Digital Object Identifier 10.1109/ACCESS.2023.0322000

Impact of Decentralisation on Electronic Voting Systems: A Systematic Literature Survey

RICARDO LOPES ALMEIDA[1,2], Fabrizio Baiardi[2], Damiano Di Francesco Maesa[2] **(Fellow, IEEE)**, Laura Ricci[2]

[1]Universitá di Camerino, 62032 MC, Camerino, Italy (e-mail: ricardo.almeida@unicam.it)
[2]Dipartimento di Informatica, Universitá di Pisa, 56127 PI, Pisa, Italy

Corresponding author: Ricardo Lopes Almeida (e-mail: ricardo.almeida@unicam.it).

ABSTRACT The modernization of voting methods is a dynamic area of research currently. In the past, innovation in voting methods was limited to the automation of steps in the process through mechanical means. This changed with the introduction of commercial cryptography in the 1970s, whose applications to voting triggered a new era in this research field. Researchers used the following years to apply tools derived from cryptographic methods to build increasingly secure, transparent, and practical electronic voting systems. Despite the effort, a true remote electronic voting system was never achieved with the technology available. The introduction of Bitcoin in 2009 brought much attention to the blockchain concept that supported it. This new data model offered new levels of transparency, data immutability, and pseudo-anonymity that made it attractive and useful to e-voting researchers. Soon after, articles detailing the first blockchain-based e-voting systems were published, and the research field entered a new era. This article presents a study on the evolution of research in electronic voting systems, following a systematic literature review methodology and a chronological evolution from the first systems that employed public cryptographic concepts up to blockchain-based proposals, with the objective of detailing the evolution of the technology as a whole, as well as all the elements, centralised and decentralised, created and used to implement voting systems.

INDEX TERMS blockchain, cryptography, e-voting, survey, systematic literature review

I. INTRODUCTION

Modern democracies provided societies with levels of comfort and security that enabled their citizens to move on from simply surviving to engaging in high-level intellectual activities. This fostered scientific and technological advancements that systematically upgraded almost all societal aspects. Amid such innovations, voting as an exercise remains largely unaltered, even in advanced democracies. Voting exercises may have evolved from dropping pebbles into clay pots [1] to touching a virtual button on a touchscreen, but the essence of the process has not changed significantly. The nature of this exercise makes it a hard one to upgrade.

Regardless of the lack of innovation in the voting process itself, voting systems have been the object of academic research. Developments in computer-based cryptography opened new research avenues, with electronic voting systems being one of the beneficiaries. The mechanisation and automation of voting systems is an old topic, but advancements in this field were restricted to supporting existing voting systems.

Therefore, in essence, not much changed from the voter's perspective.

Until recently, research on this topic followed a centralised design paradigm under a "classic" server-client architecture. These systems used cryptographic methods to blind sensitive information within the system, with the bulk of the computations reliant on a central computational unit, or cluster. So far, researchers have been unsuccessful in conceiving a secure, transparent, and scalable e-voting system that also addresses voter mobility and could translate into a widely adopted real-world application. This problem increases in importance when one considers modern levels of human mobility and emigration.

An initial surge of cryptography-based voting systems was followed by a steady stream of improvements over older proposals, mostly due to technological advancements in subsequent years. This tendency remained unchanged until 2009, when the world's first cryptocurrency, Bitcoin, and the decentralised design paradigm brought along by blockchain technology [2] that supported it were introduced. In simple terms, a blockchain is a data structure replicated and distributively managed across a number of active machines, or nodes, in a computer network. Data is written into discrete blocks that are cryptographically chained linearly together, hence the name. Blockchain protocols employ the same cryptographic techniques used by centralised e-voting systems to achieve voter privacy, system transparency, universal verifiability, etc. in their basic mechanics. Therefore, it is possible to tap into these native features to implement transparency and data immutability in a voting system, something that requires dedicated resources in a centralised model. Blockchain did not need to wait too long to make its impact within e-voting research, with the first blockchain-based e-voting proposal appearing less than a decade after Bitcoin's debut. The analysis executed in Section IV-C1 explores these early proposals and how they evolved into more mature and realistic solutions over this period.

This work intends to provide a systematic analysis of the evolution of the e-voting research field, with particular emphasis on how the introduction of the decentralised design paradigm influenced academic research on this topic. To achieve this goal, we have split the publications considered into centralised and decentralised proposals, the latter denoting blockchain-based solutions solely. The analysis of these works is framed under a *systematic literature review (SLR)* [3] [4] [5] approach, with a detailed classification, selection, and characterisation criteria detailed in the following sections, framed around a set of research questions, followed by the analysis to answer them.

This article follows the structure outlined in Fig. 1:

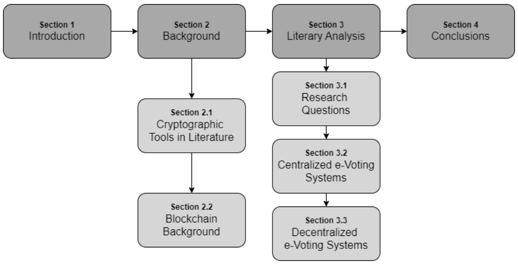

Figure 1. Survey article structure.

II. RELATED WORKS

Literature surveys on e-voting systems before the introduction of blockchain protocols are few. [6] is the oldest publication on the topic we could find. It is a brief report exploring the problem by considering the state of academic research and presenting practical cases where some form of electronic voting has been used in a real-world scenario. In 2003, most of these examples were limited to the use of DRE (direct recording electronic) voting machines. [7] presents the first academic survey of this kind, but this publication is centred on the technological details of DRE voting machines more than the theoretical academic approaches to the problem. This approach was emulated in [8], which presents a more exhaustive approach to DRE voting machine applicability as well as an analysis of the outcome of real-world elections where these devices were used. An interesting addition to this publication is the formalisation of a series of "desirable e-voting system properties", which are then used to frame the analysis of real-world implementations. [9] follows a similar strategy, defining a set of "requirements of e-voting", which follow closely on the set considered in [8]. But, as the title implies, the authors are mostly focused on the societal and technological challenges foreseen in the implementation of such systems. Regarding the formalisation of criteria used to classify e-voting systems, [10] presents the most complete analysis to date in this regard. The authors used the additional space provided by publishing their research in a book chapter to present a detailed list of classification criteria, named "Security Properties of Voting Systems", which are quite similar to the set determined in Section IV-B2a.

Contrary to the pre-blockchain era, literature surveys on blockchain-based e-voting systems are more popular. The first publication that we found that could fit this mould was [11]. Published in 2018, this white paper does a somewhat comprehensive survey on existing e-voting technology, of which the vast majority still fell under the centralised approach. The authors proceed to explore how adding a blockchain to these earlier proposals may solve some of the identified problems, but they make no attempt to review any of the early blockchain-based proposals that were already published. This article was followed shortly by more thorough surveys: [12] and [13] presented short surveys, while [14] continues the trend of defining classification criteria to characterise e-voting systems, now under a decentralised, blockchain-based approach. However, the publication set considered in that work is too small to determine any valuable trends. This publication also provides a short overview of the real-world applications of blockchain-based e-voting systems, an element overlooked in most surveys.

In [15], we found a good example of a systematic literature review covering a subset of the same publications considered in this work. The authors do formalise a set of classification criteria, following the same logic as previous authors, but their analysis focuses only on the cryptographic characteristics of blockchain-based e-voting systems and other practical elements, not on the implementation of these criteria by the solutions reviewed.

The adoption of classification criteria for e-voting systems does become more apparent in later blockchain-based surveys. Publications such as [16], [17], [18], and [19] provide good examples of this strategy, which produce quite similar works, often diverging mostly on the publication set considered for analysis.

A. RESEARCH GAPS

Blockchain-based e-voting systems have been properly surveyed during their short window of existence, but the same cannot be said for their pre-blockchain counterparts. As we have indicated in Section II, the few surveys found under this classification are mostly focused on practical applications of e-voting technology through its application in DRE voting machines, or they consider publication sets much smaller and/or limited to a single computational approach.

Another gap that we identified was the lack of a formal definition of the basic properties that an e-voting system should have, regardless of the actual implementation. [10] provides a complete analysis in this regard, but it is limited to centralised proposals. No other survey extends this analysis to both paradigms. In other surveys, the criteria set considered is still too informal and subjective to be used as a standard classification method for these works. The type and nature of the criteria used to characterise an e-voting system are highly variable from author to author, both in concrete proposals and in surveys. Though most authors tend to elaborate their list based on past references, they do take liberties to change the criteria at will, which complicates any effort to find a consensus.

Finally, a need for a chronological analysis was also identified related to the evolution of these systems, considering the overlap in terms of cryptographic techniques implemented. We consider an analysis from this point of view important, as it allows us to infer future trends for this technology.

B. OUR CONTRIBUTION

The contributions of this work are directly related to the research gaps identified. As far as we could discern, this is the first survey that covers the complete history of e-voting system development. The articles that we considered for our analysis include the earliest publications that used commercial cryptography to establish secure voting channels, up to the latest proposals based on smart-contract-enabled blockchain protocols. Along with such extensive analysis, we used a significantly larger publication set to discern chronological trends that become apparent with a systematic approach.

Our work also provides a systematisation of the set of criteria used by the selected authors to characterise their solutions. As it was indicated previously in Section II-A, there is a clear lack of standardisation around these criteria. During the systematic literature review, we grouped the criteria used by authors into a smaller set of broader criteria with the objective of establishing a more clear picture of the proper classification of these systems.

We applied a uniform characterization framework derived from the uniformization effort around the classification criteria to all publications considered. We also presented an innovative analysis that characterises the e-voting systems considered using a standardised classification method and applied the same logic to e-voting proposals from both computational approaches considered. In addition, our analysis of blockchain-based e-voting systems provides a characterization in much greater detail than previous works of this type. In this regard, we considered a much larger publication set when compared with similar works and an extended characterization set for a decentralised approach that provides a more detailed account of the evolution of research in this field. As far as we could determine, ours is the first survey of this kind to present a literary analysis with such an extended publication set, as well as framing this analysis under a research period defined by two types of computational approaches.

III. BACKGROUND

Prior to moving into the main literature analysis, we detail the cryptographic tools identified across the literature considered. These tools are transversal to both the considered paradigms and are critical for the understanding of our analysis, especially for readers that are not familiar with general cryptography.

The focus of this paper is to characterise and organise how proposals for e-voting systems use cryptographic tools to ensure the security of their usage of the systems proposed. Though there are abundant non-cryptographic methods to achieve this goal, our analysis is centred only on the ones that are based on cryptographic primitives.

A. CRYPTOGRAPHIC TOOLS IN LITERATURE

1) Threshold Systems

Threshold systems were developed to add redundancy to the sharing of secret information. Adi Shamir introduced the concept in 1979 [20] and since then it has been used in many e-voting system proposals. In *threshold systems*, a piece of encrypted data D that has been split into n pieces or shards, i.e.,

$$D = D_1, D_2, ..., D_n \tag{1}$$

can be reconstructed as long as k, $1 \leq k \leq n$ shards are known. As long as only $k-1$ pieces are known, recovering D is theoretically possible but computationally infeasible. But as long as k or more shards are known, recovering D is a trivial operation [21]. A system with these properties is classified as a *(k, n) threshold system*.

a: Homomorphic (k, n) Threshold Cryptosystems

Threshold systems define two operational domains: P, the plaintext domain and C, the ciphertext domain. A message M written in human-readable format belongs to domain P, but it is possible to switch between these domains using the encryption function E to transverse from P to C and the corresponding decryption function D to move from C to P.

Formally, it means that

$$E(M) = M' \tag{2}$$

and

$$D(M') = M \tag{3}$$

where *M'* is the message in the ciphertext domain.

Binary operations can also be defined within these domains, such as addition, subtraction, multiplication, exponentiation, etc. To differentiate between which operation is used in which domain, the symbol $\oplus$ denotes an operation in the P domain, whereas $\otimes$ the respective operation when in the C domain [21] [22].

Consider a secret s belonging to the P domain that was split into k parts such that

$$s = F(t_{i1}, t_{i2}, ..., t_{ik}) \tag{4}$$

where F is a cryptographic function used to switch between domains, such as the encryption and decryption functions, i.e., $F, F \in \{C, D\}$.

Now consider another secret *s'* that was also split into k pieces by the same function F:

$$s' = F(t'_{i1}, t'_{i2}, ..., t'_{ik}) \tag{5}$$

This encryption scheme is $(\oplus, \otimes)-$*homomorphic* if:

$$s \oplus s' = F(t_{i1} \otimes t'_{i1}, t_{i2} \otimes t'_{i2}, ..., t_{ik} \otimes t'_{ik}) \tag{6}$$

i.e., *the composition of the shares of the secrets are shares of the composition of the secrets* [23].

A practical consequence of this property is the ability to perform arithmetic operations directly on the C domain using the homomorphic equivalent operation from the plaintext P domain. The result of such an operation can then be decrypted, thus obtaining the plaintext equivalent. Consider Fig. 2 as an example:

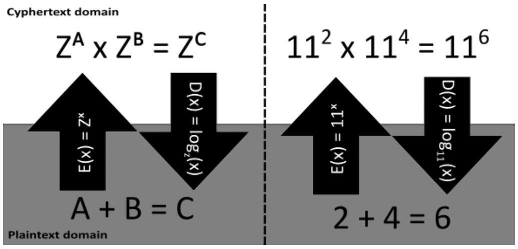

Figure 2. Example of a simple homomorphic threshold system.

In this example, data transverses to the cyphertext domain by elevating a fixed base Z to the power defined by the data itself. The encryption is defined by the function:

$$F = E(x) = Z^x. \tag{7}$$

Information can move to the plaintext domain by applying the inverse operation, i.e., the natural logarithm of the same base Z:

$$F' = D(x) = \log_Z(x). \tag{8}$$

We can verify this with the numerical example on the right side of Fig. 2. The example shows that an addition, the $\oplus$ operator, in the plaintext P domain is homomorphic equivalent to a multiplication, the $\otimes$ operator, in the ciphertext C domain.

2) Mix-Nets

Mix-nets take advantage of a property of asymmetrical cryptosystems: *when data is encrypted and decrypted using multiple pairs of encryption keys, including re-encryptions using the same key, the order in which the respective decryption keys are applied to retrieve the original plaintext* *is irrelevant as long as the same number and type of keys are respected* [24] [25].

Consider a finite set of N public encryption keys:

$$E_{set} = [E_1, E_2, ..., E_N] \tag{9}$$

and the corresponding set of decryption keys:

$$D_{set} = [D_1, D_2, ..., D_N] \tag{10}$$

Key pairs are defined by:

$$(E_1, D_1), (E_2, D_2), ..., (E_N, D_N) \tag{11}$$

Note that E_{set} and D_{set} have the same number of elements. Now consider a message M encrypted by the set of encryption keys sequentially. For simplicity's sake, assume that these keys are applied in order:

$$M^{e_{set}} = E_1(E_2(...E_N(M))) \tag{12}$$

M can be recovered using any permutation from the respective decryption key set D_{set} as long as both sets match in the number of elements and the number of applications of each key is respected:

$$\begin{aligned} M &= D_1(D_2(...D_N(M^{e_{set}}))) \\ &= D_3(D_1(D_N(...D_8(M^{e_{set}})))) \\ &= ... \\ &= D_N(...(D_2(D_1(M^{e_{set}})))) \end{aligned} \tag{13}$$

which is valid for any permutations of D_{set}.

A mix-net protects information by applying multiple levels of encryption by applying multiple keys to it and explores the fact that, to recover the initial message, the corresponding decrypting keys need to be used but only the same number of times in which the asymmetrical counterpart was used. The order in which the decryption key set is applied is irrelevant in this case.

If only one encryption key pair is used, the ciphertext can be attacked by providing a piece of plaintext data chosen to reveal statistical or language-specific characteristics of the encryption process [26]. Employing a mix-net does not eliminate this threat completely, but it does weaken it substantially.

Fig. 3 exemplifies this process. Messages *A to D* were encrypted with a series of encryption keys from asymmetrical pairs. Each message has, effectively, N encryption layers applied to it. Each server in the mixed network contains only one of the decryption keys from the set applied before; thus, it removes only the encryption that fits that key. As long as the encrypted messages are routed through the servers that can remove all encryption layers, i.e., a set of servers that have the complete set of corresponding decryption keys used in the initial encryption, at the end of this routing process, we have the original messages.

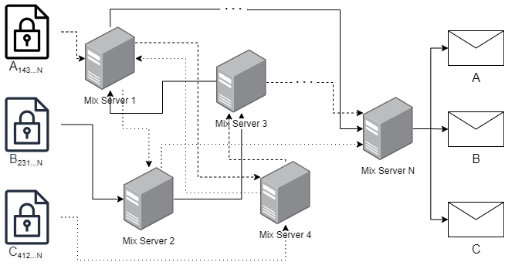

Figure 3. Mix Net scheme.

Data to mix is previously encrypted according to a secret permutation of keys in E_{set}, with that permutation transmitted to all servers in the network. Each server in the mix-net network removes one layer of encryption at a time since each server stores only one decryption key. The trajectory of the encrypted data in the Mix-net network is determined by the permutation applied during the initial encryption. As long as the number and types of encryption keys are respected, the output on the last server produces the data in plaintext format.

3) Blind Signatures

This cryptographic method was introduced in [25] in 1982 as a method to implement untraceable electronic payments. This method relies on a trusted party or element to validate a digital signature without seeing the contents of what was signed, hence the *blind* adjective. This method emulates what already happens with some vote-by-mail procedures, where eligible voters receive at home a ballot *blindly signed* by a trusted authority.

As in the original article [25], we are going to use an election example to illustrate this protocol.

Consider two actors in an exercise: a voter that wishes to validate a voting ballot M and a trustee that can validate messages. The trustee creates a pair of asymmetrical encryption keys, keeps the private key D secret, and publicises the public one E. The voter fills out his or her ballot, M, but it needs the trustee's signature. To get it without revealing its contents, the voter follows these steps:

1) The voter begins by selecting a random number K and keeps it secret.
2) The voter encrypts the random number using the trustee's public encryption key:

$$E(K) = K^e \tag{14}$$

3) The voter proceeds to blind the filled-out ballot M by applying the previous factor to it, which in this case consists of:

$$B = M \cdot K^e \tag{15}$$

   The blinding process is the binary multiplication of the message contents to be signed by the encrypted random number K, and B is the blinded message or ballot.
4) The voter provides B to the trustee to be signed. The trustee digitally signs B by encrypting it with its private key D, which actually corresponds to a decryption operation in this asymmetrical context:

$$D(B) = B^d = (M \cdot K^e)^d = M^d \cdot K^{ed} \tag{16}$$

   where

$$K^{ed} = D(E(K)) = K \tag{17}$$

   and thus

$$B^d = M^d \cdot K \tag{18}$$

   By multiplying the message by a factor only known to the voter, this effectively prevents the trustee from ever recovering M without knowing K.
5) The trustee returns the signed blinded message B^d to the voter, and he/she can easily recover a signed M by computing:

$$\frac{B^d}{K} = \frac{M^d \cdot K}{K} = M^d \tag{19}$$

   since K is only known to the voter.

Blind signatures provide decoupling between the information to be certified and the certification process itself. Using this method, it is possible to validate election ballots and other data without needing to reveal its contents, thus preserving voter privacy while adding an important layer of security to the process.

4) Cryptographic Proofs

Cryptographic proofs are protocols detailing a series of steps that must be executed between two parties - a *prover* and *verifier* - such that the *prover* can show the knowledge of a *claim* to the *verifier*. Digital signatures are good examples of cryptographic proofs: signatures are valid if the signed message is obtained after decrypting the signature with the public encryption key. This implicitly proves that the signer owns, or at least knows, the private encryption key in question.

Cryptographic proofs can be *interactive* or *non-interactive* depending on the required exchanges between the *prover* and *verifier*.

a: zk-SNARKs Proofs

Proofs that are executed without revealing any knowledge about the claim are classified as *Zero-knowledge proofs (ZKP)* and are extensively used in the e-voting context. From among the types of ZKP identified in the literature, the most frequent one observed is the *Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK)* proof. As the name implies, it is a type of *non-interactive* cryptographic proof, and its popularity in the e-voting context warrants a detailed explanation of it.

ZkSNARKs proofs are introduced in [27]; maturated by [28]; and employed formally in [29]. These proofs consist of a set of verifiable computation schemes with the following properties:

- Proofs are short and non-interactive, i.e., a verifier can be convinced with a single message from a prover.
- Private information can be used by the prover during the generation of the proof without the verifier learning anything about that information.
- The cost of verifying a proof is independent of the computational complexity of the input.

The *zkSNARKs* verification scheme is based on a polynomial-based abstraction called Quadratic Arithmetic Programmes (QAPs). QAPs are obtained using two higher-level abstractions to specify arithmetic and rank-1 constraint systems (R1CS), which are needed to verify the proof. Arithmetic circuits are obtained by flattening the compiled high-level code defining the proof. This flattening operation consists of converting complex mathematical operations (exponentiations, roots, logarithms, etc.) to simpler algebraic operations, namely additions, subtractions, multiplications, and divisions. The flattened code is then converted into algebraic gates, which are then transformed into a R1CS. These can then be interpolated (using the Lagrange interpolation formula, for example) to find the QAPs that allow the original proof to be verified [30] [31].

5) Additional Tools: Public Bulletin Board and Anonymous Communication Channel

The public bulletin board and anonymous communication channels are tools that normally pair up in the literature due to their complementary nature and usage, thus being set in the same category. They relate to the tools presented due to their use in the e-voting context and, because these are used to propagate information within the system, can have it encrypted for safety, though they do not present any new cryptographic ideas in themselves.

The *Public Bulletin Board* is an abstraction of a communication channel that can be accessed by anyone, but data can only be appended in a controllable fashion, i.e., anyone can read the append-only board, i.e., deletions are not possible, but only a few authorised entities can add new data to it. Some authors base their solution on the assumption that such a channel exists, or at least is conceivable, but not many went to the extent of providing details of a practical implementation [24].

The *Anonymous Communication Channel* is similar in terms of access control, but this one adds the property that all communications are done via pseudonyms, as well as protecting data during transmission using encryption, thus adding anonymity to the previous concept [32].

B. BLOCKCHAIN BACKGROUND

A blockchain is a distributed data structure replicated through all the active nodes in a *peer-to-peer (P2P)* network. New data can only be added in the form of a new block on the chain. Due to the decentralised manner in which data is distributed through the network, operations on it, namely appending new data, are regulated by a network-wide protocol that establishes the rules that must be followed by a node that pretends to add new data to it. In a blockchain, data is accessible to anyone, but deletions and modifications are not even available in the protocol.

This protocol regulates not just the addition of new blocks to the existing chain but also ensures that only one version of the data structure exists at all active nodes through periodic synchronisation actions. This removes any sort of centralised control over this structure, thus establishing a decentralised management structure that is able to ensure the correct functioning as well as the integrity of the data in the structure without requiring any administrators or nodes with special permissions.

Many blockchains available publicly were engineered and configured to support cryptocurrencies. As such, most of the data in these blockchains is transactional data indicating the exchange of a cryptocurrency between two addresses. All these concepts are going to be explained in greater detail, but for now, it suffices to say that these characteristics give all blockchains a set of built-in features, with *immutability*, *pseudo-anonymity*, and *verifiability* among the ones that are more relevant in an electronic voting context.

1) Blockchain Mechanics and Consensus Protocol

New data is added to the blockchain in the form of a *block*, which is, in its essence, a file filled with transactional data. For cryptocurrency-based blockchains, the transactions in a block that is about to be added to the top of the chain are waiting for confirmation, and it is the addition of that block to the chain that confirms a transaction. An active node in a blockchain network is a computational unit that keeps a full and synchronised copy of the blockchain in its permanent storage and can also *"mine"* for new blocks.

Though paradoxically distributed, a new block is appended to the chain with the authorization of the governing protocol. This append extends the official state of the blockchain, also updated by the remaining nodes. The consensus protocol decides which of the active nodes appends the new block. Section III-B1a details some popular choices of consensus protocols developed and implemented in public blockchains since the inception of this technology. In blockchains that support cryptocurrencies, the node that appends a new block typically gets rewarded with newly minted tokens from the supported cryptocurrency, a method that is popular among these blockchains to increase their total supply of tokens gradually and to provide an incentive for active participation in the network.

a: Popular Blockchain Consensus Protocols

Blockchain consensus protocols are a dynamic area of research among blockchain enthusiasts. As such, new protocols are being conceived and implemented at a fairly high rate, which makes it difficult to completely characterise the landscape. This section aims at introducing and briefly explaining the most popular protocols among existing blockchain solutions.

- Proof-of-Work (PoW) - The original consensus protocol in earlier blockchains, such as Bitcoin and Ethereum version 1.0. This protocol establishes race conditions between nodes by rewarding the first node that solves a computationally intensive cryptographic puzzle. To solve this puzzle, nodes need to find the hash digest fitting a predetermined format, and nodes are limited to brute force approaches to tackle this problem. This protocol is responsible for significant energy waste since all hash computations for losing nodes are simply discarded [33].
- Proof-of-Stake (PoS) - This protocol was created as an alternative to PoW once this one became problematic due to its high energy expenditure. In a PoS, nodes increase their probability of selection by the protocol by increasing their stake in the network, namely, by holding tokens of that blockchain in their accounts. The bigger the stake, the higher is the probability of them being selected to add the next block and receiving the associated block reward [34]. Ethereum's version 2.0 upgraded the consensus algorithm from PoW to PoS.

- Delegated Proof of Stake (DPoS) - This version of the previous protocol uses the same stake principle, but, in this case, it is used to gain voting power, which is then used to elect block verifiers, the ones that actually create new blocks. So nodes actually delegate the power they have due to their stake in the network, hence the name [35].
- Practical Byzantine Fault Tolerance (PBFT) - In this protocol, a node submits a block proposal to a principal node, which effectively acts as a network administrator, which then propagates it to multiple other backups. If enough backup nodes agree on the proposed block, it gets added to the chain. Otherwise, it is discarded [36].

These four protocols were among the first to be suggested and are now considered core protocols from which others are derived [35]. As the application range of blockchain solutions increased, so did the consensus protocols that supported them: applications used for medical decisions use a PoS-based protocol, aptly named *Proof-of-Disease* [37]. *Proof-of-Elapsed-Time* is a PoW replacing consensus where nodes have to wait a period of time generated in a trusted execution environment [38] instead of solving cryptographic puzzles. *Proof-of-Luck* consensus is based in PoW and PoS, but also the Proof-of-Elapsed-Time just referred, which creates an energy efficient and low latency for transaction validation consensus protocol [39]. In *Proof-of-Authority*, an easily scalable, reputation-based protocol, validator nodes are incentivized to maintain their position (the authority) by refraining from dishonest behaviour that could warrant them a negative reputation. This is the protocol of choice of private permissioned blockchain frameworks such as the ones from the Hyperledger family, along with the *Practical Byzantine Fault Tolerance* [40]. [34], [35], and [36] for more comprehensive surveys on this topic.

b: Transactional Metadata

Before Ethereum increased considerably the scope of applications that could be supported by a blockchain, any attempt to use existing blockchains (whose options were realistically reduced to Bitcoin) for any purposes other than transacting cryptocurrencies required creative ways to use the meagre set of options made available by this blockchain. Among the most popular ones was the usage of Bitcoin's *OP_RETURN* instruction, which allowed adding custom data, i.e., non-transactional data, to the information pertaining to a transaction.

Older, Bitcoin-based decentralised proposals analysed in this exercise use this instruction, hence the importance of clarifying how it works.

c: Bitcoin OP_RETURN Instruction

Before *OP_RETURN*, developers were already experimenting with adding custom text to a block, and they first used *Pay-to-PubkeyHash*, a standard Bitcoin script used to implement signature verification. Without getting into much detail, this script expects a hash digest as one of its inputs, and it writes it in the transaction output, specifically in the *out-script* portion of the transactional data, along with the boolean result of the signature verification operation. Users are free to provide custom data instead of a valid hash digest, and it would be written regardless of the result of the function. But because these outputs are not easily distinguishable from the *UTxO (Unspent Transaction Output)* ones used by nodes to keep up with account balances, using this instruction implies subjecting the network to additional strain. This occurs by requiring extra memory from nodes, which kept a set of UTxO in RAM for efficiency, as well as the extra computations required to determine that particular output as an invalid UTxO.

A more efficient alternative that produces the same result arrived with the *OP_RETURN* instruction, which was not part of the initial standard of the Bitcoin scripting language and was only standardised in March 2014. Unlike the *Pay-to-PubkeyHash* function, the output of *OP_RETURN* is always false, so it does not require any additional computations, and the output is not recognisable as a potential UTxO value, thus avoiding unnecessary memory storage as well. This instruction was made available for the first time with the release of Bitcoin client 0.9.0 but was limited to only 40 bytes of available text space. Release 0.11.0 increased this value to 80 bytes, and 0.12.0 established the current maximum of 83 bytes [41].

d: Cryptocurrency Wallets

Crypto wallets are the entry point of interaction for users who wish to transact within a blockchain. But the name can be misleading since crypto wallets do not store any cryptocurrencies. Instead, the cryptocurrency balance of a given wallet, identified as an account address, is determined from transactional data associated with that wallet or account rather than from a variable stored in a location or any other digital construct to that effect.

Crypto wallets allow the abstraction of creating an account in a blockchain by storing a pair of asymmetrical encryption keys in their application software or even hardware in some cases.

Transactions in a blockchain require a signature using the private key from an asymmetrical pair, and this connects balances and crypto wallets. Users sign transactions that change the balances in their accounts once they become valid in the blockchain, i.e., inserted in a block. This also means that transactions occur between wallet addresses only, which are long binary strings often represented in an alphanumeric format for human consumption, whose size depends on the blockchain where they are used [42].

Bitcoin calculates balances for each account by looking at the transactions from or to a specific address and computes the account balance by adding all the UTxO for that account. Ethereum uses an account-based system instead, where the EVM keeps a global state with all the accounts and corresponding balances, along with other relevant elements.

2) Blockchain Properties

a: Immutability

Blockchain ensures data integrity by chaining a sequence of blocks of data cryptographically. Explaining how blockchain achieves *data immutability* also explains its fundamental functioning.

A blockchain starts with a *genesis block*. As the name implies, this is the first block of the structure, and it is the only one that does not have a connection to a previous one. Subsequent blocks added after contain an element that establishes the integrity of the data and consequently the *immutability* of the data blocks: the hash digest of the contents of the previous block in the chain. This digest is a unique string, a fingerprint of the state of the blockchain up to that point. This data dependency effectively "chains" the blocks to each other while protecting the integrity of the data.

The *immutability* aspect of it derives from the impossibility of altering any data in it while keeping the sequence of hash digests intact and from the fact that blockchain data is replicated through all the machines that compose the network at a given point. To be able to successfully change the contents of a block already integrated into a larger chain, an attacker needs to:

1) Change the block contents in such a way that the new data somehow produces the hash digest to keep the hash digest sequence intact, or
2) Change the block contents and all the subsequent blocks (in order to reflect the new block hashes) in the local copy of the data in 51% of the active nodes of the network. This type of attack is known as the *51% attack*, which forces the consensus protocol to "spread" the erroneous version of the data through the rest of the network, albeit in a time window defined by a block cycle, i.e., right after the addition of the last block and

before a new one is added ahead of it.

Both of these scenarios are technically possible but highly improbable. In other words, the success probability, even if not zero, is still too small to consider them feasible. The amount of computational resources, time, and energy needed to increase the success probability to reasonable values is beyond anyone's individual capabilities. Furthermore, considering the values transacted currently on public blockchains, the cost-benefit of such an operation is prohibitive. Due to all these considerations, data in a blockchain is considered *immutable* once it gets written.

b: Pseudo-anonymity

As indicated in Section III-B1d, transactions in a blockchain are executed between two addresses and require only three elements: the sender and receiver's crypto wallet addresses and the amount to transact. This is the information that ends up written in the data blocks.

As such, cryptocurrency transactions effectively are decoupled from the true identity of the user, thus establishing anonymity as a built-in feature in blockchain as well. But this anonymization effort is not perfect. Having all transactions abstracted by a string of seemingly random bits offers limited protection. Every cryptocurrency transaction gets recorded in the blockchain, and it is relatively trivial to filter all transactions from or to a single address. From here, and considering that there are few goods and services to swap cryptocurrencies for fiat currencies at some point, it is possible to de-anonymize the final user through these regulated interfaces and/or using statistical analysis on the transactional data, hence why the feature is actually preceded by a "pseudo" for sake of correctness.

c: Verifiability

The data in a blockchain is constantly replicated through a dynamic network that grows and shrinks randomly, since each active node has an independent identity. But even in such a fluid network, the consensus protocol is able to maintain a unified image of the data structure, allowing unsynchronized nodes to converge to the consensual image among the network's majority.

This constant effort to keep a unified image, coupled with the freedom that any user, not just active nodes, has to consult the historical transactional data at will, gives public blockchains an unparalleled level of transparency and verifiability when compared with conventional centralised databases. There are many online tools available, e.g., Blockchain Explorer [43], Blockchair [44], BscScan [45], etc., that enable a user to consult data from a public blockchain without downloading the whole, or even just a portion, of the data first. Public blockchains are fully verifiable as a consequence of their architecture, without having to develop additional features.

d: Smart Contracts

[46] coined the term *smart contract* in reference to the automation of general-purpose legal contracts. In the blockchain context, this term was co-opted to refer to a code script that runs synchronously on multiple nodes on a distributed ledger [47]. Smart contracts are, by default, open source code in the sense that, if deployed on a public blockchain, the code is accessible in a human-readable format for anyone to see. Smart contract support is a feature that was only introduced in 2015 by the Ethereum blockchain. Bitcoin's blockchain had limited capacity to run scripted code, but the smart contract concept is a level above this ability.

Smart contracts execute in a distributed virtual machine, a computational abstraction of the collective resources available in the network and organised by the blockchain protocol. Because of the relative freedom that smart contracts offer to developers programming-wise, blockchains protect themselves against malicious code, i.e., code that can capture and waste network resources (infinite loops, unoptimized code, etc.), by demanding that each instruction consumes *gas*, which are units of a finite resource with monetary value associated. In the Ethereum blockchain, gas is bought using Ether, its native cryptocurrency, but gas in itself is not a subunit of Ether. Instead, it is an abstraction used to decouple the cost of gas from the high volatility associated with the price of Ether, or any other cryptocurrency for that matter. The network adjusts gas prices in order to keep a "steady price" of execution for smart contracts, but the maximum amount of gas that a transaction can consume is always defined by the user that initiates the transaction. Cryptocurrency wallet clients often abstract this process. These often automate the calculation of this gas value automatically based on the current price and the amount calculated to execute the transaction successfully. But ultimately, users always have the power to change this value.

If a transaction runs out of gas during its execution, regardless if it is stuck in a loop or if not enough gas was allocated for the full execution, the process is aborted and the state of the blockchain is reverted to its initial one. Smart contracts operate directly on the state of the blockchain network, namely, they are used to change the current state through transactions that become permanent due to the immutable nature of blockchain [48].

A successfully deployed smart contract is characterised by its deployment address. In a public blockchain, this address can be consulted, often using a blockchain explorer application, to access the contract code. To execute it, the contract needs to receive a transaction indicating the contract address, the function to execute, since one contract can expose multiple functions, any necessary arguments, and allocate enough gas for its execution. Smart contracts can call functions of other smart contracts, similar to how a normal software programme can use libraries and functions from other modules [49].

IV. LITERATURE ANALYSIS

Following the *Systematic Literature Review (SLR)* guidelines established in [3], [4], and [5], literature in both the centralised and decentralised domains was analysed in order to extract common characteristics and technological trends.

This literature analysis was split into two sections:

I **Centralised e-voting systems** For publications detailing architectures that fall under the traditional server-client model.

II **Decentralised e-voting systems** For more recent publications that used blockchain implementations to base their systems on, thus following a decentralised design approach.

A. RESEARCH QUESTIONS

1) General Research Questions

To set boundaries for this review, we established sets of research questions to be answered with the analysis of each paradigm. Due to the fact that the decentralised design evolves from the previous, centralised one, we begin by establishing a set of broader research questions that can be applied to both approaches:

On Security Criteria Implemented:

RQA1. *What are the security criteria for electronic voting systems that are consistently addressed in the research literature?*

RQA2. *Which security criteria are implemented at higher rates in the proposals considered?*

On Cryptographic Methods Identified:

RQB1. *What are the main cryptographic tools used by the e-voting systems analysed?*

RQB2. *Which cryptographic tools identified in academic proposals were used in real-world e-voting implementations?*

2) Decentralized Proposals-specific Research Questions

Proposals under the decentralised paradigm are also going to be analysed based on the security criteria, albeit with an adapted set, and the usage of cryptographic methods is also adapted to the new paradigm. The differences between them justify adding a new set of research questions to establish clear boundaries for the analysis of the latter:

On Decentralised Proposal Characteristics:

RQC1. *Which blockchain type and access control are used in decentralised e-voting solutions?*

RQC2. *Are smart contracts used in the decentralised e-voting proposal?*

RQC3. *Are there any centralising elements in the proposed solution?*

RQC4. *What are the methods used in blockchain-based e-voting systems to cast votes?*

Each section begins by establishing a frame of analysis based on the applicable set of *research questions* defined so far, establishes the criteria necessary to answer these, performs the literature analysis, and concludes with a survey of real-world system implementations and respective analysis.

B. CENTRALISED E-VOTING SYSTEMS

1) Introduction

E-voting systems under a centralised paradigm (typically a server-client architecture) concentrate all processing power into a unit or cluster of processing units. This implies that the security of the system depends directly on the resources allocated to it, namely memory, storage, processing power, etc. A user's trust in a system is directly dependent on the security of the system, which is formally divided into discrete implementation criteria. As mentioned in Section III, there are multiple strategies and tools to increase system security, but for this study, we are only interested in the ones implemented through cryptographic tools. As such, these systems can be characterised by the set of security criteria used to establish trust in the system, as well as the cryptographic tools with which they achieve this goal.

These criteria have long been proposed by researchers as a way to quantify and establish trust in the proposed system, albeit in an informal fashion. The importance of such elements is evident in all publications considered. As publications grew, a consensus began to take shape among the researchers dedicated to this area but lacking any formalization. As a consequence, it is common to see independent researchers referencing the same criterion but with different names and similar, but not identical, definitions of it. As such, the goal of this analysis is the identification and characterization of these criteria into a unified set.

The implementation of these security features often relies on cryptographic methods that have been developed alongside these systems. Electronic voting systems are just one of the potential applications of the cryptographic methods developed from the ideas presented in [50]. The *Diffie-Hellman Protocol* was the first implementation of such ideas [51], which was soon followed by applications still used today such as Secure Shell (SSH), Transport Layer Security (TLS), Secure Sockets Layer (SSL), Public Key Infrastructure (PKI), Internet Key Exchange (IKE), and Internet Protocol Security (IPSec), among others less known [52], [26] [53].

2) Classification Criteria for e-Voting Systems

The majority of works presented use a classification framework to compare their solution with previous works. This type of analysis becomes more frequent as more e-voting systems are proposed through academic publications due to the enrichment of the set of previous works used for comparison.

[54] in particular, presents one of the first attempts to formalise a "generic set of criteria in concept to existing security criteria" related to several other computer systems with strong security components. Publications prior to this one, such as [55], [56], and [57], used criteria to evaluate the solution proposed from a security standpoint, but only [54] did it explicitly and formally. As such, this publication has been referenced by later authors to introduce their own

set of classification criteria.

Though this logic may appear to have a common origin, in subsequent publications we were able to verify that the authors continue to diverge from a common set of listing criteria that varies highly in number and detail. Considering the lack of a common framework of analysis for these systems and the considerable size of the set of articles that we selected for analysis, we took the opportunity to execute a statistical analysis to determine the most common criteria considered among the pool of authors.

We analysed the publications set, and we were able to extract two sets of security criteria standing on different levels of importance in the e-voting context. The first set was considered *minimal* and contains the most common criteria found among them. The number of criteria considered varies significantly with authors, given the subjectivity associated with their choice. We also detected redundancy in larger criteria sets. The following statistical analysis defines each criterion considered so that it matches as closely as possible, starting from a base definition retrieved from the cited authors.

The second set was found to be not as related to security itself, but perhaps more applicable as usability enhancers. These criteria, from here on considered *additional*, can improve the experience of a voter with the system but do not increase the system's security, unlike the *minimal* set, while still consuming significant resources.

We follow these remarks with an enumeration and discrimination of the security criteria identified. This list also provides the answer to research question **RQA1**.

a: Minimal Classification Criteria for Voting Systems

The following list contains a set of common criteria identified in the literature whose definitions were compiled from the individual explanations provided by the analysed works.

1) **Accuracy.** An accurate voting system does not allow changes to a vote after submission, changes to the final tally, or for invalid votes to be counted. Accurate voting systems also have the ability to detect and remove erroneous votes without invalidating the whole election. Accuracy can be quantified as the inverse of the overall probability of obtaining an erroneous final result [58], [59].
2) **Eligibility.** A voting system implements eligibility if it allows *only* registered or valid voters to submit votes. This also means that, in the final tally, only one vote per voter is counted [60].
3) **Privacy.** If a voting system is able to remove all the links between the voter's personal information, which may be required for *eligibility* purposes, and the vote submitted while preventing voters from disclosing what votes they cast, it is considered private. [61], [62].
4) **Verifiability.** Verifiable voting systems allow for independent confirmation of the final tally. This criterion is sometimes split into *Individual* and *Universal Verifiability* regarding the scope of the ability [22], i.e., if individual votes can be verified or if it is limited to the final tally [63]. Since the details that differentiate between these are outside the scope of this document, any system that supports any kind of verifiability is counted as such, regardless of its specific type.
5) **Robustness.** Robust voting systems are able to prevent, or even withstand, dishonest voters, either individually or working in a coalition, in order to prevent the successful and correct completion of a voting exercise. Otherwise, they at least increase the probability of such an event being detected, either by design or by implementing consequences that make such act not profitable [64].

Table I summarises the grouping strategy used to deal with the high variability of denominations found in the publication set. It presents a summary of our findings in this regard, where publications are listed according to the exact denomination used to denote the criterion implemented. Due to the high degree of freedom each author used to name these properties and the limitations of space available, only the authors that mention the criterion under the considered name are listed, as well as the two most popular designations with the list of authors that used them. Definitions used by smaller groups of authors were simply indicated as such.

In accordance with the majority of publications, these criteria are also referred to as *security criteria*, given that the implementation of such features in an e-voting system directly translates into a more secure and reliable system.

b: Additional Criteria for Voting Systems

The next list details criteria deemed as *additional* according to the classification defined above in Section IV-B2. These were implemented at a much lower rate than the *minimal* set. So, a detailed analysis such as with the *base* set was not justifiable.

1) **Convenience.** A voting system is convenient if it takes measures to ease the voter's experience with the exercise itself. This can be achieved in a number of ways, such as an intuitive interface, the availability of related information, both about the system and the exercise (candidates, options, etc.), or anything that can improve the voter's experience [119].
2) **Flexibility.** Flexible voting systems allow multiple voting methods, such as one of many, binary choice, rank choice voting, etc., with minimal reconfiguration, as well as supporting multiple vote casting, where a voter can cast multiple ballots and the system only considers the last one submitted as valid [63].
3) **Mobility** A voting system addresses mobility if it does not restrict a voter geographically, i.e., voters can access and use such a system from either multiple alternative locations or from anywhere they can connect to it [120].

c: Voting Scope

Scaling centralised systems upwards is a resource-demanding operation since, in this context, it requires an increase on the existent resources in order to maintain the same security level while dealing with a higher throughput.

The majority of authors were explicit regarding the scope to which the proposed system was applicable, discerning between small-scale (often referred to as *boardroom elections*) and large-scale voting exercises (national-wide referendums and elections, for example). This is an important criterion to take into consideration since it also directly relates to security. Large voting exercises are intrinsically less secure due to the amount of computational resources that need to be diverted from implementing security features in order to scale up the system, whereas smaller ones are able to deliver greater security at the expense of scalability. This logic assumes a limit to how many resources can actually be allocated due to physical or economic constraints. Scalability is not a problem for centralised models if there are potentially infinite computational resources, but this is an unrealistic assumption, hence the importance of considering the voting scope in this analysis.

Some authors are explicit regarding the scope intended for the system proposed; others opted to address scalability in their solutions, which allows us to assume the solution is fit for large-scale elections. But in some cases, it was not possible to determine this with reasonable certainty. We decided not to assume anything in such cases.

3) Search Process

The evolution of this research field establishes a chronological search window. The bulk of the articles relevant to this problem were published between 1980 (right after the introduction of commercial cryptography) and 2010. Research in centralised e-voting systems did not stop completely after Nakamoto's 2009 publication [2], but did diminish significantly.

Table I. Statistical analysis of the classification criteria

Minimal Classification Criteria				
Accuracy	Referenced as:			Also referenced as:
	"Accuracy" in:	"Correctness" in:	"Completeness" in:	Integrity ([65], [66], [67], [68], [69]), Data Integrity ([70]), System Integrity ([54]), Tally Correctness ([71], [72], [73])
	[74], [75], [76], [77], [78], [79], [80], [81]	[55], [62], [56], [82], [73], [61], [83], [84], [85], [67], [86], [87], [88], [69]	[57], [71], [89], [90], [60], [91], [77], [92], [93], [94]	
Eligibility	Referenced as:			Also referenced as:
	"Eligibility" in:	"Authentication" in:	"Collision freedom" in:	Soundness ([71], [72], [89]), Validity ([86]), Voter qualification ([95]), Operator Authentication ([54])
	[57], [76], [90], [96], [60], [91], [79], [17], [77], [97], [68], [98], [99], [100], [101], [67], [87], [88], [69], [92], [102], [93], [103], [104], [94], [81], [105]	[90], [77], [92], [93]	[56], [82]	
Privacy	Referenced as:			Also referenced as:
	"Privacy" in:	"Anonymity" in:	"Confidentiality" in:	Ballot secrecy ([103]), Ballot privacy ([77]), Election secret ([87]), Maximal Ballot secrecy ([106]), Voter anonymity ([79], [70], [95])
	[57], [76], [90], [96], [60], [91], [79], [17], [77], [97], [68], [98], [99], [100], [101], [67], [87], [88], [69], [92], [102], [93], [103], [104], [94], [81], [105]	[90], [77], [92], [93]	[56], [82]	
Verifiability	Referenced as:			Also referenced as:
	"Verifiability" in:	"Transparency" in:	"Auditability" in:	Full-Traceability ([107]), Dispute-freeness ([106]), Verifiable elections ([55]), Voting verifiability ([108], [102])
	[57], [63], [82], [76], [109], [110], [90], [83], [96], [60], [65], [111], [91], [78], [79], [17], [77], [112], [97], [68], [98], [85], [101], [67], [86], [66], [113], [80], [88], [92], [114], [115], [93], [116], [103], [104], [94], [81], [117], [118], [105]	[112], [17], [117], [113], [80]	[99], [79], [17], [80], [112]	
Robustness	Referenced as:			Also referenced as:
	"Robustness" in:	"Fairness" in:	"Coercion-Resistance" in:	Receipt-Freeness ([61], [82], [119], [22]), Consistency ([112], [17]), Data Integrity ([70], [17]), Dependability ([112]), Irrevocability ([111]), Soundness ([91], [93]), Uncoercibility ([88], [118])
	[110], [22], [60], [91], [94], [115], [88], [66], [67], [101], [77], [87], [68], [92], [103], [86], [81]	[76], [71], [72], [73], [89], [60], [115], [88], [66], [67], [101], [97], [77], [87], [68], [92], [108], [103], [104], [17], [94], [118], [105], [106]	[83], [92], [66], [67], [101], [77], [97], [68], [102], [93], [103], [105]	

The investigative process begins by using the broadest of academic search engines, Google Scholar, with *electronic voting system* as the main query. The division between centralised and decentralised systems is still limited to the present document. Academically, there is no formal division, which was revealed by the uselessness of adding the term *"centralised"* to the main query. As such, it is necessary to determine which centralization model is implemented from the details in the document. Since there is a chronological division between the usage of each design paradigm considered, the publication date is often a good indicator. An overview of the system specifications is enough to determine the correct publication set where a given article should be included. For example, the use of blockchains in the solution is a clear indication that the proposal belongs to the decentralised set. On the other hand, if the authors are explicit in mentioning the use of public bulletin boards and/or anonymous communication channels, which were determined to be redundant when also using a blockchain, it is a good indicator that the solution fits into the centralising approach instead.

We also use citations from publications to infer the potential impact of some of the articles in the set considered. Researchers operate on a continuum and are often the ones paying greater attention to new articles in their area of expertise, which are often mentioned in their publications through citations. As such, if a publication is frequently cited by others of the same nature, this is often a good indicator to consider that publication for further analysis. This method enables us to establish a "seed" set for each paradigm, from which a more thorough analysis follows.

Using this approach, the search process is able to identify **133** articles that fit in the definition of a centralised e-voting system.

4) Criteria for Inclusion and Exclusion of Articles

The goal of this study is to characterise implementable systems. This means that only articles that provide experimental results from a practical implementation, i.e., a prototype and/or a software simulation of the proposed system, are considered for future analysis. Though software simulations are not as complete as a functioning prototype, they provide enough useful information for our analysis.

By following these criteria, we exclude articles without a practical implementation that could produce objective metrics under the assumption that these present ideas are still too far from being implementable, which means that the current state of the system can change significantly in the future. The initial set considered in Section IV-B3 includes several highly cited articles that were added due to their apparent popularity. But on close inspection, these revealed that they established relevant cryptographic tools in the e-voting setting but did not present any system proposals by themselves. Therefore, these fall outside the bounds considered and were removed from further analysis. [50], [121], [122], [25] and [123] are good examples of such articles.

Only **33** publications remained after applying this criterion to the initial set.

a: Data Extraction and Mapping Process

Determining if a given criterion was considered in the solution presented was not complicated in most cases. Most authors, especially more recent ones, have adopted the classification criteria strategy to validate their solution when compared to others.

But the relevance that each researcher attributes to this aspect varies considerably. We identified a spectrum of approaches from the examples in the literature set: some authors limit themselves to only mentioning the existence of these criteria, typically in non-dedicated sections (e.g., Introduction, Discussion, etc.), with a generalised reasoning for the implementation of a given criterion in the solution. Other works dedicate entire sections to a single criterion, which include a proper definition, implementation details, and even mathematical proof of their implementation. Given the variability of approaches, we devised a comparison framework to evaluate each proposed system based on how researchers implemented each of the *minimal* criteria considered in their solution.

Each publication was classified according to each of the five criteria from the *minimal* set, according to the following levels of implementation:

- **Weak** - The criterion is only mentioned in the publication in a non-dedicated section (such as Introduction or Result Analysis) with no additional details provided nor explicit proof of its implementation in the solution presented, or if the criterion had to be inferred through the implementation details, i.e., without any explicit mention in the system details. In some instances, authors did a thorough exposition of the criterion, which according to this classification would render it not weak, but the authors admit themselves that their implementation is a "weak" one when compared to other solutions, even providing details for that classification. In these cases, the judgement from the authors prevails over our own classification.
- **Normal** - The criterion is explicit in a dedicated section or series of, while also providing a basic definition, at least, of exactly what the authors interpret the criterion to be (given that some authors use the same name to indicate different criteria, e.g., "fairness" has been used to indicate "accuracy" in some cases and "robustness" in others), but omitting or providing only poor implementation details.
- **Strong** - The criterion is introduced in a dedicated section or group of sections, along with an unambiguous detailed definition, specific implementation details, and/or experimental results or conceptual proofs supporting the claim.

This comparative analysis was only applicable to the *minimal* classification criteria due to the high rate of their implementation. The criteria in the *additional* set registered lower implementation rates as well as more coarse implementation details. We were not able to retrieve enough detailed data to justify a comparative analysis of the previous sort for this case. As such, the analysis for the *additional* set considers a binary option to all remaining criteria except for the election scope. A ✓ is used to denote its presence in the publication considered, with the absence of any marks denoting otherwise. The election scope can be Small, Large, or omitted if the authors did not take into account the scalability of their solution or the solution details are not sufficient to infer about this parameter.

b: Systematic Map

Applying the logic defined above to the set of publications indicated in IV-B4 produced Table II. The analysis matrix contains the articles organised chronologically by their publication year.

c: Analysis of Results

The results obtained are consistent with the expected limitations of a centralised paradigm that requires substantial resources to implement security. A more thorough analysis allows us to answer the research question **RQA2** formulated in Section IV-A1.

The concentration of resources and the dependence on their availability regarding adding additional security features limit their implementation. This dependence becomes even more evident when the election scope is considered: proposals that do consider scalability typically implement fewer security criteria. One publication that stands out in this regard is [75], which suggests a scalable system while implementing the most security criteria among all large-scale proposals. This would disprove the rationale if not for the fact that, when compared to most of the other large-scale publications, [75] is the most recent one. This means that the authors had access to more powerful hardware than any of

Table II. Quantitative criteria analysis for centralized e-voting systems.

References	**Classification criteria for electronic voting systems**								Election Scope
	Accuracy	Eligibility	Privacy	Verifiability	Robustness	Convenience	Flexibility	Mobility	
Cohen and Fisher (1985) [55]	Strong		Strong	Strong					Small
Benaloh and Young (1986) [56]	Strong		Strong		Normal				Large
Boyd (1990) [124]		Weak	Normal	Weak	Strong				Large
Nurmi et al. (1991) [125]	Weak		Weak	Weak	Weak	✓	✓	✓	Small
Iversen (1992) [126]	Normal	Weak	Strong		Strong	✓	✓		Small
Fujioka and Okamoto (1992) [57]	Normal	Strong	Strong	Strong	Strong				Small
Benaloh and Tuinstra (1994) [62]	Strong		Normal						Large
Sako and Kilian (1994) [63]				Normal			✓		Large
Park and Itoh (1994) [32]	Weak								Large
Niemi and Renvall (1995) [127]		Weak	Weak	Weak			✓		
Baraani-Dastjerdi (1995) [96]	Normal	Normal	Strong	Normal	Normal				Large
Radwin (1995) [119]			Normal	Normal		✓			
Cramer et al. (1996) [110]			Weak	Weak	Weak				Large
Okamoto (1996) [128] Okamoto (1998) [129]			Strong	Weak					Large
Cramer et al. (1997) [109]	Weak		Normal	Weak	Weak		✓		
Herschberg (1997) [90]		Normal	Strong		Strong		✓		Large
Juang and Lei (1997) [71]	Strong	Strong	Strong	Weak	Strong				Large
Sako and Kilian (1998) [82]	Strong			Strong			✓		
Ku and Wang (1999) [89]	Strong	Normal	Strong	Normal	Strong				Small
Niemi and Renvall (1999) [130]	Strong		Weak	Weak	Weak		✓	✓	Small
Hirt and Sako (2000) [61]	Strong		Normal		Strong		✓		
Lee and Kim (2000) [60]	Weak	Weak	Weak	Normal	Normal				Large
Magkos (2001) [131]			Weak		Strong				Large
Rjašková (2002) [22]		Strong	Normal	Normal	Strong		✓		
Juang et al. (2002) [73]	Strong	Strong	Strong		Normal				Large
Cranor and Cytron (2002) [74]	Normal		Weak	Weak	Normal	✓	✓	✓	Small
Ibrahim (2003) [65]		Normal	Weak	Normal	Normal			✓	
Joaquim et al. (2003) [75]	Normal	Normal	Normal	Weak	Normal	✓		✓	Large
Moran and Naor (2006) [132]	Weak		Weak	Strong	Strong				Large
Chaum and Van De Graaf (2007) [133]	Normal	Normal	Normal	Normal	Normal				
Adida (2008) [134]			Normal	Normal		✓		✓	Small
Araújo et al. (2010) [83]	Strong		Normal	Normal	Weak		✓		Large
Locher and Haenni (2016) [84]	Strong		Normal	Weak	Strong				Large

their older counterparts, which can explain the completeness of their system.

There is a limiting balance between security, transparency, and scalability determined by the resources available to the central unit, and most authors seemed to try to find the optimal balance between these three aspects. It seems that secure and transparent systems are bound to serve only small voting exercises, as supported by [74], [125], [130], and [126]. These proposals implement the most security criteria of all, yet they are also limited to small-scale exercises.

On the other end of this spectrum, we have the proposals from [62], [56], [63], [32], [128], [129], and [132]. These authors proposed scalable e-voting systems suitable for large-scale voting exercises, but this feature was costly in terms of security and transparency.

Taking only security and transparency into consideration for now, the preference among authors seems to be for the latter. An analysis over table II shows that the majority of authors gave priority to *Privacy* and *Verifiability*. These are related to the transparency of the system, as opposed to *Accuracy* and *Eligibility*, which are more security-oriented. This is consistent with the notion that trust in the system derives directly from the voter's experience with it and the inherent need that voters have to ensure that their vote is counted. It is suggested that regular voters are more comfortable watching their ballot disappear into a sealed box than having it encrypted and saved on a hard drive. The authors seem to be aware of this, hence the prioritisation of transparency over security.

Privacy and *Verifiability* were the most implemented security criteria, both with an implementation rate of 88.24% and 79.41% respectively. The remaining security criteria from the *minimal* set are implemented at the following rates: *Accuracy* was implemented in 64.71% of the cases, *Robustness* presents a 70.59% implementation rate, and *Eligibility* was the least concerning security criterion, being implemented in just 41.18% of the cases considered.

Convenience, *Flexibility* and *Mobility* had implementation rates of 18.2%, 30.3% and 24.2% respectively. It appears that most authors prefer to pool their limited resources where they could have a greater overall impact, and that is in features that translate directly into security.

5) Cryptographic Tools in e-Voting Systems under a Centralised Model

a: Data Extraction and Mapping Process

The set of publications considered was also reviewed regarding the usage of cryptographic methods usually employed to implement the security criteria indicated, with the objective of answering the research question **RQB1**. These include methods such as *homomorphism*, *blind/ring signatures*, *mix-nets*, *cryptographic proofs* and, in this particular centralised approach, *public bulletin boards* and *anonymous communication channels*. A detailed explanation of these methods was included in Section III-A.

b: Systematic Map

Applying the process to the publication set yielded Table III.

6) Analysis of Results

It is unclear if any of the cryptographic methods and tools considered are superior to the others for the centralised paradigm. *Blind signatures* and *mix-nets* are both used to implement privacy, and, as such, these were never implemented in the same solution. But *blind signatures* seem to have preference when compared to *mix-nets*, which is consistent with the fact that *mix-nets* are easier to implement but at the expense of higher resource overhead. In a centralised scheme, it seems authors prefer resource economies and/or more nimble systems at the expense of higher implementation complexity.

The utilisation of *homomorphic* properties seems to have attracted only a third (33.33%) of the authors. This is similar to the utilisation of *blind signatures*, these with a rate of usage of 31%, both significantly higher than the usage of *mix-nets* at 19%. The relatively similar usage

Table III. Cryptographic methods and tools identified in the literature.

	Cryptographic Tools				Auxiliary support tools		
References	Homo-morphism	Blind/Ring Signatures	Mix-Nets	Cryptographic Proof	Public Bulletin Board	Anonymous Communication Channel	Election scope
Cohen and Fisher (1985) [55]				Interactive	✓		Small
Benaloh and Young (1986) [56]				Interactive	✓		Large
Boyd (1990) [124]	✓	✓					Large
Nurmi et al. (1991) [125]				ANDOS protocol	✓		Small
Iversen (1992) [126]	✓						Small
Fujioka and Okamoto (1992) [57]		✓				✓	Small
Benaloh and Tuinstra (1994) [62]				Interactive			Large
Sako and Kilian (1994) [63]	✓		✓		✓	✓	Large
Park and Itoh (1994) [32]			✓		✓	✓	Large
Niemi and Renvall (1995) [127]				Zero-Knowledge	✓		
Baraani-Dastjerdi (1995) [96]		✓			✓	✓	Large
Radwin (1995) [119]		✓					
Cramer et al. (1996) [110]	✓				✓	✓	Large
Okamoto (1996) [128] /Okamoto (1998) [129]					✓	✓	Large
Cramer et al. (1997) [109]	✓					✓	
Herschberg (1997) [90]		✓				✓	Large
Juang and Lei (1997) [71]		✓			✓	✓	
Sako and Kilian (1998) [82]	✓					✓	
Ku and Wang (1999) [89]		✓			✓	✓	Small
Niemi and Renvall (1999) [130]				Zero-Knowledge	✓		Small
Hirt and Sako (2000) [61]	✓				✓	✓	
Lee and Kim (2000) [60]	✓				✓		Large
Magkos (2001) [131]					✓		Large
Rjašková (2002) [22]	✓				✓	✓	
Juang et al. (2002) [73]		✓				✓	Large
Cranor and Cytron (2002) [74]		✓			✓		Small
Ibrahim (2003) [65]		✓					
Joaquim et al. (2003) [75]		✓					Large
Moran and Naor (2006) [132]					✓	✓	Large
Chaum and Van De Graaf (2007) [133]	✓		✓				
Adida (2008) [134]	✓		✓		✓		Small
Araújo et al. (2010) [83]			✓	Non-interactive Zero-Knowledge	✓		Large
Locher and Haenni (2016) [84]	✓		✓	Non-interactive Zero-Knowledge	✓	✓	Large

rates of these methods used primarily to establish privacy do not point out an evident advantage of one method over the other. Complexity-wise, they are similar in terms of resource usage [135] [136].

Another pertinent observation is that solutions that use a higher number of these methods are limited to small-scale elections, whereas systems that support larger elections are architecturally simpler, reinforcing the notion that scalability is a limiting factor in centralised solutions. On that note, notice that *mix-nets* are more popular with scalable proposals, whereas systems that address only small-scale exercises prefer to use *blind signatures* for privacy purposes. This may be because *mix-nets* require more resources, but only to a point. Building a network of mixed servers is a high price to pay, resource-wise, for a small-scale exercise. But if the mix servers have good performance, once they are set, a network of this kind can serve progressively larger voting exercises without additional equipment, hence their preference for large-scale events.

Cryptographic proofs are not a popular alternative in the centralised paradigm. These are used to establish system transparency, but, just like with *homomorphism*, they are not an easy concept to grasp. Also, these types of proofs were not as popular during this development phase. Once it was clear how useful these proofs were in decentralised contexts, they received a significant increase in development and usage. The lack of popularity and development around these tools also justifies their lack of implementation in these types of e-voting solutions.

Knowledge-Proofs introduce new levels of complexity, which may nullify any transparency gains in the long run, but for the few authors that did use them, the preference seems to be for the *Zero-knowledge* variant. *Zero-knowledge proofs* are sealed regarding the exchange of information between the prover and verifier, which are particularly useful for dealing with sensitive voting data.

Regarding the last two auxiliary support tools, the majority of authors included the *public bulletin board* and *anonymous communication channel* in their solutions. The bulletin board is an abstraction of the ideal communication system, and the anonymous channel adds a necessary layer of privacy to all communications within the system. In most cases, the authors simply assume the existence of such tools, but in a few, such as [60], [110] and [109], the authors did reveal some architectural details of the feature, but always at a high level.

Fig. 4 summarises the statistical analysis of the rate of implementation of the classification criteria considered.

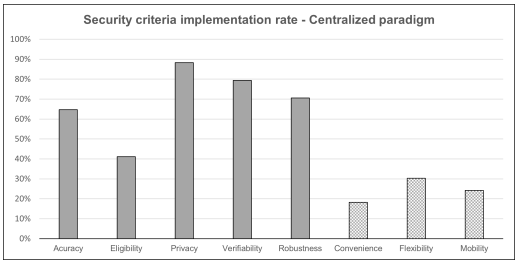

Figure 4. Statistical analysis of implementation rate.

7) Real-World Implementations of Centralised e-Voting Systems

This section details a series of cases in which voting exercises with binding consequences were executed with, or aided with, electronic voting systems based on the centralised architectural paradigm. The main goal of these experiments was to determine the viability of the tool, security problems, public acceptability, etc.

a: E-Voting Projects in Switzerland

As a means to determine the impact of increased availability on voter turnout, election costs, and potential security risks, the Swiss government established a series of e-voting trials centred around national referendums between 2004 and 2005.

The system was unable to exchange data between cantons, the semi-autonomous regions in which Switzerland is divided, but it was nonetheless successful in anonymizing the votes and decoupling sensible voter information from the vote itself [137].

The voting system was tested before deployment by a team of independent experts, with every identified flaw corrected before attempting the real exercise. The Swiss government successfully ran five independent trials in three cantons, all in national referendums. The first trial occurred in the Geneva canton in 2004. This exercise resulted in the lowest voter turnout percentage of all the online exercises, but the system was still able to attract 21.8% of all eligible voters. The last of these trials happened in the Neuenburg canton, and this one recorded votes from 68% of all eligible voters, indicating its acceptance by the population. Switzerland had among the highest levels of Internet penetration in Europe at the time, which may have contributed to the success of the online voting alternative [138].

b: e-Voting in French Elections

The French government used the elections for the Lower Chambers of the Parliament in 2012 to trial an e-voting platform made available for expatriate voting. The French constitution had to be amended in order to address the legality of the new voting mechanism.

French elections operate in rounds, and a successful identification in an earlier round carried on to the subsequent ones. All political parties involved in the election appointed observers, which joined a panel of supervisors with relevant governmental positions to observe the election process. The team of supervisors guaranteed that the vote count before the exercise's start was zero and that the final tally was consistent with the combined number of online and traditional voters.

To test the system, a large-scale exercise with 15,000 volunteers was organized. During it, cybersecurity attacks were carried out on purpose by the National Agency for Security of Information Systems (ANSSI) to evaluate its response to outside attacks. Though the results were not published for security reasons, apart from problems derived from software compatibilities between the system and the voter's own systems, as well as problems with the identification process, the test was considered a success, and the e-voting system was approved for use in future elections [139].

c: Internet Voting in Canada

The first Canadian trials with Internet voting happened as far back as 2003, during the leadership elections for the Canadian New Democratic Party (NDP). The measure was successful and replicated in other cities and municipalities. In 2011, a national survey showed that 85% of the country's population is favourable to an electronic alternative to complement the existing paper ballot system.

There is scant information about the actual architectural implementation of these systems, with emphasis on the plural. Over the years, Canadians did trials with ever-changing systems, either due to security patches or simply updated once newer, faster hardware was made available. But considering the era in which these trials happened, it is safe to assume that they were conceived following a centralised, server-client architecture [140].

d: The Estonian I-voting System

Estonia was not the first to try remote e-voting systems, but it was the first country in the world to do so at a national level with binding consequences. The *I-voting* system has been used widely in the country since 2005. That, alongside a developed telecommunications infrastructure, pushed 30% of the eligible voters in the country to adopt the online alternative.

The Estonian system uses a method that emulates the double-envelope system used in traditional voting. Voters can authenticate themselves using their national-issued ID card, which itself already possesses cryptographic capabilities on its own that can be used to both authenticate and protect ballot data, or a 2-factor authentication method that uses SIM card-based authentication.

The *I-voting* system was the first to use multiple vote casting as a measure to protect voters against coercive behaviour: the system allows multiple ballots to be cast by a single voter, but only the last one is counted. This allows voters to replace ballots cast under external coercive influence or if they simply changed their minds. But a physical vote always supersedes any online ones [141].

Independent analysis revealed that the Estonian system was actually mired in security problems. System integrity was implemented by a series of protocols that were heavily dependent on human interaction, when these could have been automated with additional effort. When human errors eventually happened, these resulted in system errors and attacks. Since the complete source code was never published, it was impossible to guarantee the integrity of the tabulated results, which in itself is a significant lack of transparency.

Because it is cheaper and simpler to maintain an online system than a traditional one, online voting in Estonia remains available for a longer period than the paper alternative. As an example, the October 2013 elections had a 7-day window for online voting. The outcome of this approach is mixed: on the one hand, the increased voting window allows voters more flexibility, giving them more time to make or change a decision. But on the other hand, keeping an unsecured system online for longer periods of time only increased the chances of success from attackers, especially for time-dependent attacks [142].

The security problems with the Estonian e-voting system were thoroughly detailed in [143]. The conclusion was that the system was planned poorly from an architectural perspective.

e: The Norwegian E-vote Project

Per request of the Norwegian government, a team of international experts from the International Foundation for Electoral Systems (IFES) independently oversaw extensive tests of the Norwegian *E-vote* online solutions in a series of less consequential elections, such as local referendums and youth council elections.

The Norwegian approach is similar to the Estonian one: both systems take advantage of a national-wide, government-issued identification system with built-in cryptographic capabilities. The Norwegian MiniID system is a 2-factor authentication system enabled and available by default to all Norwegian citizens, requiring only an active mobile phone number to receive a one-time password to register. The *E-vote* system also addresses voter coercion with a multiple vote casting feature, similar to the Estonian case [144].

Regarding the difficulties and problems, this solution was also plagued by the same problems already identified in a similar, centralised model, namely, software incompatibilities between the centralised interface and the multitude of options with which a voter might interact with the system. To mitigate them as much as possible, the *E-vote* system was developed to be accessed as a public web application. Developers had to take into account all the potential operating systems and Internet browsers at development time, which increased the system complexity without any gains in security, transparency, and/or scalability [137].

f: The New South Wales iVote System

The Australian region of New South Wales introduced its centralised e-voting system in March 2015. The development and deployment of the *iVote* system were supervised by the New South Wales Electoral Commission (NSWEC). Results were positive in terms of popular acceptance, with 5% of all votes being cast with the tool. The percentage itself is not significant in the overall exercise, but taking into account the population density of the region (which includes Sydney, one of Australia's largest cities), that percentage amounts to over 280,000 votes, which is more than any of the other exercises considered thus far.

The *iVote* system was also subjected to an unscheduled security review by an independent team of experts, which identified several flaws that stem from the centralised nature of the application, namely:

- Susceptibility to *downgrade-to-export* attacks, where the system can be "tricked" to swap a high-security communication protocol for a low-security one that can then be attacked more easily.
- Use of the breakable Transport Layer Security (TLS) protocol when the Secure Sockets Layer (SSL) protocol was already available at the time.
- Code permeability, which allowed for injections of malicious code through the voter's browser

Though these problems were identified before the real exercise, their solution, through code patching, occurred during the voting period, which created a time window in which the system could have been attacked with the security flaws still active. 3177 votes decided one of the races in that election. Further calculations revealed that the vulnerability window could have affected around 66,000 votes, which could have easily produced erroneous results in the election.

Though it seemed unlikely that the *iVote* system had been perverted during that exercise, the damage to the perception of trust by these tools could have been disastrous if such an attack had taken place. Besides these problems, the system behaved as intended, and the population was satisfied with its performance [145].

8) Conclusion

This analysis is able to elucidate regarding research question **RQB2** in section IV-A1, namely, how the cryptographic protocols and methods identified during the initial analysis had translated to real-world applications. And the answer is that they do not appear to. Despite all the work and ideas that originated the cryptographic methods, their effect on real-world applications seems marginal at best.

This analysis was somewhat superficial due to the proprietary nature of the solutions presented. Most of the publications reviewed were only suited to serve small-based voting exercises, but every real-world solution considered, even if not available to all eligible voters, processed a number of votes far beyond a number that could qualify these exercises as "small". This by itself, alongside the unsolved inherent limitations associated with a centralised model, excludes the utilisation methods hard to scale up, such as *Blind Signatures* or *Cryptographic Proofs*. The absence of mentions of these methods in either the documentation available or any of the independent reports produced supports this hypothesis.

It appears that the involved governments ordered software solutions for private companies, and these devised solutions based on existing and tried approaches from other industries, such as e-commerce and e-banking applications, and modelled the voting process around them. Given that all these exercises were moderately successful, even with security flaws identified, the strategy employed is not completely unfit but does ignore the academic research. The authentication methods employed are a testament to this assumption: 2-Factor Authentication schemes are becoming the norm whenever sensitive information, such as account numbers and balance, or personal information, is in play.

The real-world implementations analysed are pertinent examples of how important an alternative to traditional voting is. Though they do seem to ignore most academic research on the subject in the last few years, they still represent an important starting point for a transition between theoretical ideas and practical implementations.

C. DECENTRALISED E-VOTING SYSTEMS

1) Introduction

Before Bitcoin, several authors tried to propose ideas for implementing currency in a digital format, but all proposals ended up retaining a certain level of centralization. For example, [25] is often regarded as the first work to introduce the concept of cryptocurrency, as it first proposed to use cryptography to validate transactions rather than protect them. The main technical problem of a truly digital currency is to provide a decentralised way to mint new coins and prevent double spending. The Bitcoin protocol was the first to successfully solve both problems in a truly decentralised fashion, which popularised Bitcoin as the first purely digital alternative to traditional finance. Blockchain uses the same cryptographic methods referred to thus far, but it employs them in a fashion that permits fully decentralised applications to operate.

The following analysis assumes a degree of knowledge regarding blockchain and concepts related to it. Section III contains a detailed explanation of every concept referred to from this point onwards.

2) Search Process

The search process in this case follows the same approach as in Section IV-B3, but with the keyword *blockchain* added to the initial search query. This search returned a significant number of results, which became the initial seed for analysis. After a review of each element of this initial set, with the most relevant and consensual references identified using the number of citations as the main metric, the search expanded to include other popular academic search engines, such as *IEEExplorer*, *Scopus* and *ACM* to follow up on the most relevant citations.

This process extracted a total of **75** articles for additional analysis.

3) Criteria for Inclusion and Exclusion of Articles

Each of the 75 articles from the previous set is analysed taking into consideration their implementability, i.e., if a working prototype is presented, or alternatively, a software simulation with experimental data, but also relaxing these criteria to include proposals that specify an implementable system. Most researchers were able to present prototypes or a proof-of-concept based on their proposal, but we also take the liberty to consider higher-level approaches that could realistically be implementable in a future opportunity. This group of publications is analysed to answer the general set of research questions posed in Section IV-A1, as well as the more specific set defined in Section IV-A2. Regarding the specific set, the implementation details required to answer these could also be verified directly from either the implementation details of the solution or determined implicitly from other characteristics indicated in the proposal. For example, if authors explicitly support their solution on Bitcoin's blockchain but omit the intended voting scope, we can assume that the proposal is limited to small-scale elections due to the inherent scalability problems associated with the blockchain used.

Applying this set of criteria to the initial set reduces it from 76 to a pool of **63** publications, a smaller reduction from the initial set than with the previous paradigm.

4) Data Extraction and Mapping Process

We organized publications by their publication year highlighted any time-based patters present.

The following analysis answers each one of the research questions posed in Section IV-A1 and IV-A2. Some questions, such as **RQC2** and **RQC3** for example, have binary responses. In the interest of consistency, a publication where the concept at hand was verified is indicated with "✓", none otherwise. Questions with a larger set of possible answers, as the mechanism to abstract votes, for example, are discriminated against, with a more detailed explanation following in a future section.

5) Classification criteria and Systematic Maps

a: Adapted Classification Criteria

A preliminary analysis revealed that blockchain-based e-voting proposals were notoriously absent of *Convenience* and *Flexibility* criteria, while we realised that a new additional criterion (*Uniqueness*) should be included.

The lack of references to *Convenience* and *Flexibility* is justified by the development of frontend technology, not only from the user standpoint but also regarding the experience and knowledge required from developers to implement convenient and flexible interfaces for their proposed systems. The majority of centralised proposals were developed when user interfaces required significant time to implement. Internet applications were in their first version (Web 1.0), characterised by static interfaces and a higher degree of centralization regarding content development. As such, a mature and functional interface required a substantial time investment from the developers.

By the time these decentralised proposals appeared, the Internet had already transitioned to "Web 2.0", an informal label indicating a higher level of customization from the side of the user regarding online applications. This was possible due to the availability of development frameworks that significantly decreased the knowledge required to create a convenient and flexible frontend interface. For example, for an earlier centralised proposal, creating a simple interface emulating a voting ballot on screen (a simple list of names with interactive selection boxes next to each) required a frontend engineer to write, potentially, every element on screen from scratch. By the time the first decentralised proposals appeared, the same frontend engineer had a myriad of frontend frameworks at his or her disposal, as well as templates and even out-of-the-box customizable solutions that allowed him or her to develop a much more convenient and flexible interface with a fraction of the time and effort than before.

The time and effort dedicated to the implementation of *Convenience* and *Flexibility* criteria diminished considerably when compared with the centralised model, so the authors began omitting from their solutions due to the diminished impact these now have.

Uniqueness refers to the ability of a system to prevent multiple vote casting, i.e., the imposition of one voter, one vote, as opposed to some proposals that allow voters to change their minds over the voting period and cast a replacement vote if desired. This criterion plays with the balance between security and flexibility, so its implementation should not necessarily be considered a positive trait, but it is a relevant characteristic of an e-voting system.

As such, the set of security criteria used in table II was adapted to optimise the security characterization of decentralised proposals.

b: Systematic Map

The logic described was condensed in Tables IV and V.

c: Analysis of Results

Similar to what was observed with the centralised proposals, *privacy* and *verifiability* continue to be the most implemented criteria, but due to different motivations and consequences. As indicated in Section 1, blockchain's inherent transparency implements *verifiability* by default, which revealed itself as one of the main reasons why many of the authors chose to build on it. In contrast, the high rate of *privacy* implementation is explained by the same factor: the transparent nature of blockchain requires more attention towards the protection of data, especially voter data, that gets written into it. This explains the increased importance of the *privacy* criterion, given how, in this context, more transparency means less privacy. The *eligibility* criterion was more implemented in the decentralized model than in the previous centralized approach due to the anonymous nature of the usage of the underlying platform. Anonymity, albeit *pseudo* as was mentioned in section III-B2b, is another default feature of public blockchains. This clashes directly with the highly regulated nature of most

Table IV. Criteria analysis for decentralized e-voting systems.

References	**Classification criteria for electronic voting systems**							Election scope
	Accuracy	Eligibility	Privacy	Verifiability	Robustness	Mobility	Uniqueness	
Barnes (2016) [146]			Weak	Normal		✓	✓	Large
Kirby (2016) [147]		Weak	Normal	Normal	Weak		✓	Large
Zhao and Chan (2016) [111]			Normal	Weak	Weak	✓		Large
Cruz and Kaji (2016) [91]	Normal	Normal	Normal	Normal	Strong	✓	✓	Small
Ben Ayed (2017) [78]	Weak	Normal	Weak	Normal		✓		Small
McCorry (2017) [148]	Strong	Strong	Normal	Strong	Normal	✓	✓	Small
Bistarelli (2017) [79]	Normal	Normal	Normal	Normal	Weak	✓	✓	Small
Lee (2017) [70]		Weak	Normal	Normal	Normal			Small
Shaheen (2017) [149]			Normal	Strong			✓	Small
Wu (2017) [77]	Normal	Normal	Normal	Normal	Strong			Small
Liu and Wang (2017) [112]			Normal	Normal	Normal	✓	✓	Large
Hardwick et al. (2018) [97]	Normal	Strong	Normal	Normal	Weak	✓	✓	Small
Wang et al. (2018) [150]	Normal		Strong	Normal		✓	✓	Small
Koç (2018) [151]			Weak	Weak		✓	✓	Small
Hsiao (2018) [95]		Weak	Weak	Weak				Large
Chaieb et al. (2018) [68]	Weak	Normal	Normal	Normal	Weak			Large
Khan et al. (2018) [98]		Normal	Normal	Normal	Normal		✓	Small
Zhang et al. (2018) [99]	Strong	Normal	Normal	Normal	Strong	✓	✓	Small
Lai et al. (2018) [152]	Weak		Weak	Weak	Weak			Large
Dagher et al. (2018) [153]		Weak	Weak	Weak	Weak			Small
Bartolucci et al. (2018) [100]		Strong	Strong		Weak		✓	Small
Hjálmarsson et al. (2018) [154]		Weak	Weak		Normal		✓	Small
Shukla et al. (2018) [155]		Weak	Weak	Normal	Weak		✓	Large
Khoury et al. (2018) [156]		Weak	Weak	Weak	Weak	✓	✓	Small
Perez and Ceesay (2018) [157]		Weak			Weak	✓		Small
Yu et al. (Yu2018) [85]	Normal	Strong	Strong	Normal	Strong	✓	✓	Large
Matile et al. (2019) [158]			Weak	Strong		✓	✓	Small
Vo-Cao-Thuy et al. (2019) [101]	Normal	Normal	Normal	Strong	Normal	✓	✓	Small
Bosri et al. (2019) [159]	Normal	Weak	Normal		Normal	✓		Large
Yi (2019) [160]		Weak	Weak	Weak		✓	✓	Small
Singh and Chatterjee (2019) [161]		Weak	Normal	Weak		✓	✓	Small

voting exercises, therefore researchers devoted considerable effort to ensure that the anonymity offered by blockchain could not be exploited to pervert any elections based upon it. This problem is directly relevant to the degree of centralization of the solution analysed, and it is explored in greater detail in Section IV-C6.

Robustness and *accuracy* saw similar implementation rates as with the previous paradigm, but *mobility* had a higher one in this case. The interaction with the system is done via purely digital methods, a consequence of the digital implementation of blockchain, so *mobility* arrives almost as a default feature from using this element as the base development platform. Though most authors were not explicit about this criterion itself, they did describe how to interact with their proposal for voting, and in most

Table V. Criteria analysis for decentralized e-voting systems.

References	**Security criteria for electronic voting systems**							Election scope
	Accuracy	Eligibility	Privacy	Verifiability	Robustness	Mobility	Uniqueness	
Adiputra et al. (2019) [67]	Weak	Weak	Weak	Weak	Weak	✓	✓	Large
Murtaza et al. (2019) [162]	Weak	Weak	Weak	Weak	Weak			Large
Lyu et al. (2019) [86]	Normal	Weak	Normal	Normal	Normal	✓	✓	Small
Seftyanto et al. (2019) [163]		Normal		Normal	Normal		✓	Large
Chaieb et al. (2019) [66]	Strong	Normal	Normal	Strong	Normal	✓	✓	Small
Faour (2019) [87]	Normal	Normal	Normal	Normal	Normal	✓	✓	Small
Lopes et al. (2019) [113]		Weak	Normal	Normal	Normal			Small
Zhang et al. (2019) [164]		Weak				✓	✓	Small
Mols and Vasilomanolakis (2020) [88]	Normal	Weak	Normal	Strong	Normal	✓	✓	Small
Yang et al. (2020) [69]	Strong	Strong	Strong	Strong	Normal	✓	✓	Small
Chaieb and Yousfi (2020) [92]	Normal	Strong	Strong	Normal	Normal		✓	Large
Sadia et al. (2020) [80]	Weak	Weak	Normal	Strong	Normal	✓	✓	Large
Killer et al. (2020) [114]		Weak	Normal	Normal		✓	✓	Small
Shao et al. (2020) [165]		Weak	Weak			✓	✓	Small
Xu and Cao (2020) [108]			Strong	Normal		✓	✓	Small
Zaghloul et al. (2020) [102]		Weak	Normal	Weak	Strong	✓	✓	Large
Zhang et al. (2020) [115]	Weak		Normal	Normal	Normal	✓	✓	Large
Dimitriou (2020) [93]	Normal	Normal	Weak	Normal	Normal	✓	✓	Small
Alvi et al. (2020) [116]	Weak	Weak	Normal	Normal	Normal		✓	Large
Han et al. (2020) [103]	Weak	Weak	Weak	Normal	Normal	✓	✓	Small
Zhou et al. (2020) [104]	Normal	Normal	Strong	Normal	Normal	✓	✓	Large
Vivek et al. (2020) [17]	Normal	Normal	Weak	Normal	Normal	✓	✓	Large
Takabatake et al. (2021) [94]	Normal	Normal	Normal	Normal	Normal	✓	✓	Large
Larriba et al. (2021) [81]	Strong	Strong	Normal	Strong	Strong	✓	✓	Small
Li et al. (2021) [107]			Strong	Strong		✓	✓	Large
Verma (2022) [117]		Weak	Weak	Weak	Weak	✓		Small
Mani et al. (2022) [166]		Weak	Weak			✓	✓	Large
Alvi et al. (2022) [118]	Normal	Normal	Normal	Normal	Strong	✓	✓	Large
Hu et al. (2022) [105]	Normal	Weak	Normal	Weak	Normal	✓	✓	Large
Hassan et al. (2022) [167]		Weak	Weak	Weak	Weak	✓	✓	Large
Vidwans et al. (2022) [168]		Weak			Weak		✓	Large
Li et al. (2022) [106]			Normal		Normal	✓	✓	Small

cases, it was easy to understand that the access could be done remotely, which allowed us to infer about its implementation. Others, such as [163], [118] and [159] proposed adaptations to existing EVM-based traditional voting methods with blockchain schemes, the latter used as a recording and tallying element, which negates *mobility*.

Decentralised proposals were explicit regarding the way they dealt with multiple votes, with the vast majority of them disallowing the practice. Some authors simply prohibit casting any votes after the first one, such as [101], [159], [161], [17] or [168], whereas proposals such as [152], [94] and [112] allow for multiple votes to be cast, but the system only considers the first one. In these examples, the *uniqueness* criterion was inferred from the system's characteristics, namely how the authors detailed how they dealt with multiple votes cast, whereas in other cases, the proposal was explicit regarding the implementation of such a criterion.

Lastly, we observed a similar balance between the implementation of security criteria and the scope of the election, namely, the more security criteria were implemented, the smaller the scope of the election that the proposal supported. Only three proposals, [102], [115] and [105] claim to satisfy the majority of security criteria considered while claiming to be suitable for large-scale voting exercises. But a detailed analysis reveals that [102] and [115] proposed solutions without specifying the blockchain technology they intend to use, which omits which consensus algorithm is employed, an important element to determine scalability. [105] does present a complete system alongside a prototype. But looking at the experimental data provided, one can see that they implement their prototype in both Ethereum's Ropsten and Rinkeby test networks with significant differences in performance. Rinkeby uses Proof-of-Authority as a consensus algorithm, whereas Ropsten uses Proof-of-Work, and the experimental results are consistent in that regard. The PoA network is more than three times faster than its PoW equivalent, which introduces an important caveat to their claims. Their solution might be suitable for large-scale elections, as long as it is implemented on a PoA blockchain.

In fact, with the decentralised proposals, it was possible to infer the election scope by looking at the type of blockchain and respective consensus algorithm used, even if the scope was not explicitly stated in the text. As a rule, solutions based on PoW consensus were considered limited to small-scale elections only due to the known scalability problems derived from this consensus protocol. In contrast, solutions based on private blockchains, especially if implemented through high transactional throughputs frameworks such as Hyperledger Fabric, were considered suited for large-scale elections, unless the authors specified otherwise.

The graphical analysis of these results, with detailed implementation rates for each criterion, is presented in Fig. 5.

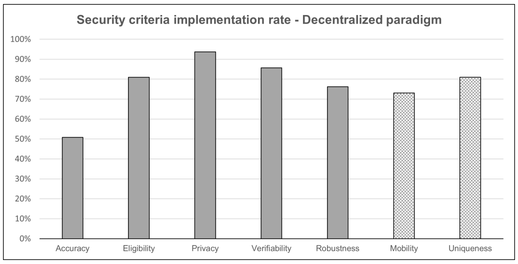

Figure 5. Statistical analysis of implementation rate.

The use of a blockchain in an e-voting solution promptly nullifies the usefulness of cryptographic methods such as a *public bulletin board* and *anonymous communication channel*. The blockchain's transparency regarding data access makes these methods redundant in a decentralised context. As such, we removed these methods from consideration in the decentralised analysis. Also, proposals that did not use or specify any of the cryptographic methods considered are omitted, which reduced the initial set considered from *63* to *34* publications.

d: Systematic Map

The results from the application of the classification strategy indicated are compiled in Table VI, also organised in chronological order.

e: Analysis of Results

Decentralised proposals use homomorphic properties of cryptosystems at a slightly higher rate than their centralised counterparts, namely 45.5% against 34%, explainable by the public nature of most of the blockchain implementations considered. In a typical system, vote data is encrypted as soon as possible, ideally after the voter has made the choice. It then remains

Table VI. Cryptographic methods identified in decentralized literature.

References	Cryptographic Protocols			
	Homomorphism	Blind/Ring Signatures	Mix-Nets	Cryptographic Proof
Zhao et al. (2016) [111]				zk-SNARKS (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge)
Cruz and Kaji (2016) [91]		✓		
McCorry (2017) [148]				Schnorr and On-out-of-two Zero Knowledge Proofs
Shaheen (2017) [149]	✓	✓		
Liu and Wang (2017) [112]		✓		
Hardwick et al. (2018) [97]		✓		
Wang et al. (2018) [150]	✓	✓		Groth-Sahai Non-Interactive Zero-Knowledge Proof
Hsiao (2018) [95]	✓			
Chaieb et al. (2018) [68]	✓			
Zhang et al. (2018) [99]	✓			
Dagher et al. (2018) [153]	✓			
Bartolucci et al. (2018) [100]			✓	
Perez and Ceesay (2018) [157]	✓			
Yu et al. (2018) [85]		✓		Non-specified Zero-Knowledge Proofs
Matile et al. (2019) [158]	✓			
Murtaza et al. (2019) [162]				zk-SNARKS
Lyu et al. (2019) [86]		✓		
Chaieb et al. (2019) [66]	✓	✓		
Lopes et al. (2019) [113]	✓			
Zhang et al. (2019) [164]	✓	✓		Non-Interactive Zero-Knowledge Proof
Mols and Vasilomanolakis (2020) [88]				zk-SNARKS
Yang et al. (2020) [69]	✓	✓		Non-specified Zero-Knowledge Proof
Chaieb et al. (2020) [92]			✓	Non-Interactive Zero-Knowledge Proof
Killer et al. (2020) [114]	✓			
Shao et al. (2020) [165]		✓		Non-Interactive Zero-Knowledge Proof
Xu and Cao (2020) [108]				Non-specified Zero-Knowledge Proof
Zhang et al. (2020) [115]	✓	✓		
Dimitriou (2022) [93]				zk-SNARKS and Pedersen commitments
Han et al. (2020) [103]				Non-specified Zero-Knowledge Proof
Zhou et al. (2020) [104]		✓		
Takabatake et al. (2021) [94]				Zerocoin's Zero Knowledge Proof system
Li et al. (2021) [107]	✓			Non-Interactive Zero-Knowledge Proof
Hu et al. (2022) [105]		✓		
Li et al. (2020) [106]				Non-specified Zero-Knowledge Proof

encrypted for as long as the process does not depend on any of the encrypted data. But whenever any counting needs to happen, either partial calculations or the final tally, vote data needs to be decrypted to perform these types of arithmetic operations, unless a homomorphic cryptosystem is used instead. In this scenario, using a threshold cryptosystem with homomorphic characteristics is an advantage since it allows executing operations in the ciphertext domain, which reduces the decrypting operations to the final tally only. This concept was present in the centralised proposals, but it is more evident when a blockchain, particularly a public one, is used. Regarding the cryptosystems used for this purpose, from the authors that did specify them, the majority are divided between the ElGamal cryptosystem [123], as in the case of [158], [150], [68], [69], [114] and [107], and the Paillier cryptosystem [169], as it is the case for [95], [99], [153], [66], [113], [164] and [115].

The usage of *blind/ring signatures* also registers an increase from 31% to 45.5% in decentralised literature for the same reasons. Blind or ring-signed ballots are a good method to privatise voter data in a public channel, like a blockchain, and this is verifiable in the analysed literature.

Both *blind/ring signatures* and *homomorphism* can be used to protect data in a publicly accessible channel, but the process in which this happens and the properties of the hidden data are different, which means that some proposals can use both without incurring redundancy: *Blind/Ring signatures* are used to protect voter data written into a blockchain, whereas *homomorphism* is used so that tallies can be computed without decrypting partial results. Proposals such as [149], [150], [66], [164], [69] and [115] implement both of these tools in their solutions.

The most evident contrast between paradigms shows itself in the implementation of *Mix-Nets*. Blockchains are not necessarily technologically comparable to these, but they achieve similar results nonetheless due to their anonymity features, which is the main objective when employing *Mix-Nets* in past proposals. Only two of the works surveyed implement structures that, though not exactly the *Mix-Net* concept defined in Section III-A2, are similar enough to be mentioned in this analysis. [100] uses a technique called *Circle Shuffle* which resembles a *Mix-Net* and, similarly to it, is used to de-link voter data from their votes, i.e., to implement *Privacy*. [92] employs a "new mix network protocol", which they claim to be used to implement coercion resistance in their solution but do not present any additional details about its implementation.

Finally, as it has been the trend so far, decentralised proposals also have a higher rate of cryptographic proof usage, and it can be attributed to the same reasons considered so far. Due to blockchain's public nature, which conflicts with the necessity of maintaining voter privacy in these systems, cryptographic proofs, with an emphasis on zero-knowledge ones, are often used to achieve verifiability while maintaining voter privacy in a public information system. Some authors are more specific than others regarding the actual type of proof used, as can be seen in Table VI, but ultimately their goal is the same.

f: Characteristic Elements of Decentralised e-Voting Proposals

Blockchain scope

Regarding their scope, blockchains can fall into one of three categories:

1) Public blockchains use encryption and sequences of cryptographic hashes to enact data privacy. Bitcoin and Ethereum are good examples of public blockchains. These blockchains provide better information transparency and auditability, but a cost in performance and through the imposition of a cost model, namely the requirement of a small quantity of cryptocurrency to be paid (gas) to operate with this type of blockchain, from cryptocurrency transactions to smart contract executions, as a protection strategy against malicious actors and unop-

timized code [170].

2) Private blockchains are opposed to the last ones, being only open to members of a particular organisation or group. External access for reading and transaction execution is forbidden. Likewise, participation as a network node is limited to members of the group or organisation. Due to these access restrictions, these blockchains do present a level of centralization in their implementation through the presence of a central authority that manages participation in the organisation and therefore in the network. A smaller network means higher block rates and transaction speeds due to less restrictive consensus protocols, as well as less communication required to achieve consensus, which also makes these networks more scalable [171].
3) Consortium blockchains sit somewhere between the two other types discussed thus far, a hybrid between private and public. These are suitable for semi-closed groups of organisations or a consortium, where the name derives from. Access to these blockchains is conditioned, just like in private chains, but the degree of data openness and access regulation varies depending on the rules agreed upon within the consortium. The nodes that compose the blockchain network are pre-selected internally, but the network can have nodes added to it if these get properly authorized. Their performance regarding block rates, transaction speeds, and scalability is also between the public and private specs [172].

Blockchain Permissions

Data in a blockchain can be restricted for *read* and *write* operations, and the permissions associated to these operations are also an important defining element for a blockchain. As defined in Section III-B1, new data can only be appended to a blockchain by mining a new block into it. This means that a blockchain that restricts *write* operations is actually preventing nodes from participating in the mining process.

A blockchain that regulates which nodes have *read* permissions is denoted as a *permissioned* blockchain. Otherwise, it is considered a *permissionless* blockchain. Regarding the *write* operation, a blockchain can be of one of the two types indicated as well, but given that there are no examples of blockchains that regulate *reads* but not *writes*, i.e., a blockchain that only allows select actors to read its data but at the same time anyone is free to join it as a mining node, from these designations, three possible scenarios follow for each blockchain:

- Permissionless reads and writes: This is the case for most public blockchains, of which Bitcoin and Ethereum are good examples. Anyone is free to read the block contents, and, theoretically, anyone can participate as a node and mine new blocks. Realistically, for PoW blockchains, the popularity of the blockchain, along with the price of the associated cryptocurrency, may create a steep entry price in terms of hardware investment to be able to actively compete with existing nodes. PoS blockchains are not as hardware-dependent as PoW ones, but the minimum stake required to be considered by the algorithm that selects the new block creator may also be outside the budget of most people. But these are economic barriers to entry. From a protocol standpoint, there are no permission-based rules preventing anonymous nodes from joining and participating in the consensus of the network.
- Permissionless reads but permissioned writes: Anyone is free to read the contents of this blockchain, but the consensus protocol is restricted to pre-selected nodes. Ripple is a good example of a public blockchain that implements this permission set: only nodes belonging to a *unique node list* are able to validate transactions and produce new blocks, while users are able to consult the data in it.

- Permissioned reads and writes: All access is conditioned in these types of blockchains. A machine needs to be granted access before it can even access the structure for reading purposes, typically from a central authority of some type.

Access is independent of the scope, but only to a point. Anyone creating a blockchain is free to create it under the scope and access permissions desired. Even though it is not practical, one can create a public blockchain with permissioned access to both reads and writes. Or a private blockchain without any read or write permissions. From a functional standpoint, these extreme cases do not make much sense, but they are still technically viable. Realistically, most public blockchains tend to be free for read and write purposes, whereas most private ones tend to regulate these operations, but this is not a strict rule.

Due to this fluidity, we opted to focus solely on the scope of the blockchain used in the proposed solution and omit its permission set since it is often not indicated in the publications.

Blockchain Types

Our analysis characterises in which one of these groups the blockchain used in each proposal falls, as well as the specific type or brand of blockchain used if indicated. Sometimes the indication of the type is enough to infer about the access control. For example, if a proposal specifies the usage of Bitcoin's blockchain, we can infer the usage of a public blockchain, given that Bitcoin does not offer any other type.

The characterization of the type of blockchain used is typically straightforward, except for the Ethereum network. Being the first network to offer explicit smart contract support and being a constant target of improvements unlike other, more static networks like Bitcoin, if a solution uses this network, it is important to specify which version or specific type was actually used, namely, if it was based on the initial, PoW-based version, Ethereum 1.0, or the newer, PoS-based Ethereum 2.0, which has appeared in later proposals. Besides these, Ethereum also provides custom implementations based on the Go programming language that can be used to implement private or consortium networks, as well as variants that use other consensus algorithms, such as the Ropsten test network, for example. This one is an Ethereum network, but it adopts Proof-of-Authority as a consensus algorithm. For the sake of simplicity, a more granular approach where only the major versions of Ethereum and any custom blockchains based on this protocol (Go-Ethereum being the most popular) are used as classes.

Some proposals do explore a blockchain-based solution from a more generalist perspective, focusing on the desired mechanics rather than the implementation details. As such, these do not specify a blockchain type, the assumption being that the solution is general enough to be implemented in most blockchain solutions available. In these cases, we indicate this option as using a *generic* blockchain type.

More recent proposals take this step further in this regard, and instead of opting for an existing solution or even assuming a generalised blockchain, they indicate the use of a proprietary blockchain, i.e., a blockchain built from scratch and tailored to the application needs. This is a consequence of the popularisation and generalisation of blockchain as a concept. With a well-defined set of basic functionalities and plenty of practical examples to use as guidance, some researchers were able to develop custom blockchains. These cases were identified using a *proprietary* blockchain type.

Levels of Decentralization

Some proposals analysed seem to be constructed just as proof that it is indeed possible to establish a fully decentralised e-voting system using blockchain alone. These proposals put themselves on the decentralised extreme of the spectrum, whereas all the publications reviewed in Section IV-B are on the opposite side.

These extreme proposals are minimal. Most publications propose a hybrid system that sits somewhere between these two extremes. Therefore, it is useful to characterise where the proposals sit in this regard, namely to infer about the advantages, disadvantages, and problems of placing the system at different points of the spectrum.

Examples of typical centralising elements that we look for include centralised databases (the most popular) and trusted third parties, such as election organisers and/or governmental agencies.

Blockchain Voting Methods

The approach taken to implement the votes themselves originated from a variety of ingenious ways to use the features offered by blockchains. This derives directly from the fact that, so far, no blockchain has been created for the sole purpose of providing support for e-voting exercises, unlike the proposals analysed in Section IV-B. The most rigid ones were created primarily as digital currency ledgers, whereas more flexible ones are similar to general-purpose computers, i.e., need extra configuration to be able to support an e-voting system. Researchers were forced to use creative methods to be able to use blockchains for that purpose. Analysing how these methods evolved is an important aspect to include in this study.

This classification exercise infers the popularity of the methods identified. During our analysis, we encountered a variety of techniques used in the publications. These were grouped into the following sets:

1) Cryptocurrency/Token Transactions: Using transactional metadata to transport non-financial information on a cryptocurrency-based blockchain is a popular approach due to its simplicity. Most blockchains provide meta fields that can be used to write information that may not be related to the transactional action. A perfect example of this is the **OP_RETURN** meta field available in Bitcoin's transaction format.
2) New Blockchain Block: Another approach is to represent a vote as a new block added to a blockchain, representing a specific voting option. This allows for quick tally computations since the size of a chain is a basic statistic in these constructs. But the feasibility of this option is minimal, especially if the target is a public blockchain, where appending new blocks is regulated by restrictive consensus protocols. But it may be a suitable option for a private blockchain with its own rules to add new blocks that do not impose a block rate as public chains do.
3) Smart Contract/Chaincode Execution: Using smart contracts to establish independent and open code that can be automatically executed to increase a count is a verifiable trend in this context. Publications based on early blockchains that did not provide this support displayed imaginative ways to use a financial tool for democratic purposes. But once smart contracts became more popular and understood, this feature was present in relevant research almost immediately. Smart contracts enable a level of transparency and human freedom that, along with the deterministic nature of their operation, increase the security of the system.

   A significant number of authors created custom blockchains to deploy their solutions, with Hyperledger Fabric among the more popular options. Hyperledger can also run scripts, which are referred to as chaincodes in that context, in a decentralised fashion, which are similar to smart contracts.

   Smart contracts are the best option considered so far for this specific purpose. Not only do they allow for the establishment of specific rules for the voting process, but they also allow for the maintenance of system-wide variables.

g: Systematic Map

Applying the mapping process and framing the analysis within the research questions defined in section IV-A2 produced tables VII and VIII.

6) Analysis of Results

The chronological order of publications shows a progression over time regarding the flavour of the blockchain used and the usage of smart contracts in the solution, which are related. Though Ethereum was available in 2015 and even the oldest of the proposals considered is from the following years, this blockchain did take a while until it was used in academic circles. The older the proposal is, the more probable it is to be based on Bitcoin, regardless of the availability of a smart contract-capable blockchain at the time.

Proposals using public blockchains migrated to Ethereum once it became available and to smart contract usage soon after. Unlike Bitcoin's, Ethereum was developed with non-financial application support in mind. There are a few exceptions: [100] and [93], publications from 2018 and 2020, respectively, that still used Bitcoin as blockchain, but other than that, the rule seems to hold. Public blockchains were maintained as a preferred choice, but in later years, there was an increased preference for frameworks that allow for the creation of custom blockchains. This can be attributed to the maturation of this technology, which makes them easier to use, and also to their increased scalability when compared to public, PoW-based blockchains. This results in innovative solutions capable of withstanding the demands of large-scale e-voting applications, even if they are still configured on mostly private networks.

All authors that based their solution on Ethereum's public blockchain also used smart contracts. Among the few exceptions, [97] did use this blockchain protocol, but there is no evidence of smart contract usage in the article. [152] used smart contracts in their systems, but it was not possible to infer with certainty if these were used to implement votes. In fact, the description of their system appears to use cryptocurrency transfers to publish voting data into the Ethereum blockchain, with smart contracts being used for other purposes. Other blockchains have been introduced that support smart contract execution, including custom blockchain frameworks in the Hyperledger family, and that added functionality is visible in the increased adoption of this technology in later years.

Another interesting observable trend is the preference for proprietary blockchains in later proposals, namely, [158], [87], [162], [160], [114], [103], and [166]. The usage of blockchain solutions conceived from scratch was not registered in any of the earlier publications, except for [78], but as the concept became more familiar and popular programming languages, such as Python or Go, began offering support for such developments, some authors seem to prefer building a blockchain from the ground up, perfectly tailored to their requirements, instead of going around the limitations of existing ones. This can also be seen as the maturation of the concept itself as it begins to permeate more and more into general computer science.

The chronological pattern that is evident in the blockchain type does not translate to the respective access control. In that regard, 25 of the 63 publications opted for private, custom blockchains, either built with the aid of software framework solutions or from scratch by the authors, whereas 34 decided for a public approach. But the chronological distribution of these does not appear to develop any evident patterns. Only four proposals, [114], [103], [167] and [166], went for consortium-type solutions. This hybrid approach seems to be more favourable in later publications, but the reason for that does not seem evident other than the maturation of the underlying implementation technology. The preference for public options is not overwhelming. The conclusion seems to be that there is a significant trade-off between network consistency and control. Public blockchains are more consistent, i.e., they offer greater availability, resilient networks, and overall higher security based on the size of the network alone. But they are also more limited when it comes to application support, do not

Table VII. Characterization of decentralized e-voting system in the literature.

References	Blockchain characteristics		Smart Contract	Centralizing Element	Vote Implementation
	Type	Access			
Barnes (2016) [146]	Proprietary	Public			Token transaction
Kirby (2016) [147]	Proprietary	Private		✓	Cryptocurrency transactions
Zhao and Chan (2016) [111]	Bitcoin	Public			Cryptocurrency transactions
Cruz and Kaji (2016) [91]	Bitcoin	Public		✓	Cryptocurrency transactions
Ben Ayed (2017) [78]	Proprietary	Private			New blockchain block
McCorry (2017) [148]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Bistarelli (2017) [79]	Bitcoin	Public		✓	Cryptocurrency transactions
Lee (2017) [70]	Bitcoin	Public		✓	Cryptocurrency transactions
Shaheen (2017) [149]	Bitcoin	Public			Token transaction
Wu (2017) [77]	Bitcoin	Public		✓	Cryptocurrency transactions
Liu and Wang (2017) [112]	Generic			✓	Cryptocurrency transactions
Hardwick et al. (2018) [97]	Ethereum 1.0 (PoW)	Private		✓	Cryptocurrency transactions
Wang et al. (2018) [150]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Koç (2018) [151]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Hsiao (2018) [95]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Chaieb et al. (2018) [68]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Khan et al. (2018) [98]	Multichain	Public		✓	New blockchain block
Zhang et al. (2018) [99]	Hyperledger Fabric	Private	✓		Token transaction
Lai et al. (2018) [152]	Ethereum 1.0 (PoW)	Public	✓	✓	Cryptocurrency transactions
Dagher et al. (2018) [153]	Go-Ethereum	Private	✓	✓	Smart Contract execution
Bartolucci et al. (2018) [100]	Bitcoin	Public		✓	Cryptocurrency transactions
Hjálmarsson et al. (2018) [154]	Go-Ethereum	Private	✓	✓	Smart Contract execution
Shukla et al. (2018) [155]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Khoury et al. (2018) [156]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Perez and Ceesay (2018) [157]	Ethereum 1.0 (PoW)	Public	✓		Smart Contract execution
Yu et al. (Yu2018) [85]	Hyperledger Fabric	Private	✓	✓	Chaincode execution
Matile et al. (2019) [158]	Proprietary	Private		✓	Token transaction
Vo-Cao-Thuy et al. (2019) [101]	Ethereum 1.0 (PoW)	Public	✓	✓	Token transaction
Bosri et al. (2019) [159]	Ethereum 1.0 (PoW)	Public		✓	Token transaction
Yi (2019) [160]	Bitcoin/Zerocoin	Public		✓	Cryptocurrency transactions
Singh and Chatterjee (2019) [161]	Generic	Private		✓	New blockchain block

allow for architectural changes, and are slower to operate than smaller custom private blockchains, mainly due to their preference for hard-to-scale PoW consensus protocols.

Regarding the levels of decentralisation, of the 63 publications selected, only 13 presented a fully decentralised solution. This question was addressed in Section IV-C5c already, since it relates to the implementation of *Eligibility*, and how these proposals used creative methods to circumvent the usage of a centralised element, such as databases or an election authority. For the remaining ones, a hybrid approach was preferred through the inclusion of a trusted third party of some kind that established eligibility criteria for its voters. A more thorough analysis of the fully decentralised proposals revealed little value in such an approach other than presenting a proof of concept. The lack of centralised elements makes it immune to cyberattacks that target a central

Table VIII. Characterization of decentralized e-voting system in literature.

References	Blockchain characteristics		Smart Contract	Centralizing Element	Vote Implementation
	Type	Access			
Adiputra et al. (2019) [67]	Generic	Private		✓	New blockchain block
Murtaza et al. (2019) [162]	Proprietary	Private		✓	Token transaction
Lyu et al. (2019) [86]	Go-Ethereum	Private	✓	✓	Smart Contract execution
Seftyanto et al. (2019) [163]	Hyperledger Fabric	Private	✓	✓	Token transaction
Chaieb et al. (2019) [66]	Generic	Private	✓	✓	New blockchain block
Faour (2019) [87]	Proprietary	Public	✓		Smart Contract execution
Lopes et al. (2019) [113]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Zhang et al. (2019) [164]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Mols and Vasilomanolakis (2020) [88]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Yang et al. (2020) [69]	Generic	Private	✓		Token transaction
Chaieb and Yousfi (2020) [92]	Monero	Public		✓	Cryptocurrency transactions
Sadia et al. (2020) [80]	Generic	Private	✓		Token transaction
Killer et al. (2020) [114]	Proprietary	Consortium	✓	✓	Smart Contract execution
Shao et al. (2020) [165]	Go-Ethereum	Private	✓	✓	Smart Contract execution
Xu and Cao (2020) [108]	Go-Ethereum	Private	✓	✓	Smart Contract execution
Zaghloul et al. (2020) [102]	Generic	Public	✓	✓	Smart Contract execution
Zhang et al. (2020) [115]	Generic	Private	✓	✓	Smart Contract execution
Dimitriou (2020) [93]	Bitcoin	Public		✓	Cryptocurrency transactions
Alvi et al. (2020) [116]	Ethereum 2.0 (PoS)	Public	✓	✓	Smart Contract execution
Han et al. (2020) [103]	Proprietary	Consortium			Token transaction
Zhou et al. (2020) [104]	Hyperledger Fabric	Private	✓		Chaincode execution
Vivek et al. (2020) [17]	Hyperledger Sawtooth	Private	✓	✓	Chaincode execution
Takabatake et al. (2021) [94]	Zerocoin	Public	✓	✓	Cryptocurrency transactions
Larriba et al. (2021) [81]	Monero	Public		✓	Cryptocurrency transactions
Li et al. (2021) [107]	Go-Ethereum	Private	✓	✓	Smart Contract execution
Verma (2022) [117]	Ethereum 1.0 (PoW)	Public	✓	✓	Smart Contract execution
Mani et al. (2022) [166]	Proprietary	Consortium	✓		Token transaction
Alvi et al. (2022) [118]	Hyperledger Fabric	Private	✓	✓	Smart Contract execution
Hu et al. (2022) [105]	Ethereum 1.0 (PoW)	Public	✓		Smart Contract execution
Hassan et al. (2022) [167]	Hyperledger Fabric	Consortium	✓		Token transaction
Vidwans et al. (2022) [168]	Hyperledger Fabric	Private	✓	✓	Smart Contract execution
Li et al. (2022) [106]	Go-Ethereum	Private	✓	✓	Token transaction

element, but that is achieved with a significant increase in system complexity and required resources.

Finally, as for the method used to implement the vote itself, there is more variability within this element than in any other. The preference appears to be using cryptocurrency transactions to store voting data in the metadata fields in the transaction format. An argument can be made for the limitations imposed in early publications by the usage of Bitcoin's blockchain that can explain this tendency. This was indeed the case for [111], [91], [79], [70] and [100], but certainly not for [97], [150], [152] or [159], which had Ethereum and the possibility to develop a custom blockchain tailored to their needs, and still they opted for using Ethereum's metadata fields to store voting data. This method is the simplest of them all, hence its popularity.

Using a blockchain block to abstract a vote was not a popular approach. This strategy implies that a block can be added to the head of the chain at will and without being affected by any block rate limits. That in itself limits these implementations to using custom blockchains to go around block rate limitations and restrictive consensus protocols in public ones. The access control is debatable but also irrelevant, since the size and members of the network are known. But this approach does require a substantial network to be available to maintain a minimum of security, availability, and, most importantly, computational capacity to add new blocks as needed. This approach is unfeasible for any non-custom, public network, mainly due to the high computational costs of adding a block, let alone being able to control when that happens. As such, the few proposals that did opt for such an approach implemented either custom blockchains, as in [78], [98] and [160] or private solutions, such as [161] and [67].

Another approach on par with this one is the use of cryptocurrency tokens to abstract votes. Fungible tokens, which include all cryptocurrencies, can be traded in fractions of a full token. As such, it is possible to establish a predefined quantity of such tokens to be counted as votes. Voters then transfer this quantity to the wallet address of the option of their choosing, and the final tally is simply the option whose wallet has the largest balance. This method takes clear advantage of blockchain's inherent features: voter authentication can be achieved by registering all eligible voters' wallet addresses beforehand, while ensuring that all vote transfers come from pre-registered addresses prevents double voting, though it can weaken voter privacy by allowing a link between vote and voter data. But this method overlooks a crucial element that limits this approach mostly to custom blockchains: regardless of how small the cryptocurrency quantities end up, there is always some monetary value associated with a vote, and, currently, cryptocurrency prices are still too volatile to ignore this problem.

We found no evidence of smart contract usage prior to 2017, which is understandable given that smart contracts were only introduced two years before. The following year saw a rise in this number, but this method is still not a preference for most researchers. This may be due to the same delay observed with the adoption of the Ethereum network, where this network was already online and available, yet most publications during Ethereum's early years were still based on Bitcoin. Solidity, the programming language in which smart contracts for the Ethereum network are written, has evolved significantly since its inception. The lack of early stability in the main programming language used may also contribute to this.

7) Real-World Implementations of Blockchain-Based e-Voting Systems

The flexibility offered by the decentralised approach has produced a number of relevant e-voting applications that are already available for use in a variety of scenarios, unlike the more restrictive centralised paradigm, which limited most of the real-world application to government-sponsored solutions. The following solutions are among the most relevant ones found:

a: Follow My Vote [173]

Follow My Vote is a non-partisan, public-benefit corporation based in the USA that has produced a blockchain-based electronic voting solution with an emphasis on voter mobility. The application provides an information-rich web portal [173] where it is possible to determine that they are implementing their solution in a BitShares blockchain [174], a public blockchain with *smart contract* support. Users interface with this blockchain through a *dApp* named *"Voting Booth"*. The system requires users to create two pairs of Elliptic Curve Cryptography (ECC) keys upon registration. One of the pairs is used to certify the identity of the voters within a trusted third party so that they can vote using their registered key as an identity instead of their own, thus achieving voter anonymity. The second pair is used to sign transactions that write vote data encrypted with the identity key into the blockchain. The two-pair approach is used to increase the security of the system; namely, an adversary needs to control both keys from the same person to be able to impersonate him or her. Votes are public in the sense that they are written on a public blockchain, but anonymous since only voters can decrypt their contents with their private key from the identity pair.

Verifiability is implemented to provide *vote-as-cast* and *counted-as-cast* capabilities. This capability is derived as a feature from their *Voting Booth dApp*, a blockchain explorer tool optimized to retrieve voting records created by the system in the public blockchain.

Follow My Vote also implements multiple vote casting, i.e., does not implement *uniqueness*. A voter can change a submitted vote by replacing it with a new one, with no apparent limitations around the number of times a voter can do this.

To the date of this writing, Follow My Vote is still in its alpha version.

b: TiVi [175]

TiVi was created as a joint venture between *Cybernetica*, an Estonian private company, and *Smartmatic*, a U.S.-based company mostly known for their work with Direct Recording Electronic (DRE) voting machines [176]. *TiVi* is a centralised e-voting solution that only employs blockchain to implement a *public bulletin board*, which is also used for *verifiability* purposes. As far as the documentation reveals, blockchain is only used in this step, which makes this solution more of a hybrid between the two paradigms considered. Although it is indicated that the *public bulletin board* application builds a dedicated blockchain for each voting event, this suggests the use of a framework such as Hyperledger or a proprietary solution. It was not possible to gather further details regarding this aspect.

TiVi approaches *eligibility* from the same angle as many of the decentralised proposals reviewed, i.e., using a centralised database administered by a trusted third party. Voters are authenticated using techniques such as multifactor and biometric authentication.

A vote in TiVi is an encrypted data packet sent by the voter's terminal, digitally signed and protected with end-to-end encryption, to a centralised computer. Shamir's Secret Sharing Scheme [20] is used to split the encryption key used to protect votes into shares, which are then distributed among the members of an electoral board. A dedicated hardware security module (HSM), a popular tool used in e-banking solutions, is used to manage the encryption keys used in this process. Voter *privacy* is achieved through encryption at the voting application level, complimented with a mixing stage using an air-gapped decryption server that suggests the usage of *mix-nets* in this solution. It does seem to use a method similar to the *blind signatures* scheme, but it is referred to as the "double-envelope" scheme instead, used in conjunction with Transport Layer Security (TLS). Along with the *verifiability* implementations indicated, TiVi claims to offer *cast-as-intended*, *stored-as-cast*, and *counted-as-cast* capabilities. The correctness of the mixing and decryption processes used to establish voter *privacy* are verified using *zero-knowledge cryptographic proofs* [177].

The final tabulation of results is computed by rebuilding the decryption key from a minimal set of N shares. This strategy suggests the use of Shamir's secret sharing scheme and the homomorphic properties of the threshold cryptosystem adopted for this solution. To address voter coercion, TiVi also implements multiple vote casting, thus negating *uniqueness*, in which only the last vote is counted and a vote at a physical voting booth on election day precedes any online vote cast previously. TiVi also employs *mobility*, offering a voting platform to users that they could access through a series of mobile devices, similar to the Estonian e-voting system.

TiVi developers seemed to be aware of the perils of the level of centralization in their system, and they addressed the problem of cyberattacks, specifically Distributed-Denial-of-Service (DDoS) attacks, in the documentation. TiVi strategy to combat these attacks is to extend the period in which online voting is operational, defining a "pre-poll" period of 7 to 10 days before the physical election, under the assumption that no adversary can sustain a DDoS attack throughout such a long period [176].

c: Agora [178]

Agora is a Swiss e-voting company in operation since 2015 that proposes a blockchain-centric solution based on a custom, i.e., proprietary, blockchain. The company provides information-rich mediums, such as its website [178] and corresponding white paper [179], which allowed for a more thorough characterization of their proposal.

Similar to TiVi, Agora is explicit in their usage of the blockchain for divulgation purposes, using it also as a *public bulletin board* but with more extended functionalities than other solutions so far. Agora implements individual and universal *verifiability* by storing all voting data encrypted in a consortium blockchain of their own creation, which, suggestively, was named *Bulletin Board*, which is based on the Skipchain architecture [180], a double-linked blockchain, which allows bidirectional navigation, which decreases search times and allows for some operations to require only a part of the blockchain to be downloaded instead of a full copy at the expense of increased implementation complexity and storage volume. This Skipchain implementation uses Practical Byzantine Fault Tolerance as a consensus protocol. Smaller networks are inherently scalable, and as such, Agora is suitable for *large-scale* elections.

Voters can use their encryption keys to validate the state of their vote throughout the election cycle, thus achieving voter *privacy* as well in the process. Vote encryption is achieved using the ElGamal threshold cryptosystem, which also suggests the use of *homomorphic* properties. Ballot anonymization is performed using Neff shuffling [181], a similar scheme to *mix-nets* that arrives at a similar result. As with TiVi, the correctness of the shuffling and decryption processes is assured through *cryptographic proofs*, though in this case there is no clear indication of their specific type.

To compensate for the drawbacks of establishing a blockchain over a smaller network, Agora implements *Cothority*, its own permissioned network for validation purposes, which is composed of politically neutral third-party organizations. These nodes are used to provide consensus and process transactions in Agora's blockchain, which also implements its custom VOTE token for its internal mechanics. VOTE tokens are bought as part of the contractual procedure to run an election and, because they acquire monetary value with this process, are used to compensate honest nodes and thus create incentives for nodes to participate honestly in the consensus protocol.

Agora includes a second layer on top of the Cothority network based on the Catena schema [182], named *Cotena*, for logging purposes. This is a mechanism built on top of the Bitcoin blockchain and uses the OP_RETURN metadata field from Bitcoin transactions to submit logging data, which is a collection of periodic snapshots taken from the bulletin board application.

The main focus of this solution appears to be voter mobility. This is addressed explicitly by the public element of their solution: the *Voteapp* mobile application. This application obfuscates the complex cryptographic operations from the user while providing a simple interface to select options and verify any votes cast with the tool.

d: Voatz [183]

Voatz is also a U.S.-based, venture-backed company from Boston, Massachusetts, dedicated to e-voting systems, with an emphasis on mobility. Voatz created an e-voting solution similar to TiVi's in the sense that it only uses blockchain as a way to implement the *public bulletin board* that was prevalent in centralised solutions. Other than that, Voatz took a conservative approach and designed a system with a higher degree of centralization than other solutions analysed thus far [183].

The centrepiece of the Voatz solution is their vote-casting mobile application. The application protects itself against a series of mobile-based cyberattacks using a Mobile Thread Defence service that is akin to the virus and malware detection software used on PCs. At the network level, Voatz protects communications with HTTPS and end-to-end encryptions, but the details on how this is done reinforce the notion that this is a primarily centralised solution, with multiple references to "smartphone-to-server" and "server-to-smartphone" flows. Voter authentication is inherited from the device where the application runs, namely through PIN codes and/or biometric authentication. Encryption key creation and management are left to the application in order to maintain simplicity of operation. Protection against DDoS attacks follows a standard approach seen in many e-commerce sites, which, ironically, consists of decentralising the main application and setting it to be served by a cluster of cloud computers instead of a single or a small set of servers, along with load-balancing servers to nullify brute force attacks.

Blockchain is only mentioned in this solution regarding the storage of voting data, but in quite detailed fashion. Voatz implements its own custom, permissioned blockchain, which is created using the Hyperledger Fabric framework. The network supporting the aforementioned blockchain derives from the application context, namely trusted entities that can provide computational platforms to work as active nodes, e.g., municipal departments in local elections, county offices in state-wide elections, etc. This suggests the implementation of a consortium blockchain. By storing the votes in a semi-private blockchain with read-only public access, Voatz also increases the transparency and verifiability of its solution with a minimal increment in implementation complexity [184].

8) Conclusion

Decentralised proposals present similarities to their centralised counterparts regarding the security criteria they implement to achieve trust in their systems. Core criteria such as *Privacy*, *Verifiability*, *Eligibility*, *Robustness*, and *Accuracy* saw similar implementation rates as in centralised proposals, as well as the balance between the number of criteria implemented and the election scope intended. *Mobility* in decentralised solutions saw a sharp increase in its implementation rate, whereas *convenience* and *flexibility* became much less popular in contrast, thus answering to *RQA1* and *RQA2*.

Regarding the next set, namely *RQB1* and *RQB2*, the analysis regarding the cryptographic methods implemented in this paradigm revealed similar rates of *homomorphism* usage, but methods such as *blind and ring signatures* and *cryptographic knowledge proofs* saw a clear increase, which is attributed to the usage of a technology that is tendentiously implemented in open access platforms. In opposition, the usage of *mix-nets* was almost non-existent due to the anonymization that a blockchain offers by default.

Regarding the research scope specific to the decentralised paradigm, which was defined via the set of research questions in Section IV-A2, the following conclusions result:

- *RQC1 - Which blockchain type and access control are used in decentralised e-voting solutions?*: A slight majority of proposals, 34 articles, adopted public blockchains as opposed to private ones, which accounted for 25 of the articles of all considered. Consortium blockchains are underrepresented, with only 4 authors preferring this hybrid approach. The type of blockchain used registered a higher variation, but Ethereum registered the most options. Though, as indicated in IV-C5f, these are split into version 1.0, 2.0, and Go-Ethereum implementations. Therefore, it was not possible to identify a blockchain type that stood out from the rest.

- *RQC2 - Are smart contracts used in the decentralised e-voting proposal?*: Slightly more than half of the analysed proposals used smart contracts, or the Hyperledger equivalent, chaincodes in their solution. But in this case, the trend was easily observable, as it was already pointed out in Section IV-C6, with later proposals preferring this approach over earlier ones, when these constructs were either not available or poorly understood. It is safe to assume that smart contracts are expected to be common going forward.
- *RQC3 - Are there any centralising elements in the proposed solution?*: Only 13 of the 63 proposals opted for a fully decentralised approach, with the majority of 51 proposals introducing some sort of centralising element, namely a database, an election authority, or a trusted third party of some kind, usually to address *eligibility* problems in their solutions.
- *RQC4 - What are the methods used in blockchain-based e-voting systems to cast votes?*: We observed some diversity regarding the methods by which authors abstracted votes in their solutions. But the preference seemed to be towards using transactional metadata, either from established cryptocurrencies or custom tokens, in 30 of the proposals and smart contract or chaincode executions, which accounted for 28 of the total sets reviewed. Only five proposals opted to encode votes into a dedicated blockchain block. Fig. 6 contains a statistical representation of these results.

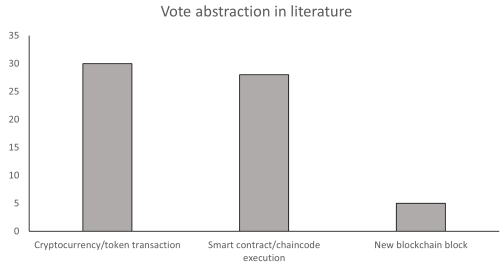

Figure 6. Vote abstraction methods encountered in literature.

The decentralised paradigm reveals greater proximity to the academic proposals reviewed than the centralised one. Similarly, most of the real-world implementations revealed enough technical details to warrant a significant comparison to the academic papers, as well as a characterization under the same set of criteria as these, which can be evidenced from the analysis in Section IV-C7. A shorter gap between academic research and practical societal applications, along with an openness to the underlying technology, justifies this encouraging trend.

V. CONCLUSIONS

This study illustrates how research into modernising voting systems divides into two overlapping eras, with each one triggered by a landmark article that introduced technological principles that allowed new research. Diffie and Hellman's [50] 1976 publication brought computer cryptography to the public realm and with it a steady stream of innovations, not just towards more secure e-voting systems but also benefiting a plethora of industries and applications. And the same seems to be happening again with Satoshi Nakamoto's [2] 2009 publication. The paradigms explored are neither exclusive nor denote a sudden shift from one to the other. Instead, they reflect how e-voting research has evolved through the years, as it is able to successfully incorporate new technological advances to further its own.

Decentralised proposals benefited greatly from the previous, centralised ideas in the sense that these have cleared plenty of cryptographic hurdles that paved the way for the current landscape of decentralised e-voting research. Ideas that were somewhat foreign and unfamiliar decades before, such as encryption methods, homomorphism, and one-way hash functions, are now commonplace within the new era of e-voting solutions, in great part due to the efforts of early researchers.

We have observed a closer relationship between academic proposals and real-world implementations with the decentralised paradigm than with its centralised counterpart, which is an objectively positive trend. Yet, both paradigms are still unable to provide an e-voting solution that could really complement existing voting methods and, perhaps, even replace them in the future. None of the systems indicated in Section IV-B7 is in use today, despite their relative success. All the information obtained thus far relating to the examples indicated in this section points to a discontinuation of the trials, and the lack of more recent examples of centralised e-voting systems being used, either as an official voting method or on an experimental basis, reinforces this notion.

The decentralised solutions indicated in Section IV-C7 have a closer relationship to their academic counterparts, but most of them were found to have never been used, or even ready for, a large-scale election setting. The sole exception is Agora, which claims to have taken part in Sierra Leone's 2018 presidential elections. But a statement issued by the company [185] indicates that the involvement of this company was minimal, namely, that it acted mostly in the capacity of an international observer and that the deployment of their technology in those elections was partial, at best. The main objective of this exercise was to demonstrate the system's capabilities in order to achieve further cooperation opportunities with Sierra Leone's National Election Commission. The remaining solutions considered are still untested in a real-world scenario at the time of this writing.

Though the paradigms are getting increasingly successful in tackling problems of trust that may prevent wide adoption of a truly remote e-voting system, it appears that the efforts undertaken so far are still insufficient. But the technological landscape that we described in this document is still changing. Technological advances such as highly scalable and efficient public blockchains and the Non-Fungible Token Standard are creating promising research avenues. This is particularly relevant for the decentralised approach, where we could identify a chronological progression with the solutions analysed. Early solutions try to adapt a financial tool for alternative purposes, which shows in their lack of applicability. As the technology evolved and researchers had more tools to use, solutions became more elegant and more fitted to the purpose, rather than being limited to adjusting something that was being used for another purpose to a different application. Smart contract usage is a good example of this tendency. Their applicability to e-voting solutions is such that they became almost ubiquitous soon after being introduced.

VI. References

[1] D. Jones. A brief illustrated history of voting. [Online]. Available: https://homepage.cs.uiowa.edu/~jones/voting/pictures/

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *www.bitcoin.org*, pp. 1–9, 2008.

[3] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - a systematic literature review," *Information and Software Technology*, vol. 51, pp. 7–15, 2009.

[4] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, vol. 17, 2008, pp. 33–55.

[5] D. Tranfield, D. Denyer, and P. Smart, "Towards a methodology for developing evidence-informed management knowledge by means of systematic review," *British Journal of Management*, vol. 14, pp. 207–222, 2003.

[6] P. P. Bungale and S. Sridhar, "Electronic voting - a survey," John Hopkins University, Technical Report, 2003.

[7] K. Weldemariam and A. Villfiorita, "A survey: Electronic voting development and trends," in *Proceedings of the 4th International Conference on Electronic Voting 2010, EVOTE 2010*, 1 2010, pp. 119–131.

[8] M. OḾeara, "Survey & analysis of e-voting solutions," Master's thesis, Department of Computer Science, Trinity College, University of Dublin, 2013.

[9] A. AlSammak, A. AbdElRahman, T. ElShishtawy, and A. Elewa, "Challenges of electronic voting - a survey," *Advances in Computer Science: an International Journal*, vol. 4, pp. 98–108, 11 2015.

[10] S. T. Ali and J. Murray, "An overview of end-to-end verifiable voting systems," in *Real-World Electronic Voting: Design, Analysis and Deployment*, F. Hao and P. Y. A. Ryan, Eds. Boca Raton, Florida, USA: CRC Press, 2016, pp. 189–234.

[11] Y. Nasser, C. Okoye, J. Clark, and P. Y. A. Ryan, "Blockchain and voting: Somwhere between hype and a panacea," Concordia University Irvine, White Paper, 2018. [Online]. Available: https://users.encs.concordia.ca/~clark/papers/draft_voting.pdf

[12] S. Kadam, K. Chavan, I. Kulkarni, and A. Patil, "Survey on digital e-voting system by using blockchain technology," *International Journal of Advance Scietific Research and Engineering Trends*, vol. 4, pp. 5–8, 2019.

[13] S. F. Sayyad, M. Pawar, A. Patil, V. Pathare, and P. Poduval, "Features of blockchain voting: A survey," *International Journal for Innovative Research in Science & Technology*, vol. 5, pp. 12–14, 2 2019.

[14] Y. Abuidris, R. Kumar, and W. Wenyong, "A survey of blockchain based on e-voting systems," in *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, 12 2019, pp. 99–104.

[15] R. Taş and Özgür, "A systematic review of challenges and opportunities of blockchain for e-voting," *Symmetry*, vol. 12, pp. 1–24, 2020.

[16] R. Fatih, S. Arezki, and T. Gadi, "A survey on e-voting based on blockchain," in *Proceedings of the 4th International Conference on Networking, Information Systems & Security*, 2021, pp. 1–8.

[17] E.-V. S. using Hyperledger Sawtooth, "Vivek s. k. and yashank r. s. and yashas prashanth and yashas n." in *Proceedings of the 2020 International Conference on Advances in Computing, Communication*

and Materials (ICACCM 2020), 2020, pp. 29–35.

[18] U. C. Çabuk, E. Adıgüzel, and E. Karaarslan, "A survey on feasibility and suitability of blockchain techniques for the e-voting systems," *International Journal of Advanced Research in Computer and Communication Engineering*, pp. 124–134, 3 2018.

[19] U. Jafar and M. J. A. Aziz, "A state of the art survey and research directions on blockchain based electronic voting system," in *Advances in Cyber Security (ACeS 2020) - Communications in Computer and Information Science*, vol. 1347, 2021, pp. 248–266.

[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612–613, 1979.

[21] J. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret," *Advances in Cryptology - CRYPTO '86*, pp. 251–260, 1986.

[22] Z. Rjašková, "Electronic voting schemes," Master's thesis, Department of Computer Science, Faculty of Mathematics, Physics and Informatics, Comenius University, 2002.

[23] J. Benaloh, "Verifiable secret-ballot elections," Ph.D. dissertation, Yale University, 1987.

[24] A. Juels, D. Catalano, and M. Jackobsson, "Coercion-resistant electronic elections," *Lecture Notes in Computer Science*, vol. 6000 LNCS, pp. 37–63, 2010.

[25] D. Chaum, "Blind signatures for untraceable payments," *Advances in Cryptology*, pp. 199–205, 1983.

[26] B. Schneider, *Applied Cryptography*, 2nd ed. John Wiley & Sons, 1994.

[27] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, "Quadratic span programs and succinct nizks without pcps," *Lecture Notes in Computer Science*, vol. 7881 LNCS, pp. 626–645, 2013.

[28] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," *Communications of the ACM*, vol. 69, pp. 103–112, 2016.

[29] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," in *Proceedings of the 23rd USENIX Security Symposium*, 2014, pp. 781–796.

[30] J. Eberhardt and S. Tai, "Zokrates - scalable privacy-preserving off-chain computations," in *2018 IEEE International Conference on Internet of Things (iThings), Green Computing and Communications (GreenCom), Cyber, Physical and Social Computing (CPSCom) and Smart Data (SmartData)*, 2018, pp. 1084–1091.

[31] D. D. F. Maesa, A. Lisi, P. Mori, L. Ricci, and G. Boschi, "Self sovereign and blockchain based access control: Supporting attributes privacy with zero knowledge," *Journal of Network and Computer Applications*, vol. 212, pp. 1–19, 2023.

[32] C. Park, K. Itoh, and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme," *Lecture Notes in Computer Science*, vol. 765 LNCS, pp. 248–259, 1994.

[33] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, 2016.

[34] S. Bouraga, "A taxonomy of blockchain consensus protocols: A survey and classification framework," *Expert Systems with Applications*, vol. 168, 2021.

[35] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Express*, vol. 6, pp. 93–97, 2020.

[36] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys and Tutorials*, vol. 22, pp. 1432–1465, 2020.

[37] V. Patel, “A framework for secure and decentralized sharing of medical imaging data via blockchain consensus,” *Health Informatics Journal*, vol. 25, pp. 1398–1411, 2019.

[38] M. Bowman, D. Das, A. Mandal, and H. Montgomery, *Progress in Cryptology - INDOCRYPT 2021*. Springer International Publishing, 2021, vol. 13143, pp. 559–583.

[39] M. Milutinovic, W. He, H. Wu, and M. Kanwal, “Proof of luck: an efficient blockchain consensus protocol,” in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, 2016, pp. 1–7.

[40] S. Joshi, “Feasibility of proof of authority as a consensus protocol model,” *arXiv: 2109.02480v1*, 2021. [Online]. Available: http://arxiv.org/abs/2109.02480

[41] M. Bartoletti and L. Pompianu, “An analysis of bitcoin op_return metadata,” in *Lecture Notes in Computer Science*, vol. 9603, 2017, pp. 218–230.

[42] S. Suratkar, M. Shirole, and S. Bhirud, “Cryptocurrency wallet: A review,” in *Proceedings of the 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2020, pp. 1–7.

[43] [Online]. Available: https://www.blockchain.com/explorer

[44] [Online]. Available: https://blockchair.com/

[45] [Online]. Available: https://bscscan.com/

[46] N. Szabo. (1997) Formalizing and securing relationships on public networks. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/download/548/469

[47] W. Zou, D. Lo, P. S. Kochhar, X.-B. D. Le, X. Xia, Y. Feng, Z. Chen, and B. Xu, “Smart contract development: Challenges and opportunities,” in *IEEE Transactions on Software Engineering*, vol. 47, 2021, pp. 2084–2106.

[48] C. Dannen, *Introducing Ethereum and Solidity*. Apress, 2016.

[49] A. M. Antonopoulos and G. Wood, *Mastering Ethereum, Building Smart Contracts and Dapps*, 1st ed. OŔeilly, 2018.

[50] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, 11 1976.

[51] U. M. Maurer and S. Wolf, “The diffie-hellman protocol,” *Designs, Codes and Cryptography*, vol. 19, pp. 147–171, 2000.

[52] W. Diffie, “The first ten years of public key cryptography,” in *Proceedings of the IEEE*, vol. 76, 1988, pp. 560–577.

[53] M. Ahmed, B. Sanjabi, D. Aldiaz, A. Rezaei, and H. Omotunde, “Diffie-hellman and its application in security protocols,” *International Journal of Engineering Science and Innovative Technology (IJESIT)*, vol. 1, pp. 69–73, 2012.

[54] P. G. Neumann, “Security criteria for electronic voting,” in *Proceedings of the 16th National Computer Security Conference*, 1993, pp. 1–7.

[55] J. D. Cohen and M. J. Fischer, “A robust and verifiable cryptographically secure election scheme,” in *26th Symposium on Foundations of Computer Science*, 07 1985, pp. 1–12.

[56] J. Benaloh and M. Young, “Distributing the power of a government to enhance the privacy of voters,” in *Proceedings of the 5th ACM Symposium on the Principles of Distributed Computing*, 1986, pp. 52–62.

[57] A. Fujioka, T. Okamoto, and K. Ohta, “A practical secret voting scheme for large scale elections,” in *Workshop on the Theory and Application of Cryptographic Techniques*, J. Seberry and Y. Zheng, Eds. Springer-Verlag, 1992, pp. 244–251.

[58] B. Schneider, “Voting security and tech-

nology," *IEEE Security and Privacy*, vol. 2, pp. 84–84, 2004.

[59] M. I. Shamos. (1993) Electronic voting - evaluating the threat. [Online]. Available: http://wheresthepaper.org/Shamos1993ElectronicVotingEvaluatingTheThreat.htm

[60] B. Lee and K. Kim, "Receipt-free electronic voting through the collaboration of voter and honest verifier," in *Proceedings of JW-ISC 2000*, 2000, pp. 1–8.

[61] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," *Lecture Notes in Computer Science - EUROCRYPT 2000*, pp. 539–556, 2000.

[62] J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections," in *26th Annual ACM Symposium on Theory of Computing*, 1994, pp. 544–553.

[63] K. Sako and J. Kilian, "Secure voting using partially compatible homomorphisms," *Advances in Cryptology - CRYPTO '94*, pp. 411–424, 1994.

[64] J. Schweisgut, "Coersion-resistant electronic elections with observer," in *Electronic Voting 2006*, 2006, pp. 171–177.

[65] S. Ibrahim, M. Kamat, M. Salleh, and S. R. A. Aziz, "Secure e-voting with blind signature," in *NCTT 2003 Proceedings - 4th National Conference on Telecommunication Technology*, 2003, pp. 194–197.

[66] M. Chaieb, M. Koscina, S. Yousfi, P. Lafourcade, and R. Robbana, "Dabsters: a privacy preserving e-voting protocol for permissioned blockchain," *Leture Notes in Computer Science*, vol. 11884 LNCS, pp. 292–312, 2019.

[67] C. K. Adiputra, R. Hjort, and H. Sato, "A proposal of blockchain-based electronic voting system," in *Proceedings of the 2nd World Conference on Smart Trends in Systems, Security and Sustainability*, 2019, pp. 185–192.

[68] M. Chaieb, S. Yousfi, P. Lafourcade, and R. Robbana, "Verify-your-vote: A verifiable blockchain-based online voting protocol," in *Lecture Notes in Business Information Processing*, 2018, pp. 16–30.

[69] X. Yang, X. Y. adn Surya Nepal, A. Kelarev, and R. Han, "Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities," *Future Generation Computer Systems*, vol. 112, pp. 859–874, 2020.

[70] K. Lee, J. I. James, and T. G. Ejeta, "Electronic voting service using block-chain," *Journal of Digital Forensics, Security and Law*, vol. 11, pp. 123–136, 2017.

[71] W.-S. Juang and C.-L. Lei, "A secure and practical electronic voting scheme for real world environments," *IEICE Transactions Fundamentals*, 1997.

[72] ——, "A collision-free secret ballot protocol for computerized general elections," *Computers and Security*, vol. 15, pp. 339–348, 1996.

[73] W.-S. Juang, C.-L. Lei, and H.-T. Liaw, "A verifiable multi-authority secret election allowing abstention from voting," *The Computer Journal*, vol. 45, pp. 672–682, 2002.

[74] L. F. Cranor and R. K. Cytron, "Sensus: A security-conscious electronic polling system for the internet," *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, pp. 561–570, 1997.

[75] R. Joaquim, A. Zúquete, and P. Ferreira, "Revs - a rocust electronic voting system," *IADIS International Journal of www/Internet*, vol. 1, pp. 47–63, 2003.

[76] O. Çetinkaya and A. Doğanaksoy, "Electronic voting protocol based on blind signatures," Middle East Technical University, Technical Report, 2005. [Online]. Available: https://user.ceng.metu.edu.tr/~corhan/Papers/ulusal05.pdf

[77] T. Wu, "An e-voting system based on blockchain and ring signature," Master's thesis, University of Birmingham, 2017.

[78] A. B. Ayed, "A conceptual secure blockchain-based electronic voting system," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 9, pp. 1–9, 05 2017.

[79] S. Bistarelli, M. Mantilacci, P. Santancini, and F. Santancini, "An end-to-end voting-system based on bitcoin," in *Proceedings of the 32nd ACM SIGAPP Symposium on Applied Computing*, 2017, pp. 1836–1841.

[80] K. Sadia, M. Masuduzzaman, R. K. Paul, and A. Islam, "Blockchain-based secure e-voting with the assistance of smart contract," in *IC-BCT*, 2020, pp. 161–176.

[81] A. M. Larriba, A. C. I. Cucó, J. M. Sempere, and D. López, "Distributed trust, a blockchain election scheme," *Informatica (Netherlands)*, vol. 32, pp. 321–355, 2021.

[82] K. Sako and J. Kilian, "Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth," *Advances in Cryptology - EUROCRYPT '95*, 1998.

[83] R. Araújo, N. B. Rajeb, R. Robbana, J. Traoré, and S. Yousfi, "Towards practical and secure coercion-resistant electronic elections," in *Proceedings of the 9th International Conference on Cryptography and Network Security*, 2010, pp. 278–297.

[84] P. Locher and R. Haenni, "Receipt-free remote electronic elections with everlasting privacy," *Annals of Telecommunications*, vol. 71, pp. 323–336, 2016.

[85] B. Yu, J. Liu, A. Sakzad, S. Nepal, R. Steinfeld, P. Rimba, and M. H. Au, "Platform-independent secure blockchain-based voting system," in *Proceedings of the 2018 International Conference on Information Security*, 2018, pp. 369–386.

[86] J. Lyu, Z. L. Jiang, X. Wang, Z. Nong, M. H. Au, and J. Fang, "A secure decentralized trustless e-voting system based on smart contract," *Proceedings of the 2019 18th IEEE International Conference On Trust, Security and Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science and Engineering*, pp. 570–570, 2019.

[87] N. Faour, "Transparent e-voting dapp based on waves blockchain and ride language," in *Proceedings of the XVI International Symposium on Problems of Redundancy in Information and Control Systems (Redundancy 2019)*, 2019, pp. 219–223.

[88] J. Mols and E. Vasilomanolakis, "ethvote: Towards secure voting with distributed ledgers," in *International Conference on Cyber Security and Protection of Digital Services and Cyber Security 2020*, 2020.

[89] W.-C. Ku and S.-D. Wang, "A secure and practical electric voting scheme," *Computer Communications*, vol. 22, pp. 279–286, 1999.

[90] M. Herschenberg, "Secure electronic voting over the world wide web," Master's thesis, Massachusetts Institute of Technology, 1997.

[91] J. P. Cruz and Y. Kaji, "E-voting system based on the bitcoin protocol and blind signatures," *IPSJ Transactions on Mathematical Modeling and Its Applications*, vol. 2016-MPS-107, 2016.

[92] M. Chaieb and S. Yousfi, "Loki vote: A blockchain-based coercion resistant e-voting protocol," *Lecture Notes in Business Information Processing*, vol. 402, pp. 151–168, 2020.

[93] T. Dimitriou, "Efficient, coercion-free and universally verifiable blockchain-based voting," *Computer Networks*, vol. 174, 2020.

[94] Y. Takabatake, D. Kotani, and Y. Okabe, "An anonymous distributed electronic voting system using zerocoin," in *International Conference on Information*

Networking, 2021, pp. 163–168.

[95] J.-H. Hsiao, R. Tso, C.-M. Chen, and M.-E. Wu, “Decentralized e-voting systems based on the blockchain technology,” in *Lecture Notes in Electrical Engineering*, vol. 474, 2018, pp. 305–309.

[96] A. Baraani-Dastjerdi, J. Pieprzyk, and R. Safavi-Naini, “A practical electronic voting protocol using threshold schemes,” in *Proceedings of the 11th Annual Computer Security Applications Conference*. IEEE Computer Security Press, 1995.

[97] F. S. Hardwick, A. Gioulis, R. N. Akram, and K. Markantonakis, “E-voting with blockchain: An e-voting protocol with decentralization and voter privacy,” in *2018 IEEE International Conference on Internet Things*, 2018, pp. 1561–1567.

[98] K. M. Kahn, J. Arshad, and M. M. Khan, “Secure digital voting system based on blockchain technology,” *International Journal of Electronic Government Research*, vol. 14, 2018.

[99] W. Zhang, S. Huang, Y. Yuan, Y. Hu, S. Huang, S. Cao, and A. Chopra, “A privacy-preserving voting protocol on blockchain,” in *2018 IEEE 11th International Conference on Cloud Computing*, 2018, pp. 401–408.

[100] S. Bartolucci, P. Bernat, and D. Joseph, “Sharvot: secret share-based voting on the blockchain,” in *WETSEB’18: IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, 2018, pp. 1–5.

[101] L. Vo-Cao-Thuy, K. Cao-Minh, C. Dang-Le-Bao, and T. A. Nguyen, “Votereum: An ethereum-based e-voting system,” in *2019 IEEE-RIVF International Conference on Computing and Communication Technologies*, 2019.

[102] E. Zaghloul, T. Li, and J. Ren, “Anonymous and coercion-resistant distributed electronic voting,” in *Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium*, 2020, pp. 389–393.

[103] G. Han, Y. Li, Y. Yu, K.-K. R. Choo, and N. Guizani, “Blockchain-based self-tallying voting system with software updates in decentralized iot,” in *IEEE Network*, vol. 34, 2020, pp. 166–172.

[104] Y. Zhou, Y. Liu, C. Jiang, and S. Wang, “An improved foo voting scheme using blockchain,” *International Journal of Information Security*, vol. 19, pp. 303–310, 2020.

[105] H. Hu, J. Ou, B. Qian, Y. Luo, P. He, M. Zhou, and Z. Chen, “A practical anonymous voting scheme based on blockchain for internet of energy,” *Security and Communication Network*, vol. 2022, pp. 1–15, 2022.

[106] Y. Li, W. Susilo, G. Yang, Y. Yu, D. Liu, X. Du, and M. Guizani, “A blockchain-based self-tallying voting protocol in decentralized iot,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, pp. 119–130, 2022.

[107] H. Li, Y. Li, Y. Yu, and B. Wang, “A blockchain-based traceable self-tallying e-voting protocol in ai era,” *IEEE Transactions on Network Science and Engineering*, vol. 8, pp. 1019–1032, 2021.

[108] Z. Xu and S. Cao, “Efficient privacy-preserving electronic voting scheme based on blockchain,” in *Proceedings of the 2020 IEEE International Conference on Smart Internet of Things, SmartIoT 2020*, 2020, pp. 190–196.

[109] R. Cramer, R. Gennaro, and B. Schoenmakers, “A secure and optimally efficient multi-authority election scheme,” *Advances in Cryptology - EUROCRYPT ’97*, vol. 8, pp. 103–118, 1997.

[110] R. Cramer, M. Franklin, B. Schoen-

makers, and M. Yung, “Multi-authority secret-ballot elections with linear work,” in *EUROCRYPT ’96 Conference Proceedings*, vol. 1070, 1996, pp. 72–83.

[111] Z. Zhao and T.-H. H. Chan, “How to vote privately using bitcoin,” in *Lecture Notes in Computer Science*, vol. 9543, 2016, pp. 82–96.

[112] Y. Liu and Q. Wang, “An e-voting protocol based on blockchain,” *IACR Cryptology ePrint Archive*, 2017.

[113] J. Lopes, J. L. Pereira, and J. ao Varajão, “Blockchain based e-voting system: A proposal,” in *Proceedings of the 25th Americas Conference on Information Systems, AMCIS 2019*, 2019.

[114] C. Killer, B. Rodrigues, E. J. Scheid, M. Franco, M. Eck, N. Zaugg, A. Scheitlin, and B. Stiller, “Provotum: A blockchain-based and end-to-end verifiable remote electronic voting system,” in *Proceedings of the 2020 IEEE Conference on Local Computer Networks (LNC)*, vol. November, 2020, pp. 172–183.

[115] S. Zhang, L. Wang, and H. Xiong, “Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability,” *International Journal of Information Security*, vol. 19, pp. 323–341, 2020.

[116] S. T. Alvi, M. N. Uddin, and L. Islam, “Digital voting: A blockchain-based e-voting system using biohash and smart contract,” in *Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020)*, 2020, pp. 228–233.

[117] G. Verma, “A secure framework for e-voting using blockchain,” *IEEE Access*, pp. 1–5, 2022.

[118] S. T. Alvi, M. N. Uddin, L. Islam, and S. Ahamed, “Dvtchain: A blockchain-based decentralized mechanism,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, pp. 6855–6871, 2022.

[119] M. J. Radwin, “An untraceable, universally verifiable voting scheme,” in *Seminar in Cryptology*, 1995, pp. 1–9.

[120] P. V. C. L. Faria, “Remote electronic voting: Studying and improving helios,” Master’s thesis, University of Minho, Portugal, 2012.

[121] T. P. Pedersen, “A threshold cryptosystem without a trusted party,” *Lecture Notes in Computer Science*, vol. 547 LNCS, pp. 522–526, 1991.

[122] D. Chaum, “The dining cryptographers problem: Unconditional sender and recipient untraceability,” *Journal of Cryptology*, vol. 1, pp. 65–75, 1988.

[123] T. ElGamal, “A public key cryptosystem and signature scheme based on discrete logarithms,” *Advances in Cryptology - CRYPTO’84, LNCS*, vol. 196, pp. 10–18, 1985.

[124] C. Boyd, “A new multiple key cipher and an improved voting scheme,” in *EUROCRYPT ’89 Proceedings - Advances in Cryptology*. Springer-Verlag, 1990, pp. 617–625.

[125] H. Nurmi, A. Salomaa, and L. Santean, “Secret ballot elections in computer networks,” *Computer and Secutity*, vol. 10, pp. 553–560, 1991.

[126] K. R. Iversen, “A cryptographic scheme for computerized general elections,” *Advances in Cryptology*, pp. 405–419, 1992.

[127] V. Niemi and A. Renvall, “How to prevent buying of votes in computer elections,” *Advances in Cryptology - ASIACRYPT ’94*, vol. 917, pp. 164–170, 1995.

[128] T. Okamoto, “An electronic voting scheme,” *Advanced IT Tools*, pp. 21–30, 1996.

[129] ——, “Receipt-free electronic voting schemes for large scale elections,” *Lecture Notes in Computer Science*, vol. 1361, pp. 25–35, 1998.

[130] V. Niemi and A. Renvall, “Efficient voting with no selling of votes,” *Theoretical omputer Science*, vol. 226, pp. 105–116, 1999.

[131] E. Magkos, M. Burmester, and V. Chrissikopoulos, “Receipt-freeness in large-scale elections without untappable channels,” *IFIP Advances in Information and Communication Technology*, vol. 74, pp. 683–694, 2001.

[132] T. Moran and M. Naor, “Receipt-free universally-verifiable voting with everlasting privacy,” *Lecture Notes in Computer Science*, vol. 4117 LNCS, pp. 373–392, 2006.

[133] D. Chaum, J. V. D. Graaf, P. Y. A. Ryan, and P. L. Vora, “Secret ballot elections with unconditional integrity,” *Cryptology ePrint Archive*, pp. 1–33, 2007.

[134] B. Adida, “Helios: Web-based open-audit voting,” *USENIX Security Symposium*, pp. 335–348, 2008.

[135] C. Hébant, D. H. Phan, and D. Pointcheval, *Linearly-Homomorphic Signatures and Scalable Mix-Nets*. Springer, Cham, 2020, vol. 12111 LNCS, pp. 597–627.

[136] V. Sidorov, E. W. Y. Fan, and W. K. Ng, “Comprehensive performance analysis of homomorphic cryptosystems for practical data processing,” *arXiv:2202.02960v1*, 2022. [Online]. Available: http://arxiv.org/abs/2202.02960

[137] N. B. Binder, R. Krimmer, G. Wenda, and D.-H. Fischer, “International standards and ict projects in public administration: Introducing electronic voting in norway, estonia and switzerland compared,” *The Estonian Journal of Administrative Culture and Digital Governance*, vol. 19, pp. 8–22, 2019.

[138] N. Braun and D. Brändli, “Swiss e-voting pilot projects: Evaluation, situation analysis and how to proceed,” in *Proceedings of the 2nd International Workshop on Electronic Voting*, R. Krimmer, Ed., 2006, pp. 27–36.

[139] T. Pinault and P. Courtade, “E-voting at expatriates’ mps elections in france,” *Electronic Voting*, pp. 289–195, 02 2012.

[140] N. J. Goodman, *Internet Voting in a Local Election in Canada*. Springer, Cham, 2014, vol. 31, pp. 7–24.

[141] S. Heiberg, A. Parsovs, and J. Willemson, “Log analysis of estonia internet voting 2013-2015,” *Lecture Notes in Computer Science*, pp. 19–34, 2015.

[142] Ülle Madise and T. Martens, “E-voting in estonia 2005: The first practice of country-wide binding internet voting in the world,” in *Proceedings of the 2nd International Workshop on Electronic Voting*, R. Krimmer, Ed., 2006, pp. 15–26.

[143] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcal, H. Hursti, M. MacAlpine, and J. A. Halderman, “Security analysis of the estonian internet voting system,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communication Security*, 05 2014, pp. 703–715.

[144] J. B. i Esteve, B. Goldsmith, and J. Turner, “International experience with e-voting: Norwegian e-vote project,” The International Foundation for Electoral Systems, Washington, U.S.A, Tech. Rep., 06 2012.

[145] J. A. Halderman and V. Teague, “The new south wales ivote system: Security failures and verification flaws in a live online election,” *Lecture Notes in Computer Science*, pp. 35–53, 03 2015.

[146] A. Barnes, C. Brake, and T. Perry, “Digital voting with the use of blockchain technology,” Plymouth University, Tech. Rep., 2016.

[147] K. Kirby, A. Masi, and F. Maymi, “Votebook, a proposal for a blockchain-based electronic voting system,” New York University, Tech. Rep., 09 2016.

[148] P. McCorry, S. F. Shahandashti, and

F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *Lecture Notes in Computer Science*, vol. 10322 LNCS, 2017, pp. 357–375.

[149] S. H. Shaheen, M. Yousaf, and M. Jalil, "Temper proof data distribution for universal verifiability and accuracy in electoral process using blockchain," in *13th International Conference on Emerging Technologies*, 2017.

[150] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu, "Large-scale election based on blockchain," *Procedia Computer Science*, vol. 129, pp. 234–237, 2018.

[151] A. K. Koç, E. Yavuz, U. C. Çabuk, and G. Dalkılıç, "Towards secure e-voting using ethereum blockchain," in *6th Symposium on Digital Foresinc and Security*, 2018.

[152] W.-J. Lai, Y.-C. Hsieh, C.-W. Hsueh, and J.-L. Wu, "Date: A decentralized, anonymous, and transparent e-voting system," in *Proceedings of the 2018 1st IEEE International Conference in Hot Information-Centric Networking*, 2018.

[153] G. G. Dagher, P. B. Marella, M. Milojkovic, and J. Mohler, "Broncovote: Secure voting system using ethereum's blockchain," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 96–107.

[154] F. þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa, and G. Hjálmtýsson, "Blockchain-based e-voting system," in *IEEE International Conference on Cloud Computing*, 7 2018, pp. 983–986.

[155] S. Shukla, S. D. O., S. A. N., and M. H. R., "Online voting application using ethereum blockchain," in *2018 International Conference on Advances in Computing, Communications and Informatics*, 2018, pp. 873–880.

[156] D. Khoury, E. F. Kfoury, A. Kassem, and H. Harb, "Decentralized voting platform based on ethereum blockchain," in *Proceedings of the 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, 2018, pp. 1–6.

[157] A. J. Perez and E. N. Ceesay, "Improving end-to-end verifiable voting systems with blockchain technologies," in *Proceedings of the 2018 IEEE Conference on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics*, 2018, pp. 1108–1115.

[158] R. Matile, B. Rodrigues, E. Scheid, and B. Stiller, "Caiv: Cast-as-intended verifiability in blockchain-based voting," in *2019 IEEE International Conference on Blockchain and Cryptocurrency*, 2019, pp. 24–28.

[159] R. Bosri, A. R. Uzzal, A. A. Omar, A. S. M. T. Hasan, and M. Z. A. Bhuiyan, "Towards a privacy-preserving voting system through blockchain technologies," in *Proceedings of the IEEE 17th International Conference on Dependable, Autonomic and Secure Computing*, 2019, pp. 602–608.

[160] H. Yi, "Securing e-voting based on blockchain in p2p network," *Eurasip Journal on Wireless Communications and Networking*, 2019.

[161] A. Singh and K. Chatterjee, "Secevs: Secure electronic voting system using blockchain technology," in *2018 International Conference on Computing, Power and Communication Technologies*, 2019, pp. 863–867.

[162] M. H. Murtaza, Z. A. Alizai, and Z. Iqbal, "Blockchain based anonymous voting system using zksnarks," in *Proceeding of the 2019 International Conference on Applied and Engineering Mathematics*, 2019, pp. 209–2014.

[163] D. Seftyanto, A. Amiruddin, and A. R. Hakim, "Design of blockchain-based electronic election system using hyperledger: Case of indonesia," in *Proceedings of the 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2019*, vol. 6, 2019, pp. 228–233.

[164] Y. Zhang, Y. Li, L. Fang, P. Chen, and X. Dong, "Privacy-protected electronic voting system based on blockchain and trusted execution environment," in *Proceedings of the 2019 IEEE 5th International Conference on Computer and Communications*, 2019, pp. 1252–1257.

[165] W. Shao, C. Jia, Y. Xu, K. Qiu, Y. Gao, and Y. He, "Attrichain: Decentralized traceable anonymous identities in privacy-preserving permissioned blockchain," *Computers and Security*, vol. 99, 2020.

[166] A. Mani, S. Patil, S. Sheth, and L. S. Kondaka, "College election system using blockchain," *ITM Web of Conference*, vol. 44, pp. 1–5, 2022.

[167] C. A. ul Hassan, M. Hammad, J. Iqbal, S. Hussain, S. S. Ullah, H. AlSalman, M. A. A. Moslch, and M. Arif, "A liquid democracy enabled blockchain-based electronic voting system," *Scientific Programming*, vol. 2022, 2022.

[168] S. Vidwans, A. Deshpande, P. Thakur, A. Verma, and S. Palwe, "Permissioned blockchain voting system using hyperledger fabric," in *Proceedings of the 2022 International Conference on IoT and Blockchain Technology, ICIBT 2022*, 2022, pp. 1–6.

[169] M. O'Keeffe, "The paillier cryptosystem, a look into the cryptosystem and its potential applications," The College of New Jersey, Technical report, 2008.

[170] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *Proceedings of the 2017 IEEE International Conference on Software Architecture*, 2017, pp. 243–252.

[171] W. Viriyasitavat and D. Hoonsopon, "Blockchain characteristics and consensus in modern business processes," *Journal of Industrial Information Integration*, vol. 13, pp. 32–39, 2019.

[172] Y. Xinyi, Z. Yi, and Y. He, "Technical characteristics and model of blockchain," in *Proceedings of the 10th International Conference on Communication Software and Networks, ICCSN 2018*, 2018, pp. 562–566.

[173] A. K. Ernest. (2021) Follow my vote. [Online]. Available: https://followmyvote.com

[174] BitShares.com. (2021) Bitshares whitepaper. [Online]. Available: https://docs.bitshares.build/docs/get-started/bitshares-whitepaper/

[175] tivi.io. (2021) Tivi - our story. [Online]. Available: https://tivi.io/

[176] Smartmatic and Cybernetica. (2021) Tivi - remote voting solutions factsheet. [Online]. Available: https://tivi.io/content/pdfs/Factsheet_TIVI-46baa9a2df.pdf

[177] ——. (2021) Tivi - remote voting solutions white paper. [Online]. Available: https://tivi.io/content/pdfs/Whitepaper_Online_Voting_Challenge_Considerations_TIVI-0fd7d7e8e4.pdf

[178] Agora. (2021) Agora - bringing voting systems into the digital age. [Online]. Available: https://www.agora.vote/

[179] ——, "Agora, bringing our voting systems into the 21st century," Agora Voting Systems, Tech. Rep., 2021. [Online]. Available: https://static1.squarespace.com/static/5b0be2f4e2ccd12e7e8a9be9/t/5f37eed8cedac41642edb534/1597501378925/Agora_Whitepaper.pdf

[180] K. Nikitin, E. Kokoris-Kogias,

P. Jovanovic, N. Gailly, and L. Gasser, “Chainiac: Proactive software-update transparency via collectively signed skipchains and verified builds,” in *Proceedings of the 26th USENIX Security Symposium*, 2017, pp. 1271–1287.

[181] C. A. Neff, “A verifiable secret shuffle and its application to e-voting,” in *Proceedings of the ACM Conference on Computer and Communication Security*, 2001, pp. 116–125.

[182] A. Tomescu and S. Devadas, “Catena: Efficient non-equivocation via bitcoin,” in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, 2017, pp. 393–409.

[183] Voatz. (2021) Voatz - secure, accessible voting at your fingertips. [Online]. Available: https://voatz.com/

[184] L. Moore, “Voatz mobile voting platform whitepaper,” Voatz, Tech. Rep., 2019. [Online]. Available: https://voatz.com/wp-content/uploads/2020/07/voatz-security-whitepaper.pdf

[185] Agora. (2018) Agora official statement regarding sierra leone election. [Online]. Available: https://medium.com/agorablockchain/agora-official-statement-regarding-sierra-leone-election-7730d2d9de4e#id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IjE4MmU0NTBhMzVhMjA4MWZhYTFkOWFlMWQyZDc1YTBmMjNkOTFkZjgiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOjE2NDM5MjEyMzgsImF1ZCI6IjIxNjI5NjAzNTgzNC1rMWs2cWUwNjBzMnRwMmEyamFtNGxqZGNtczAwc3R0Zy5hcHBzLmdvb2dsZXVzZXJjb250ZW50LmNvbSIsInN1YiI6IjEwNjQ5NTI4MDU0MTYxMDUwMzIzMSIsImVtYWlsIjoicmRsYWxtZWlkYUBnbWFpbC5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwiYXpwIjoiMjE2Mjk2MDM1ODM0LWsxazZxZTA2MHMydHAyYTJqYW00bGpkY21zMDBzdHRnLmFwcHMuZ29vZ2xldXNlcmNvbnRlbnQuY29tIiwibmFtZSI6IlJpY2FyZG8gQWxtZWlkYSIsInBpY3R1cmUiOiJodHRwczovL2xoMy5nb29nbGV1c2VyY29udGVudC5jb20vYS0vQU9oMTRHamIyMFVxM2NPMG5PU3BiVmNKRjJrYmRGZVU5NmxIejRIbFotd3Bndz1zOTYtYyIsImdpdmVuX25hbWUiOiJSaWNhcmRvIiwiZmFtaWx5X25hbWUiOiJBbG1laWRhIiwiaWF0IjoxNjQzOTIxNTM4LCJleHAiOjE2NDM5MjUxMzgsImp0aSI6ImFjNDZjYjZjOGIwMmEyNWY5ZDc4ZmQ4MmM1OWE0YjEyMDU2NzRjOWYifQ.X9Sphl8-tqUEXYO_rSh6NC1rH2dF-WlTKfqtcuOn-fho196vWw027Dbb4ZfVZSiZduKGFtfEuTkjI_ofOWdNwNVeOnUkd9Z5t0-7sRMsZke5IJHnPr52H-Qr37-ckgLFyS3GZg5MkkhfmuvDy136L3szFux-CVSs4iahU5RakJA1OZEIHV6Y9fqhN1v8sHj5MOBMEUX69qXcXBVLHM16nSfRPF4gGuQIC8QUYULcvDrAxVMpHdEt4EJIwE0pn5XXsBPCAGczTpCl6SD7KUpekryM4qmT_a7xMuN3S21e8rMgwDZ_i3TUC3TUl86-WVRjiFHimmIV_c1d0ruRJ9TXwQ

RICARDO LOPES ALMEIDA is a doctoral student in the first edition of the Italian Doctoral Programme in Blockchain and Distributed Ledger Technology - Social Systems and Smart Societies at the Univesità di Pisa and Università di Camerino. He obtained an Integrated Master’s in Electronic Engineering and Telecommunications from the University of Aveiro, Portugal, in 2012 and a graduation in Physics and Chemistry from the University of Évora, Portugal, in 2005.

He has split his career almost evenly between academia and industry. He has worked as a science teacher in public schools and as a private tutor. After acquiring a qualification as an engineer, he has worked mainly as a consulting software developer and telecommunications engineer.

FABRIZIO BAIARDI graduated in computer science at Universitá di Pisa where he is currently a full professor in computer science.

His main research interest is cyber risk assessment and management. He is a cofounder of Haruspex, a startup that develops risk assessment and management tools based upon adversary emulation. He holds some patents on intrusion detection.

DAMIANO DI FRANCESCO MAESA is an assistant professor in computer science at the University of Pisa, Italy. His past affiliations include the University of Cambridge, King's College London, and the National Research Institute of Italy.

He has a PhD in Computer Science from the University of Pisa and specialises in blockchain and Distributed Ledger Technologies (DLT). He has been involved in the topic since 2013, and, beside his academic contributions, he has held several guest lectures, seminars, and workshops to spread awareness about blockchain technology.

LAURA RICCI received the M.Sc. in Computer Science from the University of Pisa in 1983 and the Ph.D. from the same university, in 1990. She is currently a full professor at the Department of Computer Science, University of Pisa, Italy.

Her research interests include cryptocurrencies and blockchains, peer-to-peer networks, and social network analysis. In these fields, she has co-authored 150+ papers published in international journals and conference/workshop proceedings. She has served as programme committee member and chair of several conferences and workshops. She has been involved in several research projects, and she is currently the coordinator of the Italian PRIN project "AWESOME: Analysis framework for WEb3 SOcial MEdia". She has been a member of the National Committee for the definition of the Italian National Strategy for Blockchain.

• • •