- Next, v chooses $r_2 \leftarrow_\$ \mathbb{Z}_p$, sets $D \leftarrow [r_1]B + [-r_2]P_0$, $r_3 \leftarrow r_1^{-1}$ and $\bar{b} \leftarrow b - r_2 r_3$.

We rewrite π_3 as

$$\begin{aligned}\pi_3 \leftarrow PK\{(\{m_i\}_{i \notin \mathcal{D}_{\mathsf{elc}}}, a, r_2, r_3, \bar{b}) :\\ \bar{A} - D = [-a]A' + [r_2]P_0 \wedge\\ P + \sum_{i \in \mathcal{D}_{\mathsf{elc}}} [m_i]P_i = [r_3]D + [-\bar{b}]P_0 + \sum_{i \notin \mathcal{D}_{\mathsf{elc}}} [m_i]P_i \wedge\\ [m_1]\mathsf{pk_v} = Q - [\mathsf{t_{elc}}]\mathsf{pk_v}\},\end{aligned}$$

and proof creation and verification is pursued as follows.

1. Prover chooses

$$r_{m_i} \leftarrow_\$ \mathbb{Z}_p \text{ for } i \notin \mathcal{D}_{\mathsf{elc}} \text{ and } r_a, r_{r_2}, r_{r_3}, r_{\bar{b}} \leftarrow_\$ \mathbb{Z}_p,$$

then computes

$$T_1 \leftarrow [r_a]A' + [r_{r_2}]P_0,$$
$$T_2 \leftarrow [r_{r_3}]D + [r_{\bar{b}}]P_0 + \sum_{i \notin \mathcal{D}_{\mathsf{elc}}} [r_{m_i}]P_i,$$
$$T_3 \leftarrow [r_{m_1}]\mathsf{pk_v}$$

2. Based on the nonce value n_3, prover computes

$$c_3 \leftarrow \mathsf{H}(n_3, A', \bar{A}, D, T_1, T_2, T_3)$$
$$s_{m_i} \leftarrow r_{m_i} - c_3 m_i \quad \text{for all } i \notin \mathcal{D}_{\mathsf{elc}}$$
$$s_a \leftarrow r_a - c_3 a$$
$$s_{r_2} \leftarrow r_{r_2} - c_3 r_2$$
$$s_{r_3} \leftarrow r_{r_3} - c_3 r_3$$
$$s_{\bar{b}} \leftarrow r_{\bar{b}} - c_3 \bar{b}$$

3. Finally, prover sends $(c_3, T_1, T_2, T_3, \{s_{m_i}\}_{i \notin \mathcal{D}_{\mathsf{elc}}}, s_a, s_{r_2}, s_{r_3}, s_{\bar{b}})$ as π_3.

4. In order to verify π_3, the verifier first checks

$$c_3 \stackrel{?}{=} \mathsf{H}(n_3, A', \bar{A}, D, T_1, T_2, T_3),$$

with nonce n_3 and then the following hold.

$$[s_a]A' + s_{r_2}P_0 = T_1 - [c_3](\bar{A} - D)$$
$$[s_{r_3}]D + [s_{\bar{b}}]P_0 + \sum_{i \notin \mathcal{D}_{\mathsf{elc}}} [s_{m_i}]P_i = T_2 - [c_3](P + \sum_{i \in \mathcal{D}_{\mathsf{elc}}} [m_i]P_i),$$
$$[s_{m_1}]\mathsf{pk_v} = T_3 - [c_3](Q - [\mathsf{t_{elc}}]\mathsf{pk_v}).$$

Lemma 2 *The construction of π_3 forms a zero-knowledge proof of knowledge of a BBS+ signature for signatures with $A \neq 1_{\mathbb{G}_1}$.*

Proof The proof follows easily from Lemma 1 and its proof given in [17]. □

4.2. Security analysis

Naturally, the security analysis of EIV-AC scheme heavily relies on IVXV framework's assumptions and properties. The IVXV framework addresses individual verifiability with the optional individual verification tool, and universal verifiability by distributing the central roles among independent entities, and strictly logging and auditing these entities. Similarly, as in the IVXV framework, the EIV-AC scheme supports casting multiple Internet voting and paper-based voting on the election day to ensure a considerable level of coercion-resistance.

Due to the space constraint, here we are going to focus on eligibility verifiability and participation privacy.

- *Eligibility verifiability* requires that anyone can check that each vote in the election outcome is cast by an eligible voter and at most one vote per voter is included in the tallying [2]. In order to break the eligibility verifiability property of EIV-AC scheme, an adversary has to either forge an anonymous credential with nonrevocation proof or construct more than one signature key pair for an election from an anonymous credential. The digital signature scheme Sig and the proof π_3 assures that the voter's signing key pair is formed as $\mathsf{sk_v} \leftarrow (m_1 + \mathsf{t_{elc}})^{-1}$ and $\mathsf{pk_v} \leftarrow [(m_1 + \mathsf{t_{elc}})^{-1}]Q$, where m_1 is the master secret included in national ID credential α_v. This ensures that for each α_v, there exists a unique signing key pair. Since the ballots are bound with the signature verification key, multiple ballots with the same signature verification key are processed uniquely based on the revoting policy. On the other hand, the former contradicts with the existential unforgeability property, against adaptive chosen message attacks under the q-SDH assumption, of the BBS+ signature scheme [17], and the nonrevocation status of the credential assured by the SSI's distributed ledger. Besides, anyone who has access to $\mathcal{B}_{\mathsf{elc}}$ and $\mathcal{B}_{\mathsf{IBBP}}$ along with the respective nonce values can check that only the eligible voters' vote is included in tallying.

- *Participation privacy* requires that the fact of whether or not an eligible voter has participated in the election should not be disclosed to the passive adversary who only has access to the public output [3]. Here, we assume that the central authorities including EO, VC, RS and IBBP, are honest-but-curious and they would try to identify the participants. Naturally, participation privacy assumes that the voters are not actively trying to prove that they abstained or participated in the election and an anonymous communication channel is already established[4]. In order to break the participation privacy property of the EIV-AC scheme, an adversary should distinguish a ballot $\mathsf{b_v}$ for an eligible voter v. However, this would contradict the pseduo-randomness of key generation that is assured by q-DDHI assumption, see specifically Theorem 3 in [23].

4.3. Discussion

The EIV-AC scheme makes use of external authentication with anonymous credentials. This naturally requires a nation-wide adoption of the self-sovereign identity model. Sovrin Foundation is building a self-sovereign identity solution based on the Hyperledger Indy project. Even if it currently supports RSA-based anonymous credentials, BBS+ signature based anonymous credentials integration is proposed and implemented [21]. On the other hand, using anonymous credentials as external authentication is not mandatory. Indeed, in a similar way to [4], anonymous credentials can be mounted as internal authentication, and hence issued by the election organizer EO as given in Section 2.2. This variation would still support participation privacy due to BBS+ signature randomization and Lemma 2.

In order to deploy the EIV-AC scheme along with paper-based voting, eligible voters must be authenticated via their anonymous credentials and hence their election-specific signature verification key must be obtained to assure vote uniqueness. Voters' election-specific signature verification key is the only link that binds the ballots. As stated earlier, if a voter casts a paper vote in the election, it is tallied and the cast votes via EIV are revoked. In this way, if a privacy preserving selective disclosure set is determined, paper-based voting can also be pursued in accordance with the participation privacy property.

As for efficiency, BBS+ signatures are in the flexible Type-3 pairing setting and more efficient than previous Camenisch-Lysanskaya like signatures [17]. Indeed, component sizes for the generic bilinear environment

[4]Please refer to Wikipedia Anonymity Networks Category for possible anonymity network solutions to resist de-anonymization by the IP addresses.

Table . BBS+ signature element sizes.

	Generic size	Size with BLS12-381
Private key	$1\mathbb{Z}_p$	32 bytes
Public key	$1\mathbb{G}_2$	96 bytes
Signature	$1\mathbb{G}_1 + 2\mathbb{Z}_p$	112 bytes
Proof	$5\mathbb{G}_1 + (4 + \#\{\text{hidden attributes}\})\mathbb{Z}_p$	$368 + 32 \cdot \#\{\text{hidden attributes}\}$

and particular construction over BLS12-381 curve are listed in Table . Several cryptographic libraries such as Apache's Milagro Library already include support for BLS12-381. For specific construction of BBS+ signature with BLS12-381 curve, reader may refer to [21] and Hyperledger Indy and Ursa Github repositories[5].

5. Conclusion

In essence, following the concerns on the current eID solutions, this manuscript proposes an e-voting scheme EIV-AC that integrates anonymous credential-based self-sovereign identity into the EIV scheme. The EIV-AC scheme particularly utilizes BBS+ signatures as an external authentication process for proving voters' eligibility. In addition to the current scheme's security features, this scheme further provides participation privacy, hides the fact of whether or not an eligible voter has cast a vote, so that it concord more with privacy enhancing and data minimization regulations. The main limitation on the adoption of our scheme is the necessity of a new national identity solution deployment that supports anonymous credentials and self-sovereignty. Enrichment of zero-knowledge proofs of knowledge and further minimization of the trust on the central roles remain as future work.

References

[1] Bernhard M, Benaloh J, Halderman JA, Rivest RL, Ryan PY, et al. Public evidence from secret ballots. In: International Joint Conference on Electronic Voting; Springer, 2017. pp. 84-109. doi: 10.1007/978-3-319-68687-5_6

[2] Kremer S, Ryan M, Smyth B. Election verifiability in electronic voting protocols. In: European Symposium on Research in Computer Security; Springer, 2010. pp. 389-404. doi: 10.1007/978-3-642-15497-3_24

[3] Kulyk O, Teague V, Volkamer M. Extending Helios towards private eligibility verifiability. In: International Conference on E-Voting and Identity; Springer, 2015. pp. 57-73. doi: 10.1007/978-3-319-22270-7_4

[4] Juels A, Catalano D, Jakobsson M. Coercion-resistant electronic elections. In: Towards Trustworthy Elections; Springer, 2010. pp. 37-63. doi: 10.1007/978-3-642-12980-3_2

[5] Eldridge M. A trustworthy electronic voting system for Australian federal elections. 2018. arXiv preprint arXiv:1805.02202. 2018 May 6.

[6] Heiberg S, Martens T, Vinkel P, Willemson J. Improving the verifiability of the Estonian Internet Voting scheme. In: International Joint Conference on Electronic Voting; Springer, 2016. pp. 92-107. doi: 10.1007/978-3-319-52240-1_6

[7] Heiberg S, Laud P, Willemson J. The application of i-voting for Estonian parliamentary elections of 2011. In: International Conference on E-Voting and Identity; Springer, 2011. pp. 208-223. doi: 10.1007/978-3-642-32747-6_13

[8] Heiberg S, Willemson J. Verifiable internet voting in Estonia. In: 6th International Conference on Electronic Voting: Verifying the vote (EVOTE); IEEE, 2014. pp. 1-8. doi: 10.1109/EVOTE.2014.7001135

[5] https://github.com/hyperledger/ursa.

[9] Springall D, Finkenauer T, Durumeric Z, Kitcat J, Hursti H, et al. Security analysis of the Estonian internet voting system. In: 2014 ACM SIGSAC Conference on Computer and Communications Security; ACM, 2014. pp. 703-715. doi: 10.1145/2660267.2660315

[10] Heiberg S, Krips K, Willemson J. Planning the next steps for Estonian Internet voting. In: E-Vote-ID 2020; 2020. pp.82.

[11] Parsovs A. Estonian electronic identity card: security flaws in key management. In: 29th USENIX Security Symposium (USENIX Security 20); 2020. pp. 1785-1802.

[12] Khovratovich D, Law J. Sovrin: digital identities in the blockchain era. In: Rebooting Web-of-Trust 3 Workshop. 2016.

[13] Chaum D. Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM. 1985. pp. 1030-1044.

[14] Brands S. Rethinking public key infrastructures and digital certificates: building in privacy. PhD, MIT, USA, 2000,

[15] Camenisch J, Lysyanskaya A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2001); Springer, 2001. pp. 93-118. doi: 10.1007/3-540-44987-6_7

[16] Camenisch J, Lysyanskaya A. Signature schemes and anonymous credentials from bilinear maps. In: Annual International Cryptology Conference (CRYPTO 2004); Springer, 2004. pp. 56-72. doi: 10.1007/978-3-540-28628-8_4

[17] Camenisch J, Drijvers M, Lehmann A. Anonymous attestation using the strong Diffie Hellman assumption revisited. In: International Conference on Trust and Trustworthy Computing; Springer, 2016. pp. 1-20. doi: 10.1007/978-3-319-45572-3_1

[18] Camenisch J, Kiayias A, Yung M. On the portability of generalized Schnorr proofs. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2009); Springer, 2009. pp. 425-442. doi: 10.1007/978-3-642-01001-9_25

[19] El Mrabet N, Joye M, editors. Guide to pairing-based cryptography. CRC Press; 2017.

[20] IBM Research Zurich Security Team. Specification of the Identity Mixer cryptographic library. Technical Report RZ3730, IBM, 2010.

[21] Lodder M, Zundel B, Khovratovich D. Pairings-based Anonymous Credentials with Circuit-based Revocation and Permission Policies, 2019.

[22] Desmedt Y, Frankel Y. Threshold cryptosystems. In: Conference on the Theory and Application of Cryptology (CRYPTO'89); Springer, 1989. pp. 307-315. doi: 10.1007/0-387-34805-0_28

[23] Belenkiy M, Chase M, Kohlweiss M, Lysyanskaya A. Compact e-cash and simulatable VRFs revisited. In: International Conference on Pairing-Based Cryptography; Springer, 2009. pp. 114-131. doi: 10.1007/978-3-642-03298-1_9