A digital voting system for the 21st century

Davide Casaleggio[1], Vincenzo Di Nicola[2][0000−0002−6849−5500], Michele Marchesi[3][0000−0003−1540−8773], Sebastiano Missineo[4], and Roberto Tonelli[3]✉[0000−0002−9090−7698]

[1] Rousseau Association, Milan, Italy
davide@casaleggio.it
[2] Independent researcher
vincenzo@live.com
[3] DMI, University of Cagliari, Cagliari Italy
marchesi@unica.it, roberto.tonelli@dsf.unica.it
[4] Strateghia ltd, Rome, Italy
missineo@strateghia.it

Abstract. We present Terminus, a voting system based on blockchain technology. Terminus relies on technology solutions pioneered by Monero, a privacy-focused Blockchain, and on specifically designed operational procedures: this guarantees full anonymity of the vote and addresses several concerns of digital voting systems. Terminus was tested at an event of an Italian political movement, and will be used to carry out polls to drive some of the political decisions of this movement. We also introduce an evaluation framework for DLT voting systems, and use it to compare existing systems.

Keywords: E-Voting · DLT · Monero.

1 Introduction

Voting is an action we perform in several different situations: some examples are song contests (e.g., Eurovision), reality shows (e.g., Big Brother), associations/councils (e.g., residents meeting), company's decisions (e.g. shareholders meeting) or politics (e.g., country elections). Wherever allowed, digital voting systems have been introduced as tools to make it easier for voters to express their choice and to reduce the huge costs of voting in person. However, most digital voting solutions in use nowadays are centralized and affected by a number of problems (e.g., certification of results). Some solutions are available and have been adopted in some cases on traditional online voting, but these are out of scope of this paper.

Recently, the introduction of digital ledger technologies (DLT), and in particular of blockchain, led to a renewed interest in e-voting, because they provide high levels of immutability, accessibility and reliability, and are typically open source.

In this paper, we analyze the key properties a remote digital voting system must exhibit, and provide an evaluation framework to compare voting systems

based on DLT. Using this framework, we compared some of the most recent and popular e-voting systems based on DLT, with our proposal of Terminus, a voting system based on Monero technology, a privacy-preserving blockchain able to guarantee a very high degree of anonymity [1]. We then describe Terminus solution in deeper detail, and a live test which was performed after its deployment.

2 Requirements of an e-voting system

Voting systems can be used for a variety of purposes, from shareholders meetings, to contests, to political voting – both non-binding and political election. A common requirement is that only qualified voters can cast their vote, and that this vote is counted just once. In most situations, voting must be anonymous, and the voting system must guarantee this.

Though there is no accepted standard on how to evaluate an e-voting system, several recent proposals reported evaluation criteria quite similar to each other [2], [3], [4].

Starting from these criteria, we summarized them, resulting in the following criteria a remote digital voting system must satisfy:

1. **Immutability:** No one can neither delete nor modify votes. Known also as "Integrity" [2].
2. **Egality:** Each vote must be equal to each other (in some kinds of voting, however, votes might be weighted). No voter can have his/her vote counted more than once. Each voter must receive one and only one ballot.
3. **Eligibility:** Only the voter can add his/her vote; no one else can add votes.
4. **Anonymity:** No one must know what a voter has voted for, unless specified otherwise. Known also as "Privacy" [3], [4].
5. **Blindness:** During the voting session, no one must know where the votes are going to. In other words, results must not be visible in real time. Known also as "Fairness" [3], [4] or "Data Confidentiality and Neutrality" [2].
6. **No forgery:** Ballots cannot be forged, and their number must be exactly equal to the number of voters. This property can be further detailed (for instance, prescribing that a voter cannot vote more than once), but here we will consider it as a single criterion.
7. **Verifiability:** Auditors – or even voters themselves – must be able to verify that the number of ballots is exactly equal to the number of voters, that each voter has received one and only one ballot, and that votes are correctly counted. Known also as "Auditability" [2].
8. **Cost:** Deployment, management and maintenance cost are reasonably low. Known also as "Affordability" [4].
9. **Scalability:** The system is able to manage very many voters, even political elections.

We added three more criteria to the above quoted ones, targeted to practical implementation and usage of the e-voting system. They are the **10. Stability** of the approach, that is the probability that the system is long-lived, the

11. Openness of the system – it must be open source, or with inspectionable code, a feature very important to get the needed trust that the system always works properly – and the presence of **12. Actual use cases**, or at least test demonstrators, of the system.

Using these criteria, we defined a framework to evaluate e-voting systems based on DLT, especially targeting political elections.

Each criterion is evaluated using an integer scale from 1 to 5, meaning that the criterion is:

1. unsatisfied or poorly satisfied.
2. only partially satisfied.
3. fairly satisfied, but it might be better.
4. satisfied for the most part.
5. totally satisfied.

It would be possible to further weight by importance these criteria, but to the purposes of this work we assume that all criteria have the same weight. The actual comparative evaluation of existing voting system and of our proposal is reported in Section 5.

3 Existing e-voting systems based on DLT

Despite the interest and the promises of DLT for implementing better voting systems, the number of actual systems in advanced development, or already deployed, is not high. Among these, the most popular and mature DLT voting systems we found are: Agora, Vocdoni, Voatz, Follow My Vote, Polys and Colony.

Agora [5] is a project started in 2015 by a Swiss-based voting technology company which developed an end-to-end verifiable voting solution for governments and institutions. Bryan Ford, who served as the Director of the Lausanne's Swiss Federal Institute of Technology (EPFL) Decentralized and Distributed System Lab (DEDIS) gave a key contribute. Agora is maintained by a team of cryptographers of Losanna Institute of Technology already accustomed with blockchain technology. It runs on a custom blockchain with various architectural levels and with three main components: Skipchain, Cotena and Valeda. Skipchain manages consensus, with high throughput and efficient transaction validation. Cotena is the component which stores cryptographic Skipchain proofs. Valeda validates Skipchain and Cotena data by means of cryptographic proofs. The Cotena layer is also used to anchor the system to the Bitcoin blockchain, since Cotena periodically stores a hash of the most recent Skipblock in a Bitcoin transaction `OP_RETURN` opcode, which enables anyone to verify that all data remained unaltered. Agora's architecture has different interconnected layers, is quite complex and is anchored to the Bitcoin blockchain. Agora piloted the first test of Blockchain voting in a national government vote during Sierra Leone Presidential Elections in 2018, where results were counted on blockchain separately from official counting after the vote took place on paper ballots.

Vocdoni [6], which in Esperanto translates to "to give voice", is perhaps the most advanced among DLT voting systems, being very recent, and based on systematic usage of Zero Knowledge proofs. Vocdoni aims to build a general-purpose voting system, seen *"as a collective signaling mechanism that gives cryptographic guarantees about its integrity and its outcome"*. Its architecture is quite complex. The voting is handled by a Tendermint blockchain called "vochain". Data integrity is provided by Ethereum blockchain, data availability is provided by IPFS/Swarm. To date, we are not aware of real use cases of Vocdoni.

Voatz [7] is one of the first voting systems, and is that with most real use cases, being used by several counties and states in the USA. It is based on an app able to perform biometric identification of the voter. The e-voting process is quite traditional, but is registered on the Voatz blockchain, built using the HyperLedger blockchain framework. The Voatz blockchain is permissioned, run by selected nodes managed by the stakeholders of the election, such as the major political parties, NGOs, non-profits and independent auditors, etc. Voatz approach is proprietary, and has been security audited by independent third parties.

Follow My Vote (FMV) [8] is an open-source project based in USA whose code is available on GitHub. Most of the code is written in Python language, and the system is based on BitShares blockchain. The system provides the voters the possibility to monitor election results in real time and also to consequently act in order to change their mind according to partial results and to change the previous vote. Depending on the election rules this feature can be turned off. Voters register with an ID card issued by a public authority and receive a ballot for voting on the specific election they qualify to vote in. It uses a Registrar to pair the ID Key with a Blinded Token for anonymous voting.

Polys [9] is a Russian voting system based on Ethereum technology. It is in advanced development, but with already several use cases because the use of its beta version is presently free. The system is patented and proprietary, though they plan to release also an open source version. The voting is performed on a permissioned Ethereum blockchain, with added nodes managed by "trusted representatives" (TR) of the voting organization, or of interested parties. The vote anonymity is guaranteed by a Shamir's Secret Sharing schema involving private keys generated by the TRs, which is used to encrypt votes. The voting choices are in turn obscured with homomorphic encryption using the exponential ElGamal cryptosystem. Voters are provided of an app to generate their private key, and cast their unique vote after exchanging information with TRs nodes. Once cast, it is impossible to change one's vote. If the number of voters is high, homomorphic decryption can have performance issues, though they can be solved by partitioning the voters across different voting systems which run concurrently.

Colony [10] is peculiar among the considered platforms, because it is more a platform for community collaboration, rather than a true voting system. It is completely based on Ethereum, and is aimed to manage the polls of decentralized communities working on this blockchain. For this reason, it does not support

anonymous voting, but only blind voting, until the poll is closed and the votes are revealed.

The literature includes many other proposals of voting systems based on DLT. However, despite the fact that some of these look quite sound and innovative, they are still under study or development. A recent paper on an e-voting system reports and describes some of these works [4]. Finally, it is worth quoting that Estonia performs e-voting using a traditional system, but with the register of voters stored on a blockchain (ksi blockchain) to ensure their integrity and to protect them against insider threats.

4 Our proposal, the Terminus platform

Work on Terminus started in 2017, as a way to use blockchain technology to increase transparency and trust of the Rousseau voting platform, used by the Italian 5-star Movement to ask its members to define political decisions.

The use of a public blockchain was quickly ruled out, because of its cost and voting recording time. In fact, the cost is linked to the price of the underlying cryptocurrency and on the number of transactions to be processed, and is way too volatile. The recording time too can be subject to unpredictable delays. For instance, November and December 2017 saw a major congestion of the Bitcoin network. Transactions remained unconfirmed for several days, if not eventually disappearing from the mempool [11].

So, we opted for a hybrid permissioned blockchain solution. In such solution:

- Sealers are nodes run by pre-authorized separate entities, which can create ("seal") transaction blocks. In addition, by choosing anonymous blockchain technologies, such as Monero or Zcash, sealers cannot distinguish data in the underlying transactions, thus preventing a malicious sealer to effectively tamper the voting session.
- Supporters are secondary nodes which can be run by everyone. They have access to the blockchain: they cannot create blocks, but can watch them and be aware if something suspicious happens.
- Rules could be put in place so that a subset of supporter nodes are eventually promoted to sealer nodes.

This approach resembles the dynamics existing at the United Nations Security Council. A set of predetermined sealer nodes (akin to the UN Security Council 5 permanent members) and a set of supporter nodes that are temporarily promoted to sealer nodes (akin to the UN Security Council 10 non-permanent members).

As the underlying blockchain to run the platform, we chose the technology behind Monero as the most suitable one for a digital and remote voting system. As a cryptocurrency, Monero proved its strength in highly adversarial environments. It has an extreme degree of privacy protection, and its community strives to increase it even further. We are by far not the first ones to think that the technological prowess behind Monero can be applied to voting. In fact, we took

Table 1. The key features a digital voting solution must satisfy.

Key feature	Solution
No external entity can add/remove/modify votes	Native to blockchain technologies
No one must know what a voter has voted (anonymity)	Ring Signatures of voters
No one must know where the votes are going to (results must not be visible in real time)	Stealth address of Vote Receivers + Vote Receivers private keys safe management by external Custodians
Ballots cannot be forged, and its number must be the same of voters	Blockchain tokens generated before voting session begins
Each voter must receive one and only one ballot	Blockchain tokens sent by the Administrator to voters before voting session begins
Auditor must be able to verify the 2 points above (number of ballots == number of voters; each voter has received one and only one ballot) without relinquishing anything in voter anonymity and vote visibility	Auditor has access to voters view keys, thus verifying that Administrator has indeed sent one token to each voter
No voter can have his/her vote counted more than once	Native to blockchain technologies
Each vote must be equal to each other	Token fungibility

inspiration from the CryptoNote protocol [1] (on which Monero is based) that uses an optimized version of the Ring Signature scheme described by Fujisaki and Suzuki [12]. The key application mentioned in the paper is actually anonymous voting.

We believe in anonymity first: this must be the main key pillar of any digital voting solution. It is important to stress that anonymity is native to the Monero protocol, and it is very well battle-tested. Other technologies, such as Bitcoin, try to achieve anonymity by adding second layers (e.g., Lightning Network); however, as of today, such incremental approaches do not provide the same guarantees as of native solutions. Table 1 reports a summary of the key features a digital voting solution must satisfy, along with their technical solutions. Roles (such as Administrator and Custodian) are described in the following section.

We forked what at the time was the stable version of Monero (v0.13.0.4). In our permissioned solution, we removed all transaction fees (i.e., they were set to zero) and all their relative checks. Also, for the sake of scalability, regarding the consensus protocol we opted for a Proof-of-Authority (PoA) approach with pre-approved sealer nodes.

Before the voting session, all sealer nodes start mining at startup with same fixed-difficulty (100), and block rewards are sent to a special wallet called "Admin wallet". Only a total of N vote tokens (where N is the number of Voters) are created. In our solution, 1 forked XMR equals to 1 vote token.

The Admin Wallet initially hoards the N tokens of the permissioned Blockchain, to be used as vote tokens; then, before the voting session starts, the Admin Wallet distributes each vote token to the N Voters, creating N transactions of 1 forked XMR as amount. Our system uses one blockchain for each voting session: this prevents people from using unspent vote tokens of a previous election in a new one. In addition, in order to further guarantee that no additional vote tokens are created, during a voting session block rewards are zeroed out. Also, if, by any chance, a "disturber" Voter sends a fractional token value to the Vote Receiver, such vote will not be counted.

We then introduced a few tweaks on the wallet side in order to allow vote transactions to be mined. To this purpose, we forked what at the time was the stable version of Monerujo (v1.10.10), a high-quality Monero light wallet [13].

We also created a dashboard where the Administrator can distribute the vote tokens (before the voting session starts), and can calculate results, without having to perform all the operations from the command line (after the voting session is over). For sake of demonstration, for the Proof of Concept the dashboard also allowed the Administrator to create Vote Receivers keys, and start/end the voting session: these are aspects that can be solved through improvements as discussed in later section 6.

4.1 Roles in the system

Terminus voting process makes use of several roles, which are key to ensure voting fairness and trust. These roles are:

- **Voters**: the people who vote. In a real-world paper voting analogy, Voters are akin to electors.
- **Vote Receivers** (for ease of readability, also simply called "Receivers"): entities who receive the votes. In a real-world paper voting analogy, Vote Receivers are akin to candidates.
- **Administrator**: entity which, before voting session begins, grants one ballot to each Voter. In a real-world paper voting analogy, Administrator is akin to poll clerks that give a ballot to each eligible voter.
- **Auditor**: entity which ensures no foul play is done by the Administrator. In a real-world paper voting analogy, Auditor is akin to scrutineers that ensure there is no malpractice. Any Voter might also be an Auditor.
- **Custodians**: entities which, before voting session begins, create Receivers private keys, publish Receivers public keys, but cannot show Receiver private keys. In a real-world paper voting analogy, Custodians are akin to militaries that protect the ballot box to be closed till the end.
- **Sealer Nodes**: Entities which run the underlying blockchain software solution and can create ("seal") blocks. In a real-world paper voting analogy, it is a combination of poll clerks and scrutineers that ensure no vote is added/deleted/modified during the voting session.
- **Supporters**: entities which run the underlying blockchain software solution but can only watch.

4.2 Voting process

Before starting the voting session, the network must be configured. To this purpose, it is needed to setup a minimum number of Sealer Nodes, able to run the permissioned blockchain. For evaluation purposes, 5 nodes are enough, possibly located on the cloud, running a modified version of "monerod", the Monero daemon software. Real polls would require a bigger number of Sealer Nodes, each managed by an independent organization, in order to ensure the stability and trust of the system.

Before each session, the Vote Receivers are set up. The Administrator had access to a server where, through a simple dashboard, s/he will:

1. Create Vote Receivers wallets
2. Start a voting session
3. Enable Voters
4. Stop a voting session
5. Calculate results

Presently, the Administrator keeps locally all the Vote Receivers keys. Of course, this is not acceptable in a real-life voting system: the proper way to address Vote Receiver key management is discussed in later section 6 with the introduction of a custodial system.

The voting session needs that voters install on their smartphones and use an application. Each voter must create a Voter wallet, send the Voter wallet address to the Administrator, receive a vote token by the Administrator, and eventually send the vote token to one of the admissible Vote Receivers (voting options).

After the voting session, the results are processed by simply counting the number of tokens received by each Vote Receiver.

4.3 Proof of concept

On March 10, 2019, at Villaggio Rousseau in Milan we showcased a simple Proof of Concept: voters were asked to pick one of four choices of food they would have liked to eat at the end of the event. Each of the food choices (pizza, apple, oranges, sweets) had a Vote Receiver wallet associated to them. There was only one voting session, and at the end results were published. Had additional voting sessions been scheduled, the whole process would have been recreated from scratch (i.e., "one voting session, one blockchain").

Before starting the voting session, we had to configure the network. For demonstration purposes, we setup 5 instances of a modified version of monerod on AWS. We bound the daemon on localhost and linked directly every Sealer to every other one through SSH tunnels. At the end, we had a total of 20 tunnels (5 nodes, with 4 connections each).

For this demonstration, the Vote Receivers were:

1. Pizza Margherita
2. Apples (Mele)

3. Oranges (Arance)
4. Sweets (Caramelle)

The voting session lasted 1 hour (from 10:00am to 11:00am). During such time, 67 attendees of Villaggio Rousseau volunteered to install and use on their Android phones the voting wallet. Each attendee ("Voter") created a Voter wallet, sent the wallet address to the Administrator, received a vote token by the Administrator, and sent the vote token to one of the four Vote Receivers.

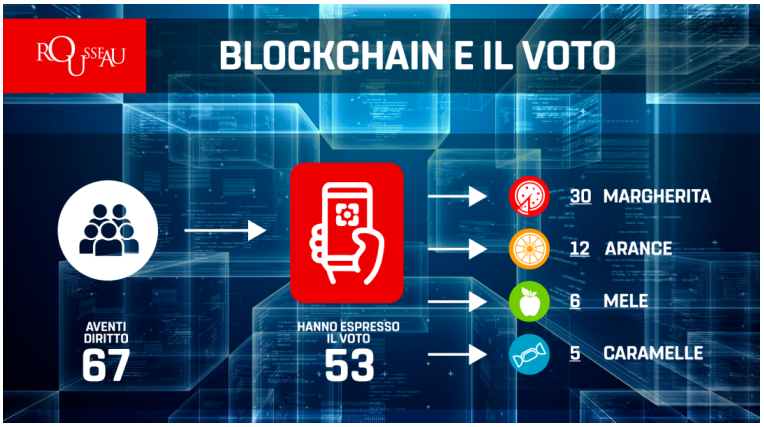

Fig. 1. The results of the demonstration voting session (in Italian).

At 11:00am on March 10, the Administrator stopped mining on each node, thus terminating the voting session. Results were immediately announced by publishing the balance of each Vote Receiver wallet. A total of 67 Voters took part to the 1-hour voting session demonstration. 53 of them actually cast a vote. The output of the vote is shown in Fig. 1. By the way the system has been designed, there is no way of knowing who the 14 people who did not cast their vote were.

5 Comparative evaluation

Using the framework reported in Section 2, we evaluated the voting systems reported in Section 3, together with Terminus. We were unfortunately unable to actually install, use and test these systems, except for Terminus. So, the evaluation is based on the information gathered collecting the information on the Web site of these systems, and on other Web sources.

The evaluation was made by polling seven blockchain app programmers, working at our department or at other firms, and taking the median value of the answers. The result is reported in Fig. 2. There is no room for a thorough discussion of these results. Basically, the scores of the voting systems do not

differ much. The most advanced systems – Agora, Vocdoni, Voatz and Polys – are somewhat penalized for being very complex and, with the exception of Vocdoni, quite closed. FMV and Colony are quite simple, and are not intended for large-scale, anonymous, blind voting. Terminus was conceived to be simple, easy to manage and scalable, hence the good score.

#	Criterion	Terminus	Agora	Vocdoni	Voatz	FMV	Polys	Colony
1	Immutability	4	4	3	3	3	4	4
2	Egality	4	4	4	3	4	4	4
3	Eligibility	4	4	4	5	5	4	4
4	Anonymity	4	4	5	3	3	4	1
5	Blindness	5	4	5	4	4	5	4
6	No forgery	5	5	5	4	5	5	5
7	Verifiability	4	4	4	3	4	4	4
8	Cost	4	2	2	2	3	2	4
9	Stability	4	4	3	3	3	4	4
10	Openness	5	3	4	1	5	1	5
11	Scalability	4	4	3	5	3	3	1
12	Actual use cases	2	4	1	5	3	4	2
	TOTAL SCORE	**49**	**46**	**43**	**41**	**45**	**44**	**42**

Fig. 2. Comparative evaluation of the considered voting systems, using the proposed framework.

Clearly, there are strong threats to the validity of the comparative analysis. The main threat is that the evaluation of most systems is not based on testing the actual system, but on information gathered on the Web. Another threat is the obvious bias of the authors, though we tried to be as impartial as possible. Nevertheless, we believe that this evaluation might be a good starting point for demonstrating the usefulness of the proposed evaluation framework for DLT-based voting systems.

6 Discussion and further improvements

So far, we have discussed the core of the technology behind Terminus. Though, in order for the system to achieve important properties of voting systems, some operational procedures must be introduced.

For example, in order to prevent visibility of voting trends, it is not enough to rely on Vote Receivers stealth addresses. In fact, if a Vote Receiver has access to his/her private keys, s/he can see in real-time the voting session results: s/he might decide to leak results, or take advantage of this knowledge. All of this can be effectively solved by introducing the role of Custodians: that is, independent

people in charge of protecting secrets. Let's consider N Custodians, and a safe environment where Vote Receiver private keys are created. These keys are then split into N shares using algorithms such as Shamir's Secret Sharing, and a threshold of M ($M \leq N$) is set. That is, it would require at least M Custodians to be able recreate the Vote Receivers private keys. The corresponding Vote Receivers public keys are generated along with the private keys, and they can obviously be shared with the world so that Voters know where to send their transaction to. Such custody schemes are today well used in the cryptocurrency world to protect wealth [14]: they involve safe procedures and hardware (e.g. HSM), which can be also directly applied in this case.

Additional operating procedures, not necessary but useful, would require the Voter to share his/her view keys with the Administrator, and the Administrator to share them to the whole public. This way, any Voter can - on his/her own - verify that the number of ballots created by the Administrator is indeed correct (no ballot forgery) and that each Voter has received one and only one ballot.

Further improvements regard voting session termination. Voting session duration must be known, and cannot be extended. For example, if it is set to last 12 hours, and the system creates blocks every 10 seconds, then the last block of the voting session must be block number 4,319. The sealers won't mine any block greater or equal to number 4,320.

A big issue which is not addressed in this work, and by any of the other considered voting system, is its ability to prevents or mitigate the risk of buying and selling votes. This bribing problem also exists in traditional remote voting (as in the case of voting via physical mail), and in other electronic voting systems. In fact, it is very easy to sell a vote in systems that use ballot paper and mail, or to sell a username and a password. The proposed solution opens up ways to mitigate the issue, and will be the main focus of future research and developments of the solution.

Finally, it is also worth mentioning that Terminus relies on the concept of digital identities, which must be created beforehand. Digital Identities Management goes beyond the scope of this voting system; however, appropriate solutions may be integrated on Terminus and improve the overall platform, both in Voter experience and in its reliability.

7 Conclusion

In this paper we described the issues of e-voting platforms using blockchain (DLT) technology, and the quality criteria such platforms should exhibit. From these criteria, an evaluation framework for these platforms is introduced and applied. We also presented Terminus, a new e-voting platform based on the privacy-preserving blockchain technology of Monero.

The elements of innovation, compared to the state of the art, are that this, to our knowledge, is the first paper explicitly targeted to analyze and compare DLT-based voting systems. Moreover, compared with existing and proposed approaches, Terminus aims to be much simpler, open and yet very scalable. This is

obtained with a solution which is solid, auditable and tamper-proof, and maintains total voters' anonymity.

Future work will be performed in two directions. The first is to extend and tune our evaluation framework, including weighting of the criteria. We will also evaluate more voting systems, using a panel of experts and a Delphi technique approach. The second direction is to make the needed improvements to Terminus, especially on the consensus protocol and protection against denial-of-service attacks or spam voting. We will also examine the introduction of the possibility to vote more than once, keeping as valid only the last vote. This would improve the resistance against voting bribery or blackmailing.

Acknowledgements

The Terminus platform was developed with a grant by Associazione Rousseau. The evaluation framework was funded by Sardegna Ricerche, project "CriptoVoting" (RICERCA 2-26), POR FESR 2014-2020, Asse 1, Azione 1.1.3, 2nd call.

References

1. Nicolas van Saberhagen. Cryptonote v 2.0, 2013. https://cryptonote.org/whitepaper.pdf, last accessed: 24 Jul 2020.
2. Stefano Bistarelli, Ivan Mercanti, Paolo Santancini, and Francesco Santini. End-to-end voting with non-permissioned and permissioned ledgers. *Journal of Grid Computing*, 17:97–118, 2019.
3. Tassos Dimitriou. Efficient, coercion-free and universally verifiable blockchain-based voting. *Computer Networks*, 174, 2020.
4. Shufan Zhang, Lili Wang, and Hu Xiong. Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *International Journal of Information Security*, 19:323–341, 2020.
5. Agora homepage, 2020. https://www.agora.vote/, last accessed: 4 Jun 2020.
6. Vocdoni homepage, 2020. https://vocdoni.io/, last accessed: 4 Jun 2020.
7. Voatz homepage, 2020. https://voatz.com/, last accessed: 4 Jun 2020.
8. Follow my vote homepage, 2020. https://followmyvote.com/, last accessed: 4 Jun 2020.
9. Polys homepage, 2020. https://polys.me/, last accessed: 4 Jun 2020.
10. Colony homepage, 2020. https://colony.io/, last accessed: 4 Jun 2020.
11. CCN. 700 million stuck in 115,000 unconfirmed bitcoin transactionse, 2017. https://www.ccn.com/700-million-stuck-115000-unconfirmed-bitcoin-transactions/, last accessed: 4 Jun 2020.
12. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, page 181–200. Springer Berlin Heidelberg, 2007.
13. m2049r. Monerujo: An android monero wallet, 2019. https://www.monerujo.io/, last accessed: 4 Jun 2020.
14. Vincenzo Di Nicola. Custody at Conio — part 2, 2020. https://medium.com/conio/custody-at-conio-part-2-21e976f86384, last accessed: 4 Jun 2020.