Democratic online referendums with opn.vote

Joerg Mitzlaff [1], Felix Maduakor

[1] openPetition gGmbH, Am Friedrichshain 34, 10407 Berlin, Germany

Abstract. A general e-voting scheme and a proof of concept implementation of a protocol for democratic online-referendums is presented. It emphasizes decentralization, trustlessness, software independence, low transaction costs and usability. Voter privacy is guaranteed by a blind signature encrypted voting entitlement. End-to-end verifiability is provided by a public blockchain bulletin board while assuring usability using a Gelato relay proxy wallet for account abstraction.

Keywords: Internet-voting, E-voting, Online-voting, End-to-end verifiable, Voter privacy, Blind signature, Blockchain, Account abstraction

1. Use Case

This voting scheme has been developed for the specific use case of having a self organized nationwide referendum in Germany, which is aimed to become a legally binding process in the future. The referendum is held annually by the non-profit organization ABSTIMMUNG21 with more than 250.000 participants in a postal vote. Reducing the environmental footprint we want to transition at least 50% of the participants to online voting, having vote transaction costs go down from 2 EUR to 0.02 EUR or less and having the user flow as convenient as postal voting or better.
The goal is to become as close as possible to postal vote behavior, while still being fully digital, since this is what will be the benchmark after all when it comes to user acceptance. This includes decentralization of the infrastructure and authorities, eliminating the need of trust in single entities and becoming software independent [2] which means that errors or unexpected changes in the software components do not lead to undetected changes in the election outcomes.
Software components will become open source and usage will be free-of-charge for non-commercial use.
Since online referendums are a specific variation of online voting we will be using the term e-voting accordingly.

2. Voting scheme

Strong voter authentication will be used but is outside the scope of the voting scheme.
Voter registration period can overlap with vote casting period, however, it is recommended that registration and vote cast are made within a time interval and from different devices and internet connections to guarantee voter privacy.

Before absentee application the voter generates a voter secret locally once for all future applications. The secret works as the voting entitlement which is blindly signed by the register, without getting to know the voting entitlement itself. That's why vote casting is completely anonymous and ballots can be published to a public blockchain without compromising privacy. Until the end of the election, ballots will be encrypted by the election public key. Absentee application is also published to the public blockchain for auditing purposes.
To accommodate users without web3 wallets, Gelato's account abstraction technology is utilized. This allows for subsidizing the blockchain transaction fees for voters, enabling them to cast ballots without owning cryptocurrency or managing a web3 wallet themselves.
The ballot is written into the blockchain without manipulation, which can be verified individually by the voter. Once the election private key is published, anyone can individually verify the correctness of their own ballot and anyone can universally verify that all voting entitlements were valid and that the election results were tallied correctly.

3. Software architecture

The overall system architecture of opn.vote is designed to provide an e-voting solution that prioritizes security, transparency, and auditability. opn.vote allows independent third parties to audit and verify each step of the voting process, while still maintaining voter privacy and the integrity of the election.
The key voting processes, including token generation, vote encryption, and blinded signature validation, occur client-side, ensuring that sensitive operations remain secure. opn.vote minimizes the use of traditional databases and centralized backends, relying on the blockchain and IPFS for critical election data to enhance security and transparency.
The opn.vote protocol utilizes RSA Blind Signatures to ensure user privacy throughout the entire voting process. By utilizing RSA Blind Signatures, opn.vote can validate a voter's eligibility without being able to link the validated credentials to the actual cast vote. Additionally, all votes are encrypted using RSA-2048 with a public key stored on the blockchain, ensuring voter secrecy until the election is completed.
opn.vote Protocol consists of several entities that work together to enable private and verifiable voting:
Authorization Provider (AP). The AP is a server operated by the election coordinator. It acts as the gatekeeper for any election on opn.vote, determining who is eligible to vote. The AP authenticates voters using methods specified by the organizer (e.g. electronic identification (eID) systems) and issues for each eligible voter a JSON Web Token (JWT). This JWT functions as the voter's initial proof of eligibility to vote for the specified election.
Register. The Register is a server that validates the JWT issued by the AP and provides a blind signature for voter privacy. It signs a token generated and blinded by the voter. The unblinded signature and unblinded token serve as the voter's official voting credentials. To ensure transparency, the Register publishes each blind signature issuance on the blockchain.
Signature Validation Server (SVS). Currently, the SVS acts as an intermediary between the voter and the blockchain. It validates the official voting credentials and signs the voting transaction, allowing it to be accepted by the blockchain.

Blockchain/IPFS. The blockchain and IPFS serve as decentralized, tamper-proof databases for any election on opn.vote. They store all election-related data, including the election description, registrations, votes, public keys, and start/end times. After the election is completed, the private key for decryption is also published here. The use of a blockchain ensures that no election data can be tampered with, maintaining the integrity of the entire voting process on opn.vote.

Transaction Relay. opn.vote utilizes a Transaction Relay service that enables gasless blockchain transactions, allowing voters to participate without needing to hold cryptocurrency. It sponsors and relays voting transactions to the blockchain.

3.1. Implementation challenges

The development of opn.vote faced several technical challenges. For storing votes and essential election data, we chose Gnosis Chain as a blockchain due to its low transaction costs (currently a vote on opn.vote costs approximately $0.0005) and EVM compatibility, which enables future migration to other Ethereum Layer 2/3 solutions. Gnosis Chain utilizes a stable gas token (xDai), which helps in subsidizing of voting transactions by reducing exposure to cryptocurrency price fluctuations while providing a sufficient level of decentralization and security for many types of elections. To enhance accessibility, we integrated Gelato, a blockchain automation network, allowing for transaction cost subsidization. This removes the need for voters to possess cryptocurrency or manage wallets, which significantly lowers entry barriers. A key innovation was the implementation of Master Tokens and Blinding Factors, from which fresh tokens and blinding factors are generated for each election. This approach simplifies key management for voters, allowing easy storage in a single QR code while maintaining security across multiple elections.

To optimize the retrieval and indexing of blockchain data (e.g., tallying votes) and IPFS data (e.g., election descriptions), we set up a The Graph node. Additionally, to prevent the selling of voter credentials, we introduced vote recasting, allowing voters to change their votes.

Another challenge was the implementation of RSA Blind Signatures. The lack of standardized JavaScript libraries for RSA Blind Signatures required us to develop a custom solution. Furthermore, on-chain verification of these signatures proved to be cost-intensive, leading us to implement a Signature Validation Server to handle the signing of voting transactions.

3.2. Current state of development

opn.vote has reached its alpha stage, with the core protocol flow fully implemented. The blockchain smart contract has been deployed on Gnosis Chain for testing purposes (0xB2971419Bb6437856Eb9Ec8CA3e56958Af45Eee9). Protocol functionalities for election creation, management, voter credential generation, voting, and tallying are fully operational. All key entities are set up and functional, with RSA and blockchain-related implementations complete. Initial voter authentication is assumed to be handled by the Authorization Provider.

3.3. Next steps

Next steps for opn.vote development is implementing eID systems (ePerso and Post-PIN) for authentication at the Authorization Provider. Creating a frontend for election creation, management and tallying. The team is also working on enabling election results and votes to be easily verified. A beta version, with publicly available source code on GitHub, is scheduled for release by the end of 2024. The production version of opn.vote is expected to launch in spring 2025.

4. Future improvements

Reduce Scheme complexity. It is planned to replace the SVS by on-chain proof of voting entitlement. We may need to switch Blind Signature encryption from RSA to Elliptic Curves.
Distributed election coordinator. To eliminate the need to trust a single election official, we plan to implement a multi-key scheme for the election's private key. This approach prevents premature revelation of election results and enhances overall security.
Decentralized electoral register. Using a multi-blind signature scheme to issue voting entitlements eliminates having to trust a single electoral register in not secretly issuing voting entitlements without absentee applications.
Proof of registration. Keeping track of digitally signed absentee applications makes the registration process auditable by being able to proof, who has registered. EUDI wallets will have individual signing capabilities no later than 2026.
Client device security. Improving the usability to easily switch between more than one personal device for voting and verification reduces the possible risk of a vote manipulation on client side by a compromised personal device.

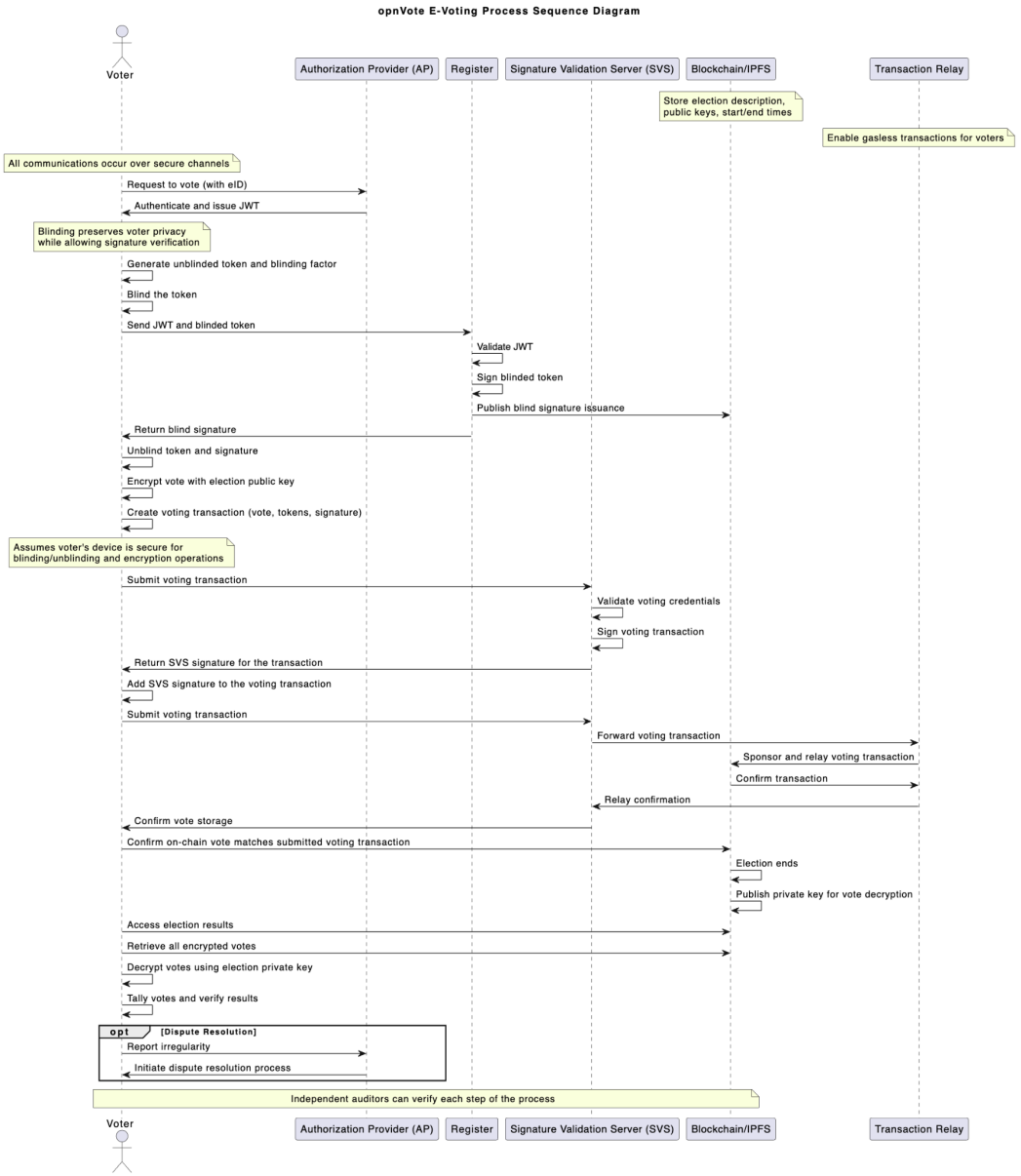
5. Conclusion

openPetition has developed a new approach to online voting systems, using a much less complex encryption scheme and newest blockchain functionality. Less complex software and less security requirements for system infrastructure helps to cut down vote transaction costs by far compared to other systems on the market. The proposed protocol offers a solution where you must not trust a single entity when it comes to privacy and verifiability. You only have to trust the protocol and the decentralized infrastructure which makes it a possible candidate for legally binding online elections.

6. References

1. Cetinkaya, O., Doganaksoy, A.:Pseudo-voter identity (pvid) scheme for e-voting protocols. In The Second International Conference on Availability, Reliability and Security (ARES'07), pp. 1190-1196 IEEE (2007)
2. Rivest, R. L. Virza, , M.: Software independence revisited In: Real-World Electronic Voting, pp. 19-34. Auerbach Publications (2016)

7. Appendix


opnVote E-Voting Process Sequence Diagram
Voter
Authorization Provider (AP)
Register
Signature Validation Server (SVS)
Blockchain/IPFS
Transaction Relay
Store election description, public keys, start/end times
Enable gasless transactions for voters
All communications occur over secure channels
Request to vote (with eID)
Authenticate and issue JWT
Blinding preserves voter privacy while allowing signature verification
Generate unblinded token and blinding factor
Blind the token
Send JWT and blinded token
Validate JWT
Sign blinded token
Publish blind signature issuance
Return blind signature
Unblind token and signature
Encrypt vote with election public key
Create voting transaction (vote, tokens, signature)
Assumes voter's device is secure for blinding/unblinding and encryption operations
Submit voting transaction
Validate voting credentials
Sign voting transaction
Return SVS signature for the transaction
Add SVS signature to the voting transaction
Submit voting transaction
Forward voting transaction
Sponsor and relay voting transaction
Confirm transaction
Relay confirmation
Confirm vote storage
Confirm on-chain vote matches submitted voting transaction
Election ends
Publish private key for vote decryption
Access election results
Retrieve all encrypted votes
Decrypt votes using election private key
Tally votes and verify results
opt
[Dispute Resolution]
Report irregularity
Initiate dispute resolution process
Independent auditors can verify each step of the process
Voter
Authorization Provider (AP)
Register
Signature Validation Server (SVS)
Blockchain/IPFS
Transaction Relay