An investigation into the applicability of a blockchain based voting system

Craig Feldman

Oriel College
University of Oxford

Supervised by Professor Andrew Martin

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Computer Science

Trinity 2017

ABSTRACT

Electronic voting offers numerous cost, efficiency, and accessibility advantages over its physical counterpart. While adoption of electronic voting systems has increased, these systems are often not used in large-scale, security critical votes. This is undoubtedly due to the difficulty in developing and implementing a provably secure system, along with the cost of implementing these systems on a large-scale. Since the introduction of the Bitcoin blockchain in 2008, there has been considerable interest in utilising the blockchain in alternative fields. Interestingly, the properties of the blockchain indirectly guarantee many of the requirements of electronic voting systems.

Thus, this project investigates the feasibility of a blockchain based electronic voting system and introduces such a system after a careful analysis of the requirements of electronic voting and a thorough analysis of prior work in this field. Blind signatures or zero-knowledge proofs are used to have voters anonymously prove they are registered to vote by transferring a vote transaction to their chosen candidate's blockchain address. While not perfect, it is shown that this system can safely be used in a variety of elections, and offers promise for future research and development in this field.

ACKNOWLEDGMENTS

This project would not have been possible had it not been for my supervisor, Professor Andrew Martin, whose support, feedback, and input helped shape the direction of this dissertation. My friends and family provided fantastic guidance and encouragement, and many a thought provoking discussion around this topic. In particular, my parents have been a source of constant support, not just through this project but through my entire academic career – thanks mom and dad! A special thanks to Abdel and Thomas, who proved to be thoroughly entertaining study buddies and were always up for a game of croquet between writing sessions. Finally, I will be forever grateful to Standard Bank and the University of Oxford for awarding me the Standard Bank Derek Cooper Africa Scholarship, without which this truly life-changing year would not have been possible.

CONTENTS

1 INTRODUCTION

Voting is a critical aspect of any democratic society whether it be on a grand scale, such as voting for a political party, or on more minor issues such as voting within a company or organisation. Nonetheless, it is imperative that voting be fair and free of manipulation and influence. A number of important factors (discussed further in Section 2.1) are often required for fair voting and includes: the ability to vote anonymously, without intimidation, and under the guarantee that your vote will be captured correctly and included in the final tally. Traditionally, physical ballots have been used as an effective means of conducting voting, especially at the critical national level. While the system of physical voting appears to be simple, it in fact masks a far more complicated system than what one initially expects. This paper will investigate the complex requirements of electronic voting and investigate the use of the blockchain as a key framework for the development of a secure electronic voting system.

1.1 DEFINITIONS

The following terms are explicitly defined to give the reader a better understanding of how they are being used in the context of this paper.

- Blockchain – A distributed and chronological digital ledger that stores blockchain transactions and makes use of cryptographic principles to achieve strong immutability and security; Section 3 provides an in-depth description of the blockchain.
- Blockchain transaction – Usually the transfer of a cryptocurrency from one user to another, but here it may also refer to any conceivable transfer or blockchain event, for example, that of a vote from a voter to a candidate (often referred to as a 'vote transaction').
- Address – A representation of the public key of an asymmetric key pair, with the associated private key being used to sign blockchain transactions. Transactions originate from, or are sent to, addresses.
- Wallet – An abstract construct used to associate and manage a set of randomly generated public/private key pairs for the user.
- Electronic voting – Using electronic means to directly assist with the casting and/or tallying of ballots.
- Internet voting – An electronic voting system that makes use of the Internet to either transmit information or allow for remote electronic voting. Except where the distinction has been made clear, the term 'electronic voting' will usually be used interchangeably with the term 'Internet voting'.
- Blockchain voting – A form of electronic voting that makes use of blockchain technology to facilitate the voting procedure.
- Election/vote – These two terms will often be used interchangeably in what follows. When the distinction between large and small-scale votes is required, this will be made explicit. The focus of this paper will be on large-scale elections, unless indicated otherwise.
- Authentication organisation / voting authority – The organisation responsible for managing and administrating an election.

1.2 PROJECT MOTIVATION, AIMS, AND OBJECTIVES

Societies, organisations, and governments rely on the use of voting for many purposes: to solve disputes, elect leaders, and decide on a course of action (among others). Often, the cost of these votes is negligible – a vote in a boardroom may be decided by a simple show of hands. However, the monetary cost and effort to host larger scale votes can be fairly substantial, whether it be a shareholders' vote, or a national electoral vote. Furthermore, it is important that these votes be free, fair, and transparent, with an additional requirement of anonymity often necessary.

One tried and tested means of voting is to use a physical ballot box (see Section 1.4), with more modern techniques leveraging the power of technology to make the process more accurate and efficient, this is discussed further in Section 1.5. However, numerous technical and security challenges exist with the implementation of electronic voting schemes. Further to this, there has been a recent drive towards organisations and states modernising their voting procedures, which often results in the acquisition and adoption of costly (and often insecure) voting systems. Thus, this project aims to contribute research into the validity of a secure, inexpensive, and effective voting protocol.

The aim of this project is to investigate the feasibility of using the blockchain as a key framework in the development of an electronic voting system. In so doing, this paper will attempt to satisfy the following objectives:

- To investigate and consolidate the current academic literature surrounding electronic voting.
- To provide a clear and concise paper that details the requirements of a secure electronic voting system, along with the importance of such a system, and why, despite much academic and organisational research, this is still an open problem.
- To introduce and analyse a blockchain based approach to electronic voting, and finally conclude whether such a system is viable.

1.3 STRUCTURE AND METHODOLOGY

The general methodology and approach taken by this paper is as follows: A careful and considered review of the literature and an analysis against known and hypothesised solutions to electronic voting is provided. Thereafter, a blockchain based voting scheme is proposed that develops upon previous work in this field, with rigorous arguments for the correctness (and shortcomings) of the proposed scheme being provided.

In what remains of Section 1, a brief introduction to both physical and electronic voting will be discussed, along with the main advantages, disadvantages, and concerns within these systems. Section 1.6 will introduce several open questions in this field that will help to lay a foundation for what is to follow. Throughout this paper, these questions will prove critical in order to produce an effective and useful solution.

Section 2 aims to provide the reader with considerable background information on electronic voting. There are three major subsections, each one dealing with a different area of interest: The requirements of secure electronic voting, prior work on electronic voting, and finally, prior work on blockchain based voting.

Section 3, introduces the reader to the blockchain, with a particular emphasis on the Bitcoin blockchain. Other blockchain frameworks are introduced in Section 3.3. The

current known vulnerabilities of blockchain technology, along with how an attacker could take advantage of these vulnerabilities, is discussed in Section 3.4.

The unique contribution of this paper, that is, a blockchain based electronic voting system, is introduced in Section 4. The technical design of the system is first introduced, along with the reasoning behind the various design aspects. Section 4.5 then introduces the protocol as a whole and discusses how an election using this system would operate. The weaknesses and vulnerabilities that exists within this system are then introduced in Section 4.6. A comparison between the system introduced in this paper and other blockchain based voting implementations is provided in Section 4.7. Finally, conclusions and areas for future work are discussed in Section 5.

1.4 WHY PHYSICAL VOTING REMAINS FAIRLY UBIQUITOUS

One of the main advantages of physical voting is that it does not require complete control of the election to be handed over to a third party who may in turn manipulate the vote. Instead, the system is distributed across many layers, with an independent[1] third party managing the process. While it is possible for physical votes to be manipulated, it is exceptionally difficult for this to be conducted on a large scale, due to the number of people that would be required to collude. These voting systems are also vastly redundant: a breakdown in one precinct doesn't cascade to undermine the whole operation. Further to this, physical ballots provide a record of the vote – the actual marked ballot – making manipulation more difficult. Nonetheless, various forms of ballot tampering and election rigging remain possible, with an in-depth discussion on electoral fraud being provided by Lehoucq [1].

There are, however, two major disadvantages of physical voting over electronic voting. The first is the monetary cost of co-ordinating large-scale votes and the other concerns the inflexibility and lack of expressive power of the votes: once a question has been taken to a vote, it is extremely difficult to modify it (if need be) or pose additional questions based on the result of the previous votes; for example, runoff voting is difficult to implement under a physical voting system as it requires first tallying the initial vote, and then hosting a second round of voting based on the first round's results.

1.5 AN INTRODUCTION TO ELECTRONIC VOTING

Electronic voting is not constrained by the processing limits of physical voting. It allows for extremely agile voting where questions can be posed to voters both proactively and reactively, with results being available almost immediately. The concern with electronic voting, however, is that it often requires a third party to manage the process, with it being almost impossible to create a provably fair system - one first has to trust the third party to develop fair voting software and then find a means of ensuring that the legitimate, unmodified, software is in fact running on the machine that one is voting with. Even if we

[1] While we are assuming that the third party is independent, even if this is not the case, it would prove extremely difficult for the third party to have any meaningful manipulative power under a physical voting system. This is largely due to the number of people, across multiple layers, required to co-operate to manipulate results.

employ a 'fourth party' to audit the third party, we are simply shifting our trust assumption to the fourth party who, if corrupt, could modify the system.

An important question to ask is whether or not parties have to collude in order to defeat the security of the system. Within electronic systems it is often possible for a single party to manage to defeat the system's security, but physical voting often requires the collusion of multiple parties due to the many independently operated layers of the system.[2]

While minor votes (say, within an organisation) may be fairly inconsequential and hence attract little attention of hackers or manipulative players, important votes (such as national votes) may have enormous consequences depending on the outcome and may draw the attention of attackers. Furthermore, even if the voting software is secure, the data has to be transmitted to a central server which is responsible for tallying all the electronic votes, adding another potentially vulnerable domain to the system. Until recently, this was purely theoretical, but now allegations of Russian involvement in the 2016 US presidential election (see [2, 3]) make the threat quite tangible. The existence of vulnerabilities within electronic voting systems is discussed further in Section 2.2.5

Thus, the main concern with electronic voting is that it appears to be impossible to guarantee a fair system, and even if a fair system is being used, the nature of electronic systems means that a vulnerability may make it just as easy for an attacker to manipulate one vote as it is for him to manipulate one million votes. Furthermore, it is a lot harder to prove a fair (or even unfair) vote has occurred when using electronic voting, as it is easier to manipulate electronic records than it is physical records.

1.6 General open questions

Electronic voting gives rise to three important questions concerning the security and implementation of a system that guarantees anonymity, integrity, and overall security. Throughout this paper, these questions will form a key aspect of developing a secure electronic voting system:

1. Is it possible to guarantee the same level of security[3] and anonymity as physical voting through an electronic voting system?

This is by far the most general and broad open question regarding electronic voting, with work on potential implementations being conducted by organisations (for example, 'Follow My Vote' who are attempting to use a blockchain based voting approach [4]) and in several academic papers that have discussed the viability and methods in which electronic voting can be conducted (e.g. [5, 6, 7]). Recently, papers looking at the relatively new approach of leveraging the power of the blockchain for electronic voting have emerged (e.g. [8, 9, 10, 11]). While the theoretical possibility of using a blockchain based voting system has been demonstrated in these papers, several issues still exist with these implementations. This prior research is discussed in detail in Section 2.2 and 2.3.

[2] The complexity of this system and the multiple layers involved has been succinctly depicted by Margot-Duclot [79].

[3] It is worth asking what is meant by 'security' in the context of electronic voting. This is elaborated on in Section 2.1.1, which highlights some of the key security requirements of electronic voting systems. However, one of the key properties that security refers to is that of system integrity – it should not be possible for a malicious actor to modify the system or alter voting data.

Further to this, anonymity is often required in voting, hence one needs to ensure that it is virtually impossible to link a vote to the identity of the voter. This problem, with specific reference to the proposed implementation, is discussed in Section 4.4.

2. How do we prevent a plutocratic system from developing?

Here plutocracy refers to not just monetary wealth, but rather a considerable stake in the blockchain being used, which makes certain attacks on the blockchain more viable. While this question can be seen as a subsidiary question to the first, it is explicitly stated as it more clearly separates out the vulnerabilities that are unique to a blockchain based implementation.

One needs to prevent users from being able to cast more votes than they are allowed. As decentralised consensus (see Section 3.2.4) is a key aspect of securing transactions in the blockchain, we need to ensure that this system is distributed and decentralised. In other words, we do not want powerful individuals, organisations, or states to be able to have sufficient resources to attack the distributed consensus mechanism of the blockchain and hence violate integrity of the system. Section 3.4 discusses some of the vulnerabilities that an attacker with considerable computing resources could perform against the blockchain.

3. How can a user be sure that a machine used for voting is not running any malicious software that could compromise the user's vote?

Malicious software, such as key loggers and viruses, now pose a threat in that they could tamper with the connection to the server and modify the data being sent and received, or even steal the user's private key (used to sign a vote) and login details. As such, the user needs to be assured that either their machine[4], or a public voting machine, has not had its integrity compromised and that their vote has been tallied as cast.

[4] Section 4.3 discusses why it would be unwise to allow remote voting.

2 BACKGROUND

2.1 THE REQUIREMENTS OF SECURE ELECTRONIC VOTING

One of the key general requirements of any election is that it be fair and free from manipulation. That is, all registered voters should be able to vote free from intimidation or duress, and with no bias being imparted on them as a result of the election system design or via the actors of the system. The security requirements of voting in general are neatly summarised by Article 21 of the Universal Declaration of Human Rights [12] which encompasses voter privacy, accurate tallying, 'one man, one vote', and the freedom of the vote. Broadly speaking, these general requirements can be broken down into both technical and societal requirements.

2.1.1 Technical and implementation requirements

When developing an electronic voting system, it is imperative to be cognisant of the security requirements of these systems. One need only look at the abundance of literature surrounding the topic to be made aware of the many difficult challenges and requirements involved. In general, many of the requirements for paper-based voting also apply to electronic voting, except one needs to consider new potential vulnerabilities introduced through the use of technology. In 1993, Neumann [13] noted that there was currently no generally accepted standard that voting systems were required to satisfy and hence proposed a set of generic requirements that electronic voting systems should aim to satisfy. The main requirements of these systems are listed as follows:

- **System integrity** – The computer systems used in the election process must be tamperproof, with modifications ideally being prohibited during the election process.
- **Data integrity and reliability** – The votes cast must be recorded correctly and must be immutable once recorded.
- **Voter anonymity and data confidentiality** – There must be no known association between a voter's identity and his recorded vote. Furthermore, the current vote tally must be protected from access during the election.
- **Operator authentication** – Any authorised administrators of the election must be able to gain access only via non-trivial authentications (for example, requiring a single password to authenticate the user is inadequate) with no trapdoors existing in the system.
- **System accountability** – All internal operations must be monitored via an immutable and permanent audit log.
- **System openness** – At any time, any part of the system must be open for random inspection by auditors.
- **System availability** – the system must be protected against denial of service attacks and should be available during the time it is required to be operational.
- **System reliability** – System development should minimise the likelihood of bugs and malicious code.
- **Interface usability** – The interface should be "inherently fail-safe, fool-proof and overly cautious in defending against accidental and intentional misuse."
- **Documentation and assurance** – All procedures involved in developing and using the system must be clearly and consistently documented.

- **Personnel integrity** – Anyone involved in the development, operation, or administration of the system must be of "unquestioned integrity".

The above requirements provide a good framework but are incomplete; often other factors exist that need to be considered when developing electronic voting systems. As an example, Saltman [14] notes that the system must conform with any applicable regional electoral laws and provide equal access to handicapped voters, among many other factors. Neumann [13] also notes that even if we ignore the incompleteness and imprecision of the above suggested criteria, these criteria are often inherently impossible to satisfy with any meaningful assurance. He further notes that, "the actions of a single person may be sufficient to subvert the process, particularly if preinstalled Trojan horses or operational subversion can be used." As a result, he concludes that, "the mere existence of generally accepted criteria coupled with claims that a system adheres to those criteria might give the naive observer the illusion that an election is nonsubvertible [sic]." As such, these factors will always lead to questions and concerns about the use of computer systems in elections.

2.1.2 Societal requirements

A report released by the Internet Policy Institute [15] aimed to examine the feasibility of Internet voting and looked at not just the technical requirements, but also the social requirements involved with using an Internet based electronic voting system. These societal factors include how electronic voting may affect different demographics, public opinion and trust towards such schemes, as well as the convenience factor: election systems should not require extra skills or unreasonable equipment to be viable.

The issues surrounding electronic voting from both a technical and a societal view are discussed in detail by Gritzalis [7] and broadly echo what has been stated in this section already. Further work on establishing the requirements and feasibility of secure electronic voting was conducted by Rubin [16] who also discussed social engineering based attacks. Rubin argued that: "Introducing state-of-the art technology into the election process implies new risks that may not be worth taking." Rubin further noted that: "The adversaries to an [national] election system are not teenagers in garages but foreign governments and powerful interests at home and abroad." In other words, while small-scale private elections (for example, boardroom elections) may have limited impact and thus not attract as much attention, national elections have many powerful stakeholders, with different results leading to potentially very different economic, social, and political outcomes. Often these stakeholders have great incentives to alter the outcome of an election in their favour, thus making electronic voting systems a major target of attack.

2.2 ELECTRONIC VOTING SCHEMES

2.2.1 The history of electronic voting

Electronic voting is not a recent phenomenon. The first electronic voting systems began to emerge in the 1960s with the introduction of punched card ballots, where tallying could be conducted using standard punched card tabulating equipment [7]. Punched card systems received considerable notoriety in the 2000 US presidential elections, with their use being blamed for Al Gore's loss to George Bush in the state of Florida [17, 18].

The progression of electronic voting systems continued with the introduction of optical scan voting systems that allowed for optical scanners to read marked paper ballots and tally the results. The advantage of using optical scanners, coupled with a traditional physical ballot, is that it negates the need for voters to learn to use a voting machine (as only pen and paper are required to cast a vote). This was followed by the introduction of direct recording electronic voting machines in the early 1970s [7]. These systems display the ballot to the user and capture input directly, via touchscreens or buttons; electronic means are used to capture, process, and tally the votes.

Electronic voting systems are able to leverage the power and speed of technology, which (if used correctly) will almost certainly be able to tally the votes faster and more accurately than the more traditional physical counting of ballots. Furthermore, both punched card and optical scanning systems produce a tangible record of a voter's intent, allowing for manual recounts to occur in case of disputes. The disadvantage of using a paper ballot is that they are still subject to several forms of electoral fraud and ballot stuffing.

2.2.2 Electronic voting implementations in the literature

The potential benefits of electronic voting, along with major security concerns, has led to it attracting much attention in academic literature. A considerable amount of research has been devoted to the topic, with various papers looking at potential improvements and recommendations. A large amount of literature emerged in the late 80s and 90s on various types of electronic voting schemes (see [6] for a list of the literature, along with literature on receipt-free voting schemes[5]), most likely due to the increased adoption of computer systems and the Internet around this time. An example of one widely cited electronic voting system was presented in 1999 by Schoenmakers [5] who presented an application of his publicly verifiable secret sharing (PVSS) scheme to create a "universally verifiable secret ballot [electronic] election scheme" and noted the performance benefits of using their PVSS based election system for small-scale elections, compared to the use of threshold based decryption schemes[6] as described in [19].

It is worth making a distinction between electronic voting in a polling station, and remote electronic voting via the Internet. While they have considerable technology in common, they also have significantly different risk factors. Much work has been conducted on developing Internet connected electronic voting systems that would allow for votes to be conducted remotely via the Internet (for example, see [20, 21, 22, 23]). The problem with such a system is that without careful system design, the threats of coercion and vote buying are potentially far more problematic when remote voting is allowed. While Internet-based electronic voting does not necessarily introduce these problems, it does have the potential to exacerbate them by extending the reach and data collection abilities of an attacker. The viability of remote electronic voting is discussed further in Section 4.3.

2.2.3 Usability and implementation concerns

Several countries have experimented with electronic voting systems, arguably the most famous of which is the United States in which most voters make use of direct-recording electronic devices (such as touch screens) and optical scan ballots to cast their votes [24]. The 2000 US election (discussed earlier) highlighted a key fact: the manner in which voters

[5] Receipt-free voting methods generate no evidence of whom a voter voted for.

[6] Threshold cryptosystems require that decryption only be possible when several parties (more than some threshold number) co-operate in the decryption protocol.

cast their ballot is important; the ballot design and voting technology used can affect election results by, for example, imparting bias for or against certain candidates due to their physical placement on the ballot.

The above concerns led Bederson et al. [25] to investigate potential usability issues within electronic voting. Their research highlighted several important concerns, as well as the fact that the development of voting systems poses not only major security requirements, but that they also present a unique challenge to interface designers. This is because these systems need to be usable by every eligible voter, including the disabled, elderly, uneducated, and poor. The authors also argue that the systematic issues of how voting machines get purchased and evaluated are problematic, with concern being about cost, rather than usability. The authors further argued that the issues raised in their paper regarding voting system usability, may together lead to specific classes of voters becoming disenfranchised. These systems, however, also offer promise from a usability perspective – from the opportunity to alter the language and font size on demand, to offering disabled users customized access.

2.2.4 Electronic voting implementations: A look at Estonia

In 2011, Kumar & Walia [26] conducted a review of several countries that use, or have trialled, electronic voting and discuss the various systems used and issues encountered. Although several countries have implemented and trialled electronic voting, the first practice of countrywide, binding Internet voting occurred in the 2005 Estonian local elections where the whole Estonian electorate had the ability to cast their vote via the Internet [27].

Further research on electronic voting in Estonia was conducted in 2009 by Alvarez et al. [28] who argued that although several countries such as Switzerland, the United Kingdom, and the United States have all trialled Internet voting in binding public elections, Estonia had advanced farthest in deploying Internet voting. The authors focused on how "Estonians have systematically addressed the legal and technical considerations required to make Internet voting a functioning voting platform, as well as the political and cultural framework that promoted this innovation." The authors proposed three lessons that can be learnt from the Estonian case of Internet voting:

1. Context: The context of the nation and election can impact the success of electronic voting. Estonia is a small and highly centralised nation, with a well maintained and comprehensive voter registry coupled with a very simple ballot. The simple ballot allowed for the potential complexity of election administration (for example, the design and implementation of a remote Internet voting system) to be reduced.
2. An adequate legal and regulatory framework: Estonia employed a careful and deliberate political process to examine the feasibility and notion of Internet voting before rolling out the system. Furthermore, legislation within Estonia requires all citizens to hold a digital identification card that includes a digital certificate embedded in the card. When combined with a PIN code, this card can be used for online authentication and to digitally sign interactions between citizens and the government.
3. Voter authentication: Having a strong form of online voter authentication is a critical step in the implementation of a trusted and secure electronic voting

system. These strong forms of identification (here an identification card with an embedded digital certificate) have applications not just in electronic voting, but additionally for several other interactions between citizens and the state: from paying fines, fees, and taxes, to checking out library books.

2.2.5 Are current systems secure?

While many papers have described the issues with electronic voting systems from a theoretical and technical perspective, many of these issues have also been demonstrated in practice. Even in the case of Estonia (which is widely seen as a success story [29]), an investigation into the security of this system recommended that Estonia discontinue the use of their Internet voting system due to numerous security vulnerabilities [30]. This is not a unique problem with current implementations of electronic voting; Kohno et al. [31] provided a security analysis of a US electronic voting machine in operation in 2004 and found it to be "far below even the most minimal security standards applicable in other contexts". Several other papers have aimed to demonstrate potential flaws within in-use electronic election systems (for example [17, 18, 32, 33, 34, 35], among numerous others), showing the dire need for security improvements in this field.

One possible attack scenario would involve spreading malware into voting machines and rigging them to shift a few percent of the vote to favour a desired candidate. The malware would likely be designed to remain inactive[7] during pre-election tests, erasing itself once the polls close.

2.3 BLOCKCHAIN BASED VOTING

2.3.1 End-to-end verifiability

One of the key criticisms of the Estonian based voting system is that it does not satisfy a property known as end-to-end (E2E) verifiability [30]. E2E verifiable systems enforce stringent integrity properties and strong tamper resistance by allowing voters to verify that their votes have been counted as cast, while still maintaining the secrecy of their ballot. Most electronic voting schemes proposed in the literature make use of cryptographic techniques to achieve E2E verifiability (for example, see [36, 37, 38]). This highlights one of the key sought after features of electronic election systems: to remove the need for a trusted third party to oversee major aspects of the voting process.

While it is practically impossible to alleviate the need for third parties to participate in elections, one can aim to limit their involvement and potential to commit electoral fraud. Recently, several papers have looked at the relatively new approach of leveraging the power of the blockchain[8] to develop voting protocols that reduce the reliance on third parties and achieve E2E verifiability. This is discussed further in the following sections.

2.3.2 The versatility of the blockchain

While the blockchain was originally intended to be used as a digital ledger for the management of cryptocurrency transactions, its use for non-cryptocurrency based

[7] For example, Volkswagen's emission control software, while not malware, was designed to detect when an emissions test was being performed and produce false reports [80].

[8] Although this section makes several references to the blockchain, at this stage it is not necessary for the reader to have a deep understanding of the blockchain (the definition provided in Section 1.1 should suffice); the blockchain is discussed in detail in Section 3.

transactions has been proposed in several papers, with researchers focused on attempting to apply the blockchain to several other domains and problems. These problems range from co-ordinating the Internet of Things [39] and secure data management [40] to healthcare [41]. Several compelling specific applications of the blockchain in both the financial and non-financial sector, along with the challenges and business opportunities ahead, are discussed by Crosby et al. [42] and Swan [43]. Many of these alternative uses of the blockchain are hindered by the Bitcoin blockchain's limited flexibility, hence the development of alternative blockchains such as Ethereum (see Section 3.3.1) to allow for a broader range of functions to be carried out on the blockchain.

2.3.3 Prior work on blockchain based voting

Despite the blockchain being a relatively new framework, research has already been conducted into investigating the effectiveness of blockchain based voting systems. Several academic papers have proposed protocols that leverage the power of the blockchain, with a few organisations trialling and even implementing blockchain based systems in the real world. This prior work is discussed in the following subsections.

2.3.3.1 Academic implementations

Several academic papers have discussed the application of the blockchain to electronic voting. For example, McCorry et al. [9] present 'The Open Vote Network' – an Ethereum based, decentralised and self-tallying Internet voting protocol. The authors state that it is the first implementation of a blockchain based voting protocol that does not rely on a trusted third party to tally the votes or protect the voters' privacy. This is achieved by using Ethereum to enforce the voting protocol's execution[9]. This contrasts with the implementation provided by Lee et al. [10] in which they introduce a trusted third party to authenticate voters.

In electronic voting, voters must be authenticated to verify that they do indeed have the right to vote. This is problematic in blockchain implementations, as we need to ensure that the digital signature used to sign a transaction/vote belongs to one of these authorised voters. However, maintaining a simple list linking public keys to a voter's identity would not sufficiently satisfy the voter anonymity requirement discussed in Section 2.1. This is because any breach of this database would allow an attacker to immediately view the identities of the voters, along with who they voted for, by examining the blockchain and the digital signatures used to sign votes. Lee et al. [10] get around this issue by using a trusted third party, separate to the authenticating agency[10]. Their system works as follows:

1. A potential voter hashes a secret message, known only to them, and sends the hash to the authenticating agency who links this to the voter (provided he is on the voting register).
2. The voter sends the secret message's hash to the third party, who forwards this value to the authenticating agency to confirm that the hash exists and has hence been linked to an authorised voter.

[9] The voting protocol used by the Open Vote Network was first introduced by Hao et al. [69] before an Ethereum based version was developed.

[10] The authenticating agency, in this case, is any agency that would normally be tasked with verifying whether a voter has the right to vote in an election. This is usually done by checking that their identification number is on a list of authorised voters.

3. If the hash exists, the third-party stores only the voter's public blockchain address in a list. This list then indicates which addresses are registered to vote, without linking it to a specific voter.

2.3.3.2 Practical implementations

Apart from the applications of blockchain based voting in academic literature, several organisations are also attempting to develop implementations that use blockchain based voting, for example:

- 'Follow My Vote' [4] who provide an open-source solution that allows users to audit the entire election process and watch elections progress in real time. Their system also allows users to alter their vote during the election. Follow My Vote uses the BitShares[11] blockchain.
- Blockchain Technologies Corp's 'Blockchain Apparatus' [44] utilises their own blockchain, VoteUnit, which (unlike the Bitcoin blockchain) does not require fees to ensure transactions are added to the blockchain within a certain time. To vote for a specific candidate, a 'vote unit' is transferred into the candidate's unique wallet (blockchain address), with the total number of votes for a candidate being the number of vote units received from unique and valid voters.
- 'BitCongress' [45] is a decentralised legislation and voting blockchain platform. Coupled with BitCongress, is a tool called 'Axiomity' that allows for voting that leverages the power of the blockchain.
- TIVI [46] is marketed as "a convenient and secure online voting solution allowing governments to connect with their remote voters by providing them with a transparent and universally verifiable platform." TIVI makes use of the blockchain to help guarantee the integrity of the elections, although their exact implementation details are not mentioned.

2.3.4 Summary and next steps

The academic literature and current implementations of blockchain based voting highlight a number of risks, problems, and solutions that can all be analysed and modified to develop an improved voting system. This is discussed further in Section 4, where this paper's unique contribution of a blockchain based voting system is introduced. However, it is first necessary to develop an in-depth understanding of the mechanics and technical details of the blockchain. This is discussed in the following section.

[11] https://bitshares.org/

3 THE BLOCKCHAIN

The fundamental framework behind the electronic voting method proposed in Section 4, is a distributed database system known as 'the blockchain'. The blockchain was first proposed in a paper presented under the pseudonym 'Satoshi Nakamoto' in which the framework that would allow for a purely peer to peer version of electronic cash (that negated the need for transactions to be supported by an independent third party) was proposed [47]. This cryptocurrency, 'bitcoin'[12], would make use of cryptographic protocols and a distributed ledger system – the blockchain.

3.1 AN INTRODUCTION TO THE BLOCKCHAIN

The blockchain is a continuously growing data structure that utilises cryptographic hashes to link and secure records (known as blocks) to form a virtually immutable record of past events [48]. It is distributed and decentralised, with no single owner, yet anyone can download a complete copy of the blockchain that can be trusted as the authoritative and complete record. This decentralised consensus mechanism (see Section 3.2.4) is largely reliant on a clever use of a 'proof of work' component (see Section 3.2.3) to ensure consistency and consensus among all copies of the blockchain.

While Bitcoin was seen as an exciting and innovative electronic payment system, for many, the mechanics behind it were seen as an even more important technical innovation than bitcoin itself [49]. While most databases today are centralised, and controlled by a single party, the blockchain is inherently decentralised and distributed across a number of peers. In this distributed system, each peer has a copy of the entire database, with their respective copies being periodically updated and synchronised via the Internet. This database system can be open to all (public) or provide restricted access to a select set of users (private) [50]. Essentially, the main benefit of the blockchain over traditional database models is that it does not require an additional trusted party to maintain a database that is used by two or more parties, but instead relies on a distributed consensus mechanism to achieve agreement among users about the true state of the database.

3.2 BLOCKCHAIN MECHANICS AND STRUCTURE

While the mechanics and structure of the blockchain can vary depending on the specific implementation, the standard blockchain developed as part of the Bitcoin protocol is described in the following subsections. Alternative (i.e. non-Bitcoin based) blockchains are discussed in Section 3.3.

3.2.1 Blockchain nodes

The blockchain is inherently a peer to peer system comprised of a number of connected nodes. These nodes can either be full network nodes or lightweight nodes and are discussed in depth by Antonopoulos [51].

[12] A note on capitalisation: 'Bitcoin', with capitalisation, is used when describing the concept of Bitcoin, or the entire network itself; without capitalisation, it is used to describe bitcoins as a unit of account.

3.2.1.1 Full nodes

Full nodes form the basis of the Bitcoin blockchain network as they enforce all the rules governing the Bitcoin network and transactions (transactions are discussed in Section 3.2.2.6). Full nodes maintain a complete and up to date copy of the blockchain, along with the associated transactions. Every full node independently builds up the blockchain starting from the first block (the genesis block) by requesting deeper blocks from other nodes, verifying them, and then appending them to their current copy of the blockchain. Although a full node relies on other nodes to receive new blocks (which it then verifies and incorporates into its own blockchain copy), it does not need to rely on any other node or source of information to verify prior transactions as it has a complete copy of every transaction.

3.2.1.2 Lightweight nodes

Unlike full nodes, lightweight nodes do not maintain a full copy of the blockchain and are thus suitable for space and power constrained devices. These nodes download the block headers (see Section 3.2.2) from other nodes, but do not download the transactions included in the block. Through a process known as 'simplified payment verification', lightweight nodes can still verify the existence of a transaction in a block (by establishing a link between the transaction and the block that contains it) by making use of the Merkle root (see Section 3.2.2.3). The details of this process are not discussed further but can be found in the original Bitcoin whitepaper [47].

While these nodes are capable of verifying the existence of a transaction in the blockchain, they cannot verify the validity of transactions and instead rely on the fact that other (full) nodes have accepted the transaction into the blockchain (preferably by waiting until several blocks have been added on top of the block that contains the transaction). While lightweight nodes can never be persuaded that a transaction exists in a block when it does not, the existence of a transaction can be hidden from a lightweight node as it does not contain a record of all transactions. This vulnerability can be exploited in several attacks such as the double-spend and denial of service attack (see Section 3.4). However, by connecting randomly to several nodes one can increase the probability that the node is in contact with at least one honest node, which greatly reduces the chance of a successful attack.

3.2.1.3 Mining nodes

One of the interesting features of the blockchain is that transactions are unconfirmed until a group of them have been collected into a block and 'mined' by mining nodes. These nodes are responsible for listening for new transactions that are distributed by some random peer and collecting a few of these into a block, adding in additional data to link it to the parent block, and randomly iterating over millions of possible nonce values until the resulting block's hash conforms to a set of rules (see Section 3.2.3). Once a valid hash has been found, mining nodes transmit the block to other nodes, who in turn distribute the block to their peers.

One problem with this model is that it generally requires considerable computation power and resources to successfully mine a block. Thus, an incentive for doing so is required; In the case of bitcoin, successful bitcoin miners are rewarded with a fixed bitcoin value for mining a block, with the possibility of including transaction commission at a later stage.

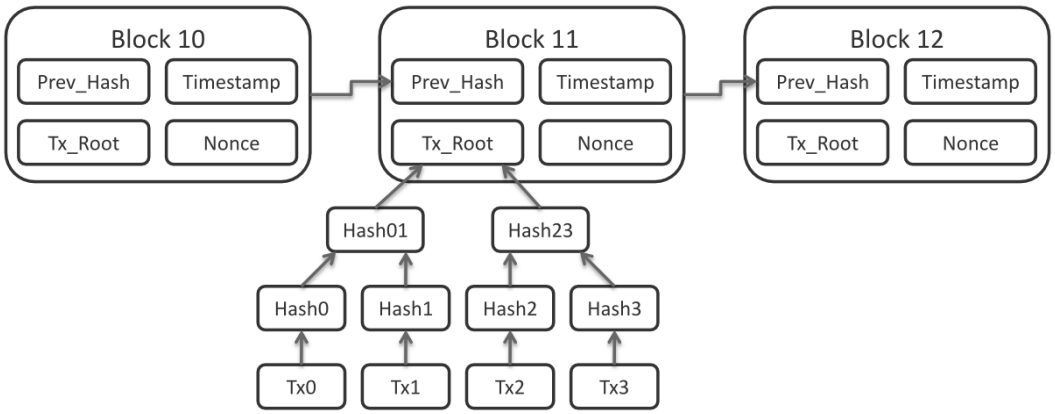

Figure 1 - The structure of the blockchain along with some important fields in the block's header. Here blocks 10, 11, and 12 are shown, along with the formulation of the Merkle root.

3.2.2 Blocks

The blockchain is essentially comprised of a number of 'blocks', where each block is cryptographically linked to the previous block to form a virtually immutable chain. In the case of Bitcoin, the blockchain forms a log of all the transactions that have ever been verified on the Bitcoin network. Any new and unconfirmed transactions are broadcast to all mining nodes on the server. These nodes then collect the transactions into a block and begin work on a difficult 'proof of work' for its block (see Section 3.2.3).

Each block contains the transactions associated with that block, as well as a header comprising of several pieces of information, including: a block version number, a timestamp, a reference to the parent (previous) block, a Merkle root, a target, and a nonce. The fields within a block's header are hashed to generate the current block's hash. The structure of these blocks is shown in Figure 1[13]. The block version number can be used to indicate which set of block validation rules[14] to follow and can prove useful in allowing the system to be updated and improved with time.

3.2.2.1 Timestamp

The timestamp simply indicates the time when a block was mined and serves two purposes: firstly, it serves as a source of additional variation for the block's hash, and secondly, it makes it more difficult for an adversary to manipulate the blockchain. This is achieved by only accepting a timestamp on a candidate block (a block that has not yet been accepted and included in the blockchain) as valid if it is greater than the median timestamp of the previous 11 blocks and less than the median of the timestamps returned by all nodes in the network plus 2 hours [51]. Thus, by imposing both an upper and lower bound on valid timestamps for a particular block, one can safely ensure a node was in fact mined around the time specified.

3.2.2.2 Previous block's hash

In order to form a chain of blocks, it is necessary to keep track of all preceding/parent blocks by linking them to any new block. This is achieved by storing a 256-bit hash of the previous block's header in the current block. As this is a field inside the block's header it thereby affects the current block's hash. Thus, any change to the parent block will result

[13] Illustration: Matthäus Wander. Reproduced under Creative Commons Attribution-ShareAlike License. https://commons.wikimedia.org/wiki/File:Bitcoin_Block_Data.png.

[14] A set of validation rules that full nodes follow to stay in consensus with other nodes.

in a change to the child's own identity, as any modification to a parent block necessitates a change in the child block's pointer to the previous block. This in turn leads to a change in the child's hash which requires that the grandchild block's pointer be changed, affecting the hash of the grandchild block. This cascading effect guarantees that a modification of any block must lead to the recalculation of all subsequent blocks. As a large amount of computational resources are required to re-mine these modified blocks, the long chain of blocks makes the blockchains previous history virtually immutable [51].

3.2.2.3 The Merkle root

The Merkle root ('Tx_Root' in Figure 1) is a 256-bit hash based on all of the transactions contained in the block and is used to verify a set of transactions. In Figure 1, the transactions included in block 11 are represented by 'Tx0' to 'Tx3'; the transaction identifications are represented by hashes of these transactions, which are then used to construct a Merkle tree. The Merkle tree is constructed by pairing each transaction hash with another transaction hash and hashing them together, with any unpaired transaction hash being paired with itself. The resulting hashes are paired with another hash and hashed again with this process being repeated until only one hash remains, the Merkle root. This process guarantees that any modification to an individual transaction, or the order of the transactions, will result in a change in the Merkle root. The Merkle root thus provides a cryptographically secure proof of which transactions have been included in a block and that their integrity has not been compromised [51].

3.2.2.4 Difficulty target

An interesting and fundamental feature of the blockchain is its ability to self-regulate the rate at which blocks are added to the blockchain. In the Bitcoin framework, this is achieved via the inclusion of a difficulty target. With all the other fields within a block complete, the goal of a miner is to find a valid nonce value such that the hash of the block's header conforms to the proof of work requirement by being less than the difficulty target [51]. Proof of work is discussed further in Section 3.2.3.

3.2.2.5 The nonce

The final field within the block header is an arbitrary integer value that can be altered by miners in order to produce a hash less than or equal to the target threshold as discussed in Section 3.2.2.4. In the case where all valid nonce values fail, the timestamp field could be updated as it is included as an input to the hash function.

3.2.2.6 Transactions

The actual payload of a block contains a number of bitcoin transactions. These transactions essentially detail the transferal of bitcoins from one user to another. In this paper, the term transaction (when referring to transactions within a block) will refer not only to bitcoin based transactions, but in a much broader sense to any payload attached to a block. In the case of voting, this could be a set of votes that have been cast. Transactions are digitally signed by the user who generates the transaction to provide guarantees of integrity and non-repudiation.

3.2.3 Proof of work

One of the key features of the blockchain is its ability to remove the need for an independent third party and generate a trusted, decentralised database. Since any node on the network can suggest a new block, the system needs to be able to agree on what

the next block should be. The ability to achieve distributed consensus (see Section 3.2.4) is largely reliant on a proof of work system similar to the one introduced by Adam Back [52], known as 'Hashcash'.

Proof of work systems were first introduced by Dwork & Naor [53] in 1992, primarily as a means to combat junk mail. The term was later coined and formalised by Jakobsson & Juels [54] who describe the system as being used to demonstrate to a verifier that a certain amount of computational work has been performed within a specified time.

The proof of work function introduced by Nakamoto [47] involves trying to find a set of inputs that when hashed using SHA-256, has a specified number of leading zero bits. The work expended increases exponentially with each additional leading zero bit required. In the case of the blockchain, this is implemented via generating random values for a nonce (as discussed in Section 3.2.2.5) until a value is found that gives the block's hash the required number of leading zero bits. Since this requires considerable computation, once a block has been mined that satisfies the proof of work condition, it cannot be altered without redoing the work. As blocks are chained onto previous blocks, the work to change an earlier block would include redoing all subsequent blocks, hence earlier blocks become more and more secure as the chain length increases. Such a system forms the basis of the creation of a decentralised consensus mechanism.

While a typical computer may take months or even years to find such a value, the difficulty target (as discussed in Section 3.2.2.4) allows the system to automatically adjust how frequently a block is added to the chain by factoring in the desired rate of block adoption and the total processing power of the mining nodes on the blockchain network. This block adoption method does however pose a problem if two or more nodes simultaneously find a valid nonce and distribute their blocks to the network. Achieving this decentralised consensus is a key aspect of the blockchain and is discussed further in the following section.

3.2.4 Decentralised consensus

Despite the fact that the blockchain is not governed or maintained by a central authority, the system allows every full node to have a complete copy of a public ledger that can be trusted as the authoritative record. Even though all data is transmitted across unsecure network connections, we can leverage Nakamoto's main contribution of a decentralised mechanism for 'emergent' consensus. The term 'emergent' derives from the fact that the consensus is not achieved explicitly but rather as an artefact of the asynchronous interaction of thousands of independent nodes, constrained only by a simple set of rules [51]. According to Antonopoulos [51], the interplay of four processes that occur independently on nodes across the network give rise to emergent consensus:

- Having every full node independently verify each transaction based on a list of criteria that those transactions must follow.
- Having a set of mining nodes independently aggregate transactions into new blocks, coupled with demonstrated computation through a proof of work algorithm (as discussed in Section 3.2.3).
- Having the nodes independently verify the new blocks and add new blocks onto the previous blocks, forming a chain.
- Having every node independently select the longest verified chain as the main chain.

The last point allows for nodes on the network to decide between competing chains. As mentioned earlier, it may often be the case that more than one block is mined simultaneously. In such a case, the successful miners push these blocks to other nodes on the network. However, since multiple nodes have solved the problem simultaneously, we have ambiguity at the end of the chain, with different nodes potentially maintaining different versions of the blockchain. Each node maintains three sets of blocks [51]: the main blockchain (the chain with the most cumulative proof of work associated with it), secondary chains (chains that form branches off the main chain) and finally, orphan blocks (blocks with no known parent). An invalid block is rejected instantly and is therefore not included in any chain.

By summing the difficulty associated with each block, a cumulative difficulty for a chain can be computed. At any given time, a node will select the chain with the most cumulative difficulty as the main chain, with other chains forming secondary chains. Secondary chains are maintained in the event that a new block extends a chain that is not in the main chain. When a new block is received, a node will attempt to add it to the main chain. If the referenced parent can be found and is at the tip of the main chain (i.e. it has no child block), it is appended to that chain. If the referenced parent has already been extended with a child block, the new candidate block is connected on a secondary chain. Now, if the proof of work associated with this secondary chain is greater than that of the main chain, the newly extended chain becomes the new main chain. It can also be the case that when a new block is found quickly, not all nodes would have received the parent block when the new block arrives. In this case, the block is added to a pool of orphan blocks to be referenced later once the parent is received.

Since each node is required to set the chain with the most cumulative difficulty as the main chain, the network eventually converges to a consistent state. Forks in the main chain can occur as temporary inconsistences caused by two or more candidate blocks competing to form part of the longest chain. These forks are however resolved by the eventual convergence to a universally accepted main blockchain as more blocks are added to one of the forks. However, forks can also be initiated deliberately by the community, as was the case in 2016 when the Ethereum[15] community, after much debate, initiated a fork to restore stolen funds [55]. Another recent and major deliberate fork involved Bitcoin splitting into 'Bitcoin Classic' and 'Bitcoin Cash' in response to the community remaining divided about whether to adopt new network rules [56]. As an example of how forks can occur, assume there is currently a universally accepted main chain of blocks 'A-B-C' with miners competing to extend block C:

If two miners both find competing candidate blocks X and Y, they begin broadcasting these blocks. Depending on the topological distance between nodes (impacting which block is received first) some nodes will accept 'A-B-C-X' as the longest proof of work chain and ignore block Y, while others will accept 'A-B-C-Y'. Miners begin work on extending either block X or Y, depending on which one they received first. Once a candidate block Z is found, creating a chain 'A-B-C-Y-Z', that solution is broadcast to the other nodes on the network. Any nodes that had previously accepted the 'A-B-C-X' chain as the main chain will now make 'A-B-C-Y-Z' the main chain (as it has more cumulative work associated with it) and the other chain will become a secondary chain. This potential disagreement among

[15] Ethereum, an alternative to Bitcoin, is discussed further in Section 3.3.1.

nodes about which block represents the tail of the chain opens up the possibility of a double-spend attack. This, along with other potential blockchain vulnerabilities, is discussed in Section 3.4.

3.3 ALTERNATIVE BLOCKCHAIN BASED FRAMEWORKS

Since the introduction of the Bitcoin blockchain, several alternative blockchain technologies have emerged. While similar, they have slightly modified structures and offer a series of trade-offs compared to the original Bitcoin blockchain. With the development of alternative blockchains, a number of alternative cryptocurrencies were created that make use of their respective blockchain technology, for example, Litecoin[16] and Ethereum's[17] 'ether'. Ethereum will be discussed further in this section due to its widespread adoption, power, and applicability. Furthermore, other blockchains exists that do not rely on a proof of work protocol, but instead make use of a newer, 'proof of stake' model that does not rely on computations to secure the integrity of the system; this is discussed further in Section 3.3.2.

3.3.1 Ethereum

Ethereum is a decentralised, blockchain based, platform that can not only transfer cryptocurrency, but also run 'smart contracts' – applications that are designed to run without interference and based on predefined, immutable conditions [57]. These smart contracts are built using Ethereum's object-oriented programming language, 'Solidity'[18]. The exact technical implementation differences between Ethereum and Bitcoin will not be discussed, but this section will instead focus on the practical differences.

Ethereum was proposed in late 2013, with the system going live approximately two years later [58]. The Ethereum white paper, published by Vitalik Buterin, describes in detail both the technical design, as well as the rationale behind the Ethereum protocol and smart contract architecture [59]. Ethereum smart contracts are run using the Turing-complete 'Ethereum Virtual Machine (EVM)' runtime environment, first introduced and formalised by Ethereum co-founder, Gavin Wood in 2014 [60]. Rather than provide users with a set of pre-defined operations (for example, in the case of Bitcoin, bitcoin transactions), Ethereum provides a platform to allow users to create their own operations, thus greatly expanding the scope of applications compared to the Bitcoin blockchain.

The value token of the Ethereum blockchain is known as 'ether' and is often traded as a unit of value, much like bitcoin. Another key component of the framework is that of 'gas' – a special unit used in Ethereum for ensuring an appropriate fee is paid for every transaction submitted to the network. Every operation that can be performed on the Ethereum network has a gas cost associated with it, as too does the byte size of the transaction. This ensures that the amount of gas required for large, complex transactions is commensurate to the amount of work being performed by the network to perform the operations required for that transaction. The creator of the contract specifies the amount of ether he is willing to pay for each gas unit used, as well as a maximum amount of gas (any transaction that exceeds this maximum is halted, but the current accumulated fee is still applied). Once a transaction has been processed by the network, the user is charged

[16] https://litecoin.com/
[17] https://www.ethereum.org/
[18] https://solidity.readthedocs.io/en/develop/

based on the total amount of gas and the price he was willing to pay per gas unit. The cost of these operations is measured in gas, rather than ether, as the price of ether is sensitive to market fluctuations, thus the system allows one to decouple market fluctuations in the value of ether from the cost of operations.

3.3.2 Proof of stake blockchains

While the standard method of achieving distributed consensus on the blockchain makes use of proof of work methods (as discussed in Section 3.2.3), an alternative approach involves choosing the creator of the next block in a deterministic fashion. In this system, the probability of a user being chosen to mint[19] the next block is proportional to their stake in the network (the amount of currency that account has invested in the network). This concept, known as 'proof of stake', was first discussed among the Bitcoin community in 2011 [61] and independently conceptualised by King & Nadal [62].

In proof of stake consensus mechanisms, the owners of a digital currency effectively lock up or 'bond' a part of their stake in the network and in some sense, use it as collateral to vouch for the next block, with the chain with the most collateral behind it being chosen as the main chain. The collateralised stake acts an incentive to act honestly, as dishonest users will be punished by losing access to their collateralised stake. There are three major advantages of such a system compared to the alternative proof of work mechanism:

1. Reduced resources are required to mint new blocks.
2. Lower transaction fees.
3. Increased protection from a malicious attack on the network.

Point 1 follows from the fact that proof of stake negates the need to contribute resources to the mining of blocks, as there is no longer a computationally expensive hashing requirement, which expends energy and hence money. This is the motivation behind the second advantage: any transaction fee must be greater than or equal to the opportunity cost of a miner and hence must be commensurate to the amount (and hence cost) of computational resources provided by the miner; if the expected cost of mining a block is less than the expected transaction fees that a miner would receive for that block, he would contribute resources to mine the block. Hence when the cost of creating new blocks is reduced (or in the case of minting, virtually zero) transaction fees can be reduced.

The final advantage, point 3, stems from two sources: (1) executing a consensus based attack is costly as it requires the malicious user to hold a considerable stake in the network[20] and (2) incentives for these attacks are greatly reduced as a malicious user with sufficient stake in the network to perform an attack would likely end up greatly reducing trust in the network and hence reduce the value of their stake. This discourages centralised mining cartels (that defeat the idea of distributing the consensus mechanism across independent nodes) from forming.

[19] Mining in proof of stake networks is often referred to as 'minting' or 'forging'.

[20] Here, an attacker would need to hold more than 50% of the digital currency to guarantee a successful consensus based attack (rather than 50% or more of the mining power). Consensus based attacks are discussed in Section 3.4.

Several blockchains exists that make use of the proof of stake consensus mechanism, including: Peercoin[21], BitShares[22], and Nxt[23], with Ethereum looking to adopt a new algorithm, Casper, that makes use of the proof of stake concept [63].

3.3.3 Private vs public blockchains

Blockchains can either be fully decentralised and open to any user, anywhere in the world, or they can have certain restrictions imposed on them, bringing them closer to a centralised system. These differences are discussed in the following subsections.

3.3.3.1 Public blockchains

A public blockchain is accessible to any user; all transactions and blocks are visible to anyone in the world. For example, any block on the Bitcoin blockchain can be inspected by using a publicly available block explorer[24]. Furthermore, anyone can contribute to the consensus protocol of determining which blocks get added to the blockchain, and what the current state of the blockchain is. The major advantage of using a public blockchain is that it is often well established and used by millions of users, resulting in considerable distributed mining power on the network. This is advantageous as it negates the need to run and administer one's own network and mining pool; the network itself takes care of this. Furthermore, the large-scale and distributed mining power makes the network less susceptible to manipulation by adversaries. An organisation or government that wishes to alter the blockchain to manipulate votes needs to overpower the entire global honest network with computational power – a virtually impossible task.

A further advantage of public blockchains is that they limit the impact developers can have on the network and in so doing, protect the users. This is achieved by restricting the ability of the developers to alter the contents of the blockchain, as well as the rules governing the blockchain, in turn creating greater trust in the network by the users of the system.

3.3.3.2 Consortium blockchains

Consortium blockchains have their consensus process controlled by a pre-selected set of nodes. In such situations, it may be required that a minimum number of nodes must sign a block for it to be validated. Read restrictions may be either public or private. These blockchains are essentially a hybrid between the fully decentralised trust model of public blockchains and the single, highly trusted, entity model that exists with private blockchains.

3.3.3.3 Private blockchains

Private blockchains, unlike their public counterparts, allow one to create a system where permissions are more tightly controlled. Rights to modify or view the blockchain can be restricted to a specific set of users, while still maintaining many of the guarantees of authenticity and decentralisation provided by the blockchain (although these guarantees may not be as strong) [50]. Private blockchains operate completely independently of a public blockchain, and mining power would have to be recruited or provided by the

[21] https://peercoin.net/
[22] https://bitshares.org/
[23] https://nxt.org/
[24] https://blockchain.info/

organisation managing the blockchain. Buterin [50] lists several advantages of private blockchains over their public counterparts, with the major advantages being:

1. The ability to restrict user access to the blockchain and thus keep information confidential.
2. The blockchain owner is not bound to a set of rules defined by the specific blockchain instantiation, and can thus, for example, alter the structure of transactions, revert transactions, or modify balances.
3. The miners can be restricted to a set of known and authorised nodes, thus negating any risk of a 51% attack from miner collusion.
4. A smaller network of well-connected nodes, with the ability to manually fix faults quickly. This allows for consensus algorithms to offer finality[25] of a block much faster.

While the distinction between private and consortium chains is not perfectly clear, consortium chains essentially provide a hybrid between the highly decentralised and distributed trust system of public chains, and the more concentrated trust model of private chains. Private chains instead more closely follow the standard, centralised system's model, with an added degree of cryptographic audibility attached. However, there is little evidence to suggest that the optimal means of authentication should consist of a series of hash-linked blocks; as an alternative, Buterin [50] argues that generalised zero-knowledge proof technology[26] provides users with a broader and more exciting array of opportunities with regards to the cryptographic assurances that applications can provide.

3.4 BLOCKCHAIN VULNERABILITIES

While the consensus mechanism described in Section 3.2.4 provides a means for distributed consensus to be reached, it is theoretically vulnerable to attack by miners who use their hashing power for dishonest means. By controlling a significant share of the mining power of a blockchain network, a miner (or pool of miners) can attack the consensus mechanism, disrupting both integrity and accessibility. One attack scenario against the consensus mechanism is known as the 51% attack or consensus attack [51]. Despite its name, these consensus attacks can be attempted with less than 51% control of the total hashing power, with 51% merely forming the threshold where these attacks are almost guaranteed to succeed.

A number of these vulnerabilities are discussed in the following subsections. The application of these attacks to blockchain based voting are discussed in Section 4.6, after the system has been introduced.

[25] 'Finality' refers to the fact that once an operation is completed and committed, it cannot be reverted. This is a property that we often associate with centralised systems, however, it can be provided either explicitly, probabilistically, or not at all in decentralised systems (such as the blockchain). Technically, any proof of work blockchain never allows a transaction to be fully finalised – there is always the possibility that someone can fork a longer chain (see Section 3.4.1). For a good discussion on finality, see [74].

[26] Zero-knowledge proofs (introduced in [71]) convey no additional knowledge other than the correctness of the proposition in question. These cryptographic methods can be used for proving/verifying, in zero knowledge, the integrity of computations (see [75, 76] for a theoretical discussion about these protocols).

3.4.1 Double-spend attack

In this scenario, if miners control enough of the network's hashing power (through collusion or independent means) they can work to force forks in the network. This can be exploited to allow a malicious user to continually issue and rescind transactions and is referred to as a 'double-spend attack'. A transaction can effectively be removed from the blockchain by forking the network at a point prior to the inclusion of the transaction (thus invalidating previously confirmed transactions) and extending the new fork such that it become the longest competing chain. The double-spend attack can only be applied to the attacker's own transactions (i.e. those that he can produce a valid signature on).

3.4.1.1 An example of the double-spend attack

A double-spend attack is illustrated in Figure 2, where the initial state of the blockchain is depicted in part (a) of the illustration. Here, the current main chain is shown by the bold arrows and is represented by the green blocks. Note that the main chain can also have secondary chains associated with it (represented by grey blocks). In part (b), Mary sends John 10 bitcoins in exchange for a product. This transaction is successfully mined and honest nodes continue extending the main chain with yellow blocks. Meanwhile, Mary secretly begins mining a fraudulent branch that instead shows a 10 bitcoin transaction to herself, represented by the red blocks, and in part (c) she continues to extend her chain. Once John is satisfied that a sufficient number of blocks have been added on top of Mary's transaction to him, he sends Mary the product.

Eventually, when the proof of work associated with the fraudulent branch is greater than the honest one, Mary publishes the branch. Since Mary's chain is longer than the currently accepted main chain, it now becomes the accepted main chain of the network, as illustrated in part (d). Thus, the attacker has successfully managed to invalidate previously confirmed blocks by forking below them and having the network converge on the alternative red chain.

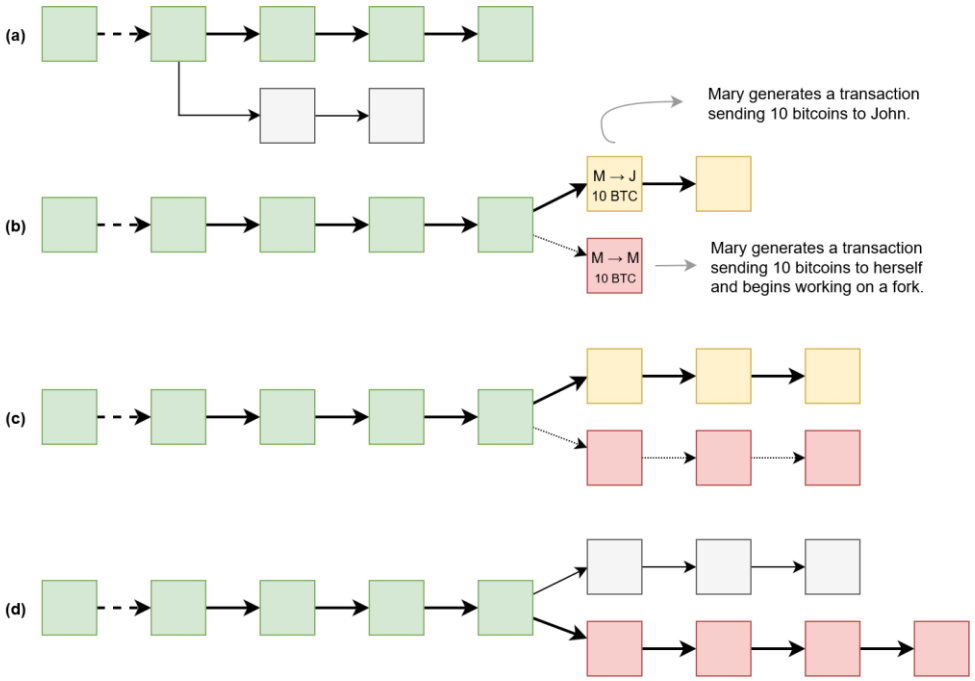

Figure 2 - A double-spend attack in which Mary is able to rescind a previously confirmed transaction by secretly producing a fork (red) that has more proof of work associated with it than the chain containing the correct transaction (yellow).

3.4.1.2 An analysis of the double-spend attack

The original Bitcoin whitepaper [47] considered the scenario of an attacker trying to generate an alternative chain in order to perform a double-spend attack. It was shown that as the depth of a required fork increases, it becomes exponentially more difficult to force a fork, making old blocks practically immutable.

Probabilistically, we can model an attempted double-spend as a random walk – a stochastic process that describes a path that consists of a succession of random steps on a mathematical space. For example, we may view a biased coin that lands on heads (with probability q) as a move forwards ($+1$) and an extension of the attacker's fork; and tails (with probability p) as a move backwards (-1) and an extension of the honest chain, as depicted in Figure 3.

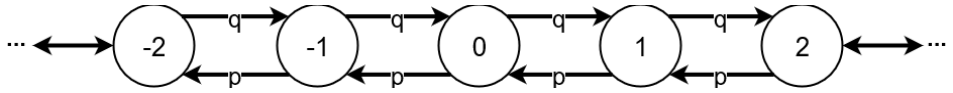

Figure 3 - The structure of a double spend attack is analogous to the toss of a biased coin. However, instead of heads or tails, we have either an extension of the attacker's fork (with probability q) or the honest chain (with probability p).

The probability of an attacker catching up from a given deficit is analogous to a slight variation of the gambler's ruin problem[27]: a gambler with unlimited credit starts from a deficit and plays potentially an infinite number of trials to reach breakeven. The probability he ever reaches breakeven is analogous to the probability of an attacker catching up with the honest chain, and is given by:

$$Q_z = \begin{cases} 1, & p \leq q \\ \left(\frac{q}{p}\right)^z, & p > q \end{cases}$$

where:

$q =$ mining power controlled by the attacker

$p =$ mining power controlled by honest nodes

$Q_z =$ probability of ever catching up from a deficit of z blocks

and $p + q = 1$

This equation was first introduced by Nakamoto [47], with the derivation being described by Ozisik & Levine [64]. As an example, consider the case where an adversary controls one quarter of the total hashing power of a blockchain network. The probability of him successfully computing a fork that has more proof of work than the current main chain (i.e. $z + 1$) from a deficit of 6 blocks is given as follows:

$$\left(\frac{0.25}{0.75}\right)^{6+1} \approx 4.57 \times 10^{-4}$$

From the above, we see that the probability of successfully forcing a fork on a proof of work blockchain decreases exponentially with the number of blocks added above the required fork location, and even when an attacker controls 25% of the network, blocks

[27] The earliest known mention of the gambler's ruin problem is a 1656 letter from Blaise Pascal to Pierre de Fermat [81].

quickly becomes practically immutable as more and more blocks are appended to the chain.

3.4.2 Denial of service attack

In addition to the double-spend attack, the other scenario for a consensus attack involves using mining power to simply ignore specific transactions by not including them in blocks to be mined [51]. If the transactions are mined by another miner, the attacker deliberately forks the blockchain, ensuring the transaction is not included in the fork.

3.4.2.1 An example of a denial of service attack

Assuming Alice wants to transact with Bob, Alice submits a signed transaction with the destination of the transaction being Bob's blockchain address. This transaction enters the unconfirmed transaction pool until a miner includes it in a block and successfully mines the block. If an adversary, Eve, is a blockchain miner who does not want the transaction to succeed, Eve never includes that transaction in a block she is mining. If another miner successfully has the transaction confirmed into the blockchain, Eve would then fork the blockchain prior to that transaction's inclusion – this requires Eve to control a considerable portion of the total hashing power of the network.

3.4.3 Sybil attack

A Sybil attack occurs when an attacker attempts to fill the network with hostile nodes controlled by the attacker. The Sybil attack was introduced in 2002 in a widely cited paper by Douceur [65] (long before the introduction of the blockchain) that demonstrated the vulnerability of large-scale peer-to-peer systems to faulty or hostile remote computing elements. The authors showed that: "without a logically centralized authority, Sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities."

By isolating a node from the honest network, an attacker could implement a denial of service attack by refusing to relay blocks, thus disconnecting an honest node from the honest network. Alternatively, an attacker could effectively place an honest node onto a separate network by only relaying blocks created by the adversary, leaving the honest node vulnerable to a double-spend attack (as discussed in Section 3.4.1).

4 A BLOCKCHAIN BASED ELECTRONIC VOTING SYSTEM

The following section describes the implementation of a blockchain based voting system. In this section, the steps taken to derive the final protocol and implementation are discussed, along with the issues (and solutions to these issues) encountered along the way. While prior blockchain voting implementations have already been introduced in Section 2.3, these will be further referenced and analysed for effectiveness and security. By evaluating prior models, along with their respective weaknesses and strengths, an improved blockchain voting protocol can be developed. A systematic comparison of some of the prior blockchain based voting systems to the protocol described in this section is provided in Section 4.7.

4.1 A SIMPLE BLOCKCHAIN VOTING MECHANISM

Perhaps one of the simplest implementations of using the blockchain for voting is the method proposed by Lee et al. [10]. While this system has been briefly described in Section 2.3, a brief recap is provided: in this method, each candidate is assigned a public blockchain address with the number of votes a candidate has received being simply defined as the number of transactions sent from authorised voters[28] to the candidate's blockchain address. The following subsections will analyse and expand upon this method.

4.2 THE CHOICE BETWEEN PRIVATE AND PUBLIC BLOCKCHAINS

As discussed in Section 3.3.3, there are essentially two major blockchain implementation methodologies: public and private (if we consider consortium chains to be a form of private chains). While the general differences between private and public blockchains (along with their respective advantages and disadvantages) has been discussed previously, it is necessary to analyse these two choices with specific reference to this blockchain voting mechanism.

4.2.1 Using a private blockchain

If a private blockchain were to be used, mining power could either be provided by any user of the system, or only a set of authorised miners. However, even if a considerable monetary reward for block mining were offered, it would be unwise and unsafe to not restrict mining to a predetermined set of authorised nodes. Lee et al. [10] discusses offering a reward of $100 000 per block mined and not requiring miners to be authorised, under the belief that this would attract sufficient mining power to secure the network. This would result in an estimated total mining cost of $7.2 million for a 12-hour election, assuming blocks are mined every 10 minutes. However, the authors are incorrect when they state, "That amount of incentive is likely to bring more than enough contributors to provide processing power to keep the block-chain [sic] from being forged." The reason is simple: the incentive for a potential miner to contribute resources is if his expected reward $E(r)$ is greater than the individual cost[29] c incurred by the miner. The network will

[28] A transaction can be seen as originating from an authorised voter if the public key of the digital signature used to sign the transaction is contained in a list of authorised public keys for that particular vote. This is discussed in detail in Section 4.4.

[29] This cost comprises of any cost involved in mining, including equipment and energy costs, as well as the opportunity cost involved.

eventually converge to a point where $E(r) = c$ for all miners in the network. There are three cases for each miner:

(1) $E(r) > c$
(2) $E(r) = c$
(3) $E(r) < c$

In case (1) the expected reward is greater than the cost incurred to a potential miner, and hence he will enter the mining pool, with the opposite being true for case (3). In case (2) a potential miner will be indifferent to entering the network.

Case (3) is interesting, as for an adversary the reward is not just the expected monetary value he will earn, but also includes the added benefit of altering the outcome in their favour. Hence, one may end up with the case where honest miners have no incentive to join the network, but due to the higher expected reward of dishonest miners, they will be incentivised to join. Eventually a point will be reached where the total reward ($7.2 million) will equal the total cost of running the network and in this case a single adversary would have to spend just over $3.6 million to control over 50% of the network and almost certainly be able to successfully implement attacks such as the denial of service attack (as discussed in Section 3.4.2) and prevent votes from being cast.

Overall, however, the issue of having to pay people to mine blocks on a private blockchain in order to run an election seems counter-intuitive. This is especially true when one considers the alternative – a public blockchain that does not require direct payment to miners.

4.2.2 Using a public blockchain

If one considers the case of transferring bitcoins (or some other cryptocurrency or 'vote transaction') to a candidate's wallet as being equivalent to a vote, using a public blockchain would mean that the act of voting could incur a monetary cost (a transaction fee and the value transferred). However, the cost per a vote would be very low as one needs to only transfer the minimum transaction value allowed by the chosen blockchain framework, as well as a small transaction fee. While the transaction fee is not a necessary requirement, including a transaction fee (or appropriate gas rate in the case of Ethereum) that is proportional to the byte size of the transaction will prevent significant delays in having the transaction mined and included in the blockchain.

While using a public blockchain is recommended from a security and accountability perspective, it does remove some control from the voting authority. Further to this, a public blockchain would allow access to anyone, even non-voters. Thus, one would be required to filter out vote transactions from unauthorised voters (see Section 4.5.3.3 for a discussion on how votes will be tallied). Furthermore, authorised voters would be able to submit votes remotely, which could lead to votes being conducted under duress or through coercion (see Section 4.3). Another problem that arises from the use of public blockchains is that anyone can inspect the transactions within a blockchain block, thus it would be simple for anyone who knows the candidates' blockchain addresses to have a rough estimate of the election tally as it progresses by inspecting only the blocks that have been mined during the election and looking for likely vote transactions.

4.2.3 Choosing between a public and private blockchain

The question of whether or not a public or private blockchain should be used is fairly subjective. While the previous subsections have stated the relevant advantages and disadvantages of both approaches, the final choice should be conducted based on the requirements of the election. Essentially, the voting authority should weigh up the associated risks and benefits of each approach, and based on the requirements of the election, choose the best approach. In the case of small, non-critical elections, a private blockchain would be easier and cheaper to manage. In the case of large-scale, security critical elections, a public blockchain that guarantees strong levels of security and integrity should be chosen.

For the remainder of this section, it will be assumed that a public blockchain is used as this approach more closely follows the aim of this paper in producing an open and trusted voting system.

4.3 Is remote electronic voting possible?

It should be clear that given a risk-free environment (i.e. one free from malicious actors), providing voters with the ability to vote remotely is beneficial. Voters would simply have to run any required voting software on their computer (or any compatible and Internet connected electronic device). A voter would then be required to authenticate themselves on the device before submitting their vote. While the exact implementation details of this authentication are not discussed, it will be shown that even if authentication is guaranteed to be correct and secure, this system is still not a viable option for any critical or public election. While the ability to vote remotely adds convenience, such a system is only plausible in the risk-free environment. Once malicious actors are included in the system, the system becomes open to vote buying and coercion.

As discussed in Section 2.1, one of the key requirements of an electoral system is that voters can vote free from duress or influence. The problem that arises if voting is possible in privacy, is that it is easy to sell one's vote or be forced to vote under duress, with the malicious actor being able to witness the way the voter voted. Furthermore, an honest voter has no means to ensure the integrity of their device has not been compromised, and this could lead to the voter thinking they voted for candidate A when in fact the vote is for candidate B. This specific blockchain implementation would however allow the voter to view that their vote was recorded as cast by allowing them to inspect the blockchain. However, a problem arises when a compromised system leads to a vote being cast for an incorrect candidate and the voter thus has to try have that vote declared void – this is discussed further in Section 4.5.3.5.

While work on developing coercion resistant remote voting systems has been carried out (for example, [23, 66]), these implementations often have unrealistic assumptions and offer no trivial adaption to be suitable for the blockchain based system proposed in this section. For example, one possible approach, similar to the method adopted by Estonia and discussed in Section 2.2.4, is to allow for arbitrarily many re-votes until the voting period closes. Thus, if a voter is coerced, he can simply change his vote at a later stage.

An additional problem with remote Internet voting is that it provides attackers with direct access to the voting system and the ability to attack the system remotely. An example of this was demonstrated by Wolchok et al. [67]: in 2010, Washington D.C. developed an

Internet voting pilot project with the aim of allowing overseas absentee voters to cast their ballots using a website. Prior to the election, a mock election was held in which anyone could test or attempt to compromise the system. Within just 48 hours of the system going live, the authors were able to gain near-complete control of the system, allowing them to change any vote and reveal almost every secret ballot.

A model developed by Van Acker [68] demonstrated that under certain conditions, vote buying is not profitable and hence remote elections can be held. However, these conditions tend to occur only when the election is fairly insignificant, which is almost certainly not the case for public elections where national interests are at stake. Thus, the issue of voter coercion and the inability to guarantee that an election has been conducted freely and fairly suggests that this blockchain based voting implementation should not be conducted remotely for any significant elections.

If a public and unrestricted blockchain is used, it would be difficult to prevent remote voting from being possible. This is because anyone would be able to submit vote transactions from a remote location, so long as they are signed by a signature that has had its public key authorised. This problem is solved by restricting voter access to key information that would allow them to vote remotely and is discussed in Section 4.5.1.2. The following section discusses how a list of authorised public keys can be developed without linking them to a voter's identity.

4.4 AN ANONYMOUS VOTER REGISTRY

One of the key requirements of voting systems is that they provide anonymity: no one should be able to deduce who a specific voter voted for, or even if an individual has voted. A further extension to this, to prevent votes from being bought, is that one should not be able to reveal a cast vote to someone else. These anonymity requirements are satisfied via physical ballot systems where no association is made between the voter and their cast ballot. Furthermore, the vote is also receipt free, meaning the voter has no record of their vote that could be used to convince others of a specific vote. It is, however, more difficult to provide these guarantees with electronic systems, especially those that utilise a publicly auditable ledger, such as the blockchain.

4.4.1 Voter authorisation techniques

When using transactions to represent votes, the problem encountered is that these transactions are digitally signed and hence the voting authority would have to maintain a list of authorised public keys (associated with authorised voters). This list would allow the voting authority to filter out unauthorised votes cast by non-authorised voters (who try to trick the system by creating transactions into candidates' wallets) during tallying. While keeping the link between a candidate's identity and his assigned blockchain address secret during the election, and withholding vote transactions until after the election has closed, may well ensure only valid voters have been able to vote (see Section 4.5.1.2), an additional layer of security is required to guarantee that only votes from authorised voters are counted. There are three possible means of adding this additional layer of authorisation:

(1) Simply link a voter's blockchain address to their record in the voting registry.
(2) Make use of a third party.
(3) Have a voter prove that they are authorised, without revealing their identity.

The first solution, while easy to implement, is flawed as it creates a direct link between a voter's identity and his blockchain address. This would allow any breach of the database to make it possible for an adversary to view who a voter voted for, thus failing to satisfy the anonymity requirement.

The second solution is the approach employed by Lee et al. [10] where the notion of a trusted third party is introduced. The role of this third party is to act as an intermediary between the voter and the voting authority. The system works as follows: During voter registration (with an authenticating agency), a voter selects a secret message and hashes the message using some pre-defined hashing protocol (e.g. SHA-256). This hash is then linked to the voter's identity by the authenticating organisation. The voter also maintains a public/private key pair and a public Bitcoin wallet based on the Bitcoin protocol. It is the responsibility of voter to keep the private key and secret message secure.

During voting, the voter submits the secret message's hash to the trusted third party, who forwards this value to the authenticating organisation. The organisation's reply simply indicates whether or not this hash exists in their voter registry. If the hash is confirmed to exist, the third party asks the voter to submit the message that generates the hash to ensure that the preimage of the hash is known and that the hash was not obtained via a database breach. Once this is confirmed, the voter's Bitcoin wallet address[30] is appended to a database controlled by the third party. This method is used to generate a list of valid voter addresses, with the most recent transaction from an address contained in the list to a candidate's wallet being counted as a valid vote. At no point is any personally identifiable information submitted via the third party to the authenticating organisation, or vice versa.

Finally, an implementation using the third solution of having a voter authenticate themselves without revealing personally identifiable information, was proposed by McCorry et al. [9]. This implementation differs from that of Lee et al. [10] and other implementations described in Section 2.3 in that it does not require a trusted third party and does not make use of blockchain transactions to vote. Instead, the blockchain is merely used as a 'public bulletin board' to store the voting data, with Ethereum being used to enforce the voting protocol's execution. Their protocol works as follows:

1. All voters, $P_1, P_2, \dots, P_n$, agree on a finite cyclic group[31] G of prime order q, as well as a generator g of G.
2. Every voter P_i selects a random $x_i \in \mathbb{Z}_q$ and broadcasts g^{x_i} along with a non-interactive[32] zero-knowledge proof to prove knowledge of x_i.
3. All voters verify the validity of the zero-knowledge proofs and compute a list of reconstructed keys:

$$Y_i = g^{y_i} = \frac{\prod_{j=1}^{i-1} g^{x_j}}{\prod_{j=i+1}^{n} g^{x_j}}$$

[30] A Bitcoin address is a 160-bit hash of the public portion of a public/private Elliptic Curve Digital Signature Algorithm (ECDSA) keypair.

[31] G should be a group where the Decisional Diffie-Hellman (DDH) problem is intractable: given g^a, g^b, g^c for uniformly and independently chosen $a, b, c \in \mathbb{Z}_q$, there is no efficient algorithm to distinguish the two distributions (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) (see [77]).

[32] A non-interactive zero-knowledge proof, in contrast to the standard interactive variant, does not require the verifier to issue challenges to the prover.

4. A voter sets v_i to 1 for a 'yes' vote and 0 for a 'no' vote and broadcasts $g^{x_i y_i} g^{v_i}$ along with a non-interactive zero-knowledge proof to prove $v_i \in \{0, 1\}$.
5. Once the final vote has been cast, compute $\prod_i g^{x_i y_i} g^{v_i}$. Now, since $\prod_i g^{x_i y_i} = 1$ (see [69]), and the number of voters is a relatively small value, the discrete log of $g^{\sum_i v_i}$ can be calculated via exhaustive search.
6. $\sum_i v_i$ reveals the number of yes votes.

There are several issues with this implementation:

- Voting is conducted in an unsupervised manner and hence does not offer any coercion resistance: a voter may vote under the direct duress of a coercer.
- The last voter to publish their vote can compute the current election tally.
- If the last voter is dissatisfied with the tally, he can withhold his vote, preventing the final tally from being computed.[33]
- The current implementation is limited to a yes/no vote only.
- Every voter must submit their vote in order for the election tally to be computable.
- Given the above disadvantages, this system is only suitable for small-scale, insignificant votes.

4.4.2 Voter authentication with anonymity

When a voter participates in a physical ballot election, he is required to authenticate himself by providing authentication information to an authentication agency. Only once he has been authenticated is he able to enter the voting booth and submit a vote. Thus, this pre-vote authentication guarantees both identity authentication (that the voter is who he says he is and is permitted to vote) and anonymity (once the voter's identity has been authenticated, there is no means of linking his identity to the cast ballot). In the context of electronic voting, it is easy to guarantee either identity authentication or voter anonymity, however, guaranteeing both simultaneously becomes a much harder problem – a voter who authenticates their identity while using an electronic voting system is providing a point for a malicious agent to couple the voter's identity to the actual cast ballot, but if this authentication is not done, the potential for voter fraud arises.

In this specific implementation, identity authentication will be provided for by having an authorised voter prove ownership of the public key associated with the digital signature that the voter will use to sign his vote transaction. The public keys of all authorised voters will be used as the official voting registry – any public key on this list will be assumed to belong to a voter who has been authenticated by the voting agency as a valid voter. Only vote transactions to a candidate's wallet that are signed with a digital signature, where the public key exists on this list, will be counted. This guarantees that these transactions are from authorised voters, since only they know the associated private key used to sign the transaction.

The above method solves the problem of voter authentication, but does not guarantee anonymity since a direct link could be established between a voter and a public key. There

[33] The last voter issues can be addressed by having all voters commit to a vote by storing a hash of their encrypted vote (prior to submission) on the blockchain as a commitment. Alternatively, an election administrator can cast the final vote which is then excluded from the tally, as discussed in [78].

are two modifications to this method that can add the additional anonymity requirement: blind signatures and zero-knowledge proofs.

4.4.2.1 Blind signatures

A public key used to sign vote transactions can be considered valid if the key itself is signed by the voting authority, under the assumption that the voting authority would have only signed a public key if it belonged to an authorised voter. The problem with the naïve signing approach of having a voter authenticate themselves and then submit their key for signing, is that it makes it possible for the voting authority to generate a link between the voter's identity and their public key. Thus, we need a signature scheme that prevents the signer from being able to link a message it signs to the owner of the message. Fortunately, this can be accomplished through the use of blind signatures.

Blind signatures, first introduced in a seminal paper by David Chaum [70], are a form of digital signatures that allows a person to have a message digitally signed by another party without revealing any information about the message to the signing party. This is achieved by having the signing party sign a variant of the original message; the original message is cryptographically modified in such a way that the new message leaks no information about the original message. Only the message owner is then able to modify the signature such that it forms a valid signature on the original message.

Chaum [70] demonstrated the implementation of this concept, using RSA signatures, as follows: Assume Alice has a message m that she wishes to have signed by Bob, however, Alice does not want to leak any information about m to Bob in the process. Now, let (n, e) and (n, d) be Bob's public and private keys respectively. Alice then generates a random value r that is coprime to n (in other words, $\gcd(r, n) = 1$) and obfuscates the message by computing the product of m and a 'blinding factor' $r^e \bmod n$:

$$m' = (r^e m) \bmod n$$

Alice then sends m' to Bob to be signed. Since m' has been 'blinded' by the random value r, Bob can derive no useful information from it.[34] Bob then signs the blinded message, creating the signature s' as:

$$s' = (m')^d \bmod n$$

It is important to note that currently the signature s' is a valid signature on the blinded message m', but not on m. Furthermore, while Bob knows the value m', he is unable to learn any information about the true message m. Now, since:

$$(m')^d \equiv (r^e m)^d \equiv r m^d \pmod n$$

Alice can now obtain the true signature s of m by computing:

$$s = r^{-1} s' \bmod n$$

This is true since:

$$r^{-1} s' \equiv r^{-1} (m')^d \equiv r^{-1} (r^e m)^d \equiv r^{-1} r^{ed} m^d \equiv r^{-1} r m^d \equiv m^d \pmod n$$

[34] As r is a random value and the mapping $r \mapsto r^e \bmod n$ is a permutation, it follows that $r^e \bmod n$ is random and hence m' does not leak any information about m.

To utilise this protocol, a voter would generate their public/private key pair that conforms to the public key cryptosystem used by the blockchain chosen for voting. The voter would then implement the protocol described above to have the voting authority generate a valid digital signature on the voter's public key (without revealing the public key to the voting authority). The problem that now arises is that the voting authority does not know what the original public key is and hence the voter must anonymously submit the new digital signature and public key to the voting authority at a later stage. As the digital signature proves the owner of the public key has already been authenticated, one does not have to worry about authentication prior to submission of the digital signature and public key to the voting authority.

4.4.2.2 Using zero-knowledge proofs

Zero-knowledge proofs (first formalised by Goldwasser et al. [71]) allow the prover to verify to another party that a given statement is true (or in this case, that a digitally committed value is a member of a given public set), without conveying any information other than the fact that the statement is indeed true. When the statement consists only of the fact that the prover possesses certain knowledge, it is a special case known as a 'zero-knowledge proof of knowledge'. It is trivial to prove that one possesses certain information by simply revealing that information; the challenge involves proving the possession of knowledge without revealing anything that would help the verifier (or an attacker) from learning what that knowledge is – hence the need to distinguish between a 'proof of knowledge' and a 'zero-knowledge proof of knowledge'.

A zero-knowledge proof of knowledge, coupled with the use of a cryptographic accumulator[35], could be used to prove that a voter is authorised to vote. Camenisch & Lysyanskaya [72] provide an implementation of an accumulator and protocol that allow for the prover, knowing an element and corresponding witness, to prove to a verifier (in zero-knowledge) that he is privy to an element contained in the accumulator. Using the techniques discussed by Camenisch et al. [73], one can develop a dynamic accumulator[36] that allows for an efficient zero-knowledge proof of knowledge of an item included in the accumulator. The outline of how this could be applied to anonymous voting is given as follows:

1. During voter registration, the voter (upon successful authorisation) submits a secret key ϕ_i to the voting authority.
2. The voting authority accumulates ϕ_i into an accumulator.
3. During voting, the voter anonymously performs a zero-knowledge proof of knowledge of his secret key, proving that he knows a value that has been included in the accumulator, and hence that he is an authorised voter.
4. The voter now anonymously submits his public key to the voting authority who adds it to the list of authorised public keys.

The technical details of the exact implementation of the accumulator and zero-knowledge proof of knowledge are provided by Camenisch & Lysyanskaya [72], with additional

[35] A cryptographic accumulator (first introduced by Benaloh & de Mare [82] in 1993) is an algorithm that compresses a list of elements into a single accumulated value such that there exists a witness that attests to the fact that a given input was indeed incorporated into the accumulator.

[36] A dynamic accumulator allows for $O(1)$ insertion and deletion of elements from the accumulator.

extensions to allow for an efficient set-membership proof being described by Camenisch et al. [73]. It is important to note that the above protocol does not allow the voting authority to associate the public key with a specific voter; the voter has performed the set-membership proof in zero-knowledge, and hence the voting authority has no way of determining which ϕ_i the voter knows, only that he does indeed know a ϕ_i that was included in the accumulator.

4.4.2.3 Blind signatures vs zero-knowledge proofs

Both blind signatures and zero-knowledge proofs provide a means for the voting authority to generate a list of authorised public keys, without associating these keys with the identity of the voter. However, for the remainder of this paper, the blind signature approach will be assumed for brevity; while both schemes give similar guarantees, the blind signature approach is arguably simpler to understand and implement.

4.5 THE ELECTION PROCESS

In this section, the general steps that would need to be followed to implement a blockchain based electoral system are described.

4.5.1 Pre-election

The requirements described in this subsection are those that would need to be implemented prior to the opening of voting.

4.5.1.1 Voter blockchain addresses

One of the first requirements is to have each voter generate their own address on the blockchain that can be used as a source for generating the vote transaction. In so doing, the voter would generate their private key along with the associated public key. It would be necessary to have a small amount of the relevant cryptocurrency associated with the address. This would likely need to be added by the voter but could be reimbursed later.

4.5.1.2 Candidate blockchain addresses

Each candidate should be assigned a unique blockchain address that will be used as the destination for vote transactions in favour of that candidate. The mapping of a candidate's identity to his blockchain address should be kept secret until voting has closed. This is necessary to prevent voters from being able to vote remotely or attempt to selectively exclude votes to specific candidates through a denial of service attack (this is discussed further in Section 4.6.4); the actual candidate addresses could be made public prior to the election, but the candidate with which that address is associated should be kept secret. While this would make it possible for an observer to see the distribution of votes during the voting period, this is prevented by only releasing vote transactions to miners once voting has closed (see Section 4.5.3.1).

The voting authority would be responsible for establishing and linking blockchain addresses to candidates, as well as keeping these links private until after the voting has closed. Further to this, the voting authority should publish a cryptographic commitment to this link. This commitment will be used as proof (once voting has closed) that the link was established prior to the start of voting.

4.5.1.3 Voter registration

Once voters have their own blockchain addresses, the voting authority would need to develop the electoral roll – a list of persons who are eligible and registered to vote. During voter registration, a voter would have his public key added to the list of authorised public keys using one of the methods described previously in Section 4.4.2.

4.5.1.4 Voting machines

Section 4.3 demonstrated that remote electronic voting is not secure for large-scale and high-stakes elections. As such, voting in large-scale elections should be conducted via electronic vote recording devices. These devices would have to allow for the voter to use their public/private key pair to generate a digital signature on their vote transaction and submit their vote transaction to the blockchain. The association between candidate addresses and candidate identities would have to be stored securely on these devices. The methods required to secure the voting machines is beyond the scope of this paper and is left as future work.

4.5.1.5 Voting cards

As voters are required to digitally sign vote transactions, they would be required to present the voting machine with the data necessary to validate themselves and sign the vote transaction. Voters could either store their validated public/private key pair on an electronic card or use a QR code. No personally identifiable information should be stored on the card. The key pair should be associated with an address on the blockchain owned by the voter and must be authorised by the voting authority. The card should be connected to the voting machine and used to generate a valid digital signature on the vote transaction. The private key on this card should be encrypted so as to prevent anyone other than the respective voter from being able to use it.

The exact implementation details and methods required to secure and issue the card are left as future work, but it should be noted that this would have to be conducted in a manner that ensures that the card cannot be linked to the identity of any voter.

4.5.2 Mid-election

The processes to be followed during the election are described in this subsection.

4.5.2.1 Voter authentication

Once a voter has arrived at a voting station, the electoral staff would verify the identity of the voter by using traditional identification means such as an identification document and/or biometric authentication and check that the voter is on the voting roll. While one could argue that this is not required and a voter could just present their voting card (which could be used to prove the voter has been verified if the public key has been verified), it is a necessary step to ensure voters are voting in their jurisdictions and to better control the election process.

4.5.2.2 Casting a vote

To cast a vote, a voter would use one of the voting machines described previously in Section 4.5.1.4. The voter would select their candidate of choice, with the voting machine then generating a transaction to this candidate. The voter would insert their voting card into the machine (or if QR codes are used, scan their QR code) and use a password to decrypt the private key so that the transaction may be signed. The machine would verify that the public key is authorised to vote and sign the vote using the decrypted private key.

This vote transaction should not be submitted immediately, but should instead be held until after the election has closed. This is necessary, even though the mapping between a candidate's identity and their respective blockchain address is currently not known to the public, as it prevents one from being able to track the distribution of votes during the election.

Further to this, if a more advanced blockchain such as Ethereum is used, smart contracts could also be created to allow for more complex election structures; voters could submit their voting preference in the event of runoff elections [37] or submit a ranking of candidates. For example, in the case of a runoff election, the voter could submit a smart contract specifying how to vote in the second round of the election and under what conditions to do so. In the case of ranked or preferential voting[38], either smart contracts could be used to encode the voter's preferred candidate order, or the transaction value could be used to rank candidates – the rank of candidates would correspond to the ranking of the transactions' values to the candidates.

4.5.3 Post-election

Once the election has concluded, a number of steps need to be followed to securely and accurately compute the election tally and open the system for public audit.

4.5.3.1 Releasing vote transactions

After the close of voting, it is safe for the voting machines to release their held votes to the blockchain for mining. These vote transactions should include a suitable transaction fee to incentivise miners to quickly include the transactions in blocks to be mined and added to the blockchain.

4.5.3.2 Freezing the blockchain

Once a sufficient time has passed to virtually guarantee that all vote transactions will have been added to the blockchain by miners, with a suitable number of blocks added on top to enforce immutability of the blocks containing the votes, the election tally can be computed. At this time, the blockchain will be 'frozen' and that particular instance of the publicly accepted main chain will be used as the final blockchain for tallying – any changes made after this point will not be considered. This is necessary to ensure that once the candidates' blockchain addresses have been released, voters are not able to submit votes that would be included in the tally.

4.5.3.3 Tallying the votes

In tallying the votes, one needs to ensure that each voter has cast only one vote. One of the core questions asked in Section 1.6 is how one could prevent a plutocratic system from developing – we need to guarantee that each authorised voter will only have one vote (or the maximum permitted number of votes per voter) included in the tally. While it would be easy to restrict voters to only casting one vote at an election centre using the provided electronic voting machine, it is difficult to prevent voters from voting remotely,

[37] Runoff voting is a voting method whereby a single candidate is elected through multiple rounds of voting. A voter casts a vote for their chosen candidate, and if no voter receives the required number of votes, then the candidates having less than a certain proportion of votes are eliminated and another round of voting occurs.

[38] Preferential voting describes a voting method in which voters rank outcomes in a hierarchy on an ordinal scale; it expresses more information than simple plurality voting methods.

hence the need to keep the association between a candidate's identity and blockchain addresses secret during the election.

If voters have somehow managed to submit multiple votes, only the most recent vote on the frozen blockchain will be accepted. The possibility of voters learning the identity of the candidate to which an address is assigned prior to the close of the election (allowing them to vote remotely and hence submit multiple votes) is discussed in Section 4.6.4.

The final tally of votes will be counted as the number of vote transactions to each candidate from unique and authorised voters. In the case where multiple votes have occurred, only the most recent vote will be counted. Where more complex voting methods are used, the tallying system is adjusted accordingly.

4.5.3.4 Releasing information to the public

Once the blockchain has been frozen, it is safe to make additional information such as the final election tally and the link between a candidate's identity and blockchain address public. One of the major benefits of using the blockchain for voting is that it allows for an open and publicly auditable election process. Thus, the link between a candidate's identity and address should be released (with the commitment serving as proof that this link was defined prior to the start of voting), making it is possible for any user to inspect the votes to the candidates. The list of authorised public keys would also have to be released so that people wishing to audit the tally can verify which vote transactions are legitimate. There is no security concern to the voter with releasing this list, as this data is stored openly on the blockchain and does not allow for one to link a vote to a voter's identity.

4.5.3.5 Receiving election feedback

Given that voters know their own blockchain address that was used for voting, as well as the fact that anyone can inspect the blockchain, once the election has concluded it would be possible for voters to inspect their transactions and ensure that their vote has been cast as intended. While unlikely, a compromised system may trick a voter into signing an incorrect vote and hence incorrect votes would be cast. This threat is discussed further in Section 4.6.7. If multiple users report that their votes have been cast incorrectly this would raise concern and a further investigation would be warranted. However, if only a limited number of users report incorrectly captured votes, it is more difficult to warrant a thorough investigation into the election as such issues may well be due to user error or voters who are dissatisfied with the election results trying to have the election voided.

4.6 SYSTEM VULNERABILITIES AND WEAKNESSES

This section examines several potential vulnerabilities and weaknesses of this blockchain based voting system. It is important to be aware of any potential issues and how these might affect the use and implementation of this system in the real world. Furthermore, it will become evident that the main potential weaknesses of this system do not arise as a result of flaws in the protocol itself, but rather the reliance on a voting authority. Generally speaking, these vulnerabilities arise only if the initial assumption that the voting authority is impartial is incorrect or if the voting authority fails to correctly and securely implement and administer the system.

4.6.1 Blockchain forks and the double-spend attack

Section 3.4.1 discussed the threat of double-spend attacks on the blockchain, whereby an attacker who controls enough of the network mining power (or in the case of proof of stake blockchains, a large enough stake) can successfully force a fork in the blockchain. It is important to note that double-spend attacks can only be implemented on transactions for which the adversary can produce a valid signature. As such, the adversary is unable to alter other users' transactions/votes. An adversary also has no motivation to perform this attack; no goods are transferred to the voter upon submitting a vote transaction, and the final, longest proof of work blockchain is always chosen as the final chain once a vote has completed (see Section 4.5.3.2). Thus, a voter cannot alter someone else's vote, submit multiple votes, or gain an advantage by performing a double-spend attack.

While blockchain forks by dishonest miners are a legitimate threat, the system introduced in this paper attempts to overcome it by requiring enough blocks to be added onto the blockchain, prior to freezing it, to virtually guarantee that the vote transactions would have all been confirmed. Furthermore, the use of a public blockchain makes this attack infeasible due to the large stake or hashing power required to successfully force a fork.

4.6.2 Denial of service attacks

In the application of voting, such an attack would allow an adversary to filter out votes for a candidate that he does not approve of, while letting votes for a favoured candidate be included in the blockchain. Depending on the type of blockchain used (see Section 3.3.3) this attack is near impossible to perform. In the case of public blockchains, the attacker would have to control a considerable portion of the hashing power of the entire worldwide blockchain network.

Private blockchain networks could restrict mining to an authorised and trusted set of miners. However, assuming an adversary could circumvent this restriction, or become an approved miner, the total mining power of the private network would be considerably less than that of a public network. As a result, the miner would likely be able to perform this attack with considerably less hashing power than would be required on the public network. Thus, while this attack scenario would not be infeasible on a private network, it would be near impossible on a large-scale public network.

This system overcomes the threat of a denial of service attack through two means: using a public blockchain, and hiding the identities associated with candidates' addresses. The use of a public blockchain makes it very difficult for a malicious actor to prevent a transaction from being included in the blockchain. This is because an attacker would likely have insufficient mining power to ensure that blocks created by honest miners are excluded from the main blockchain. Secondly, attackers do not know which candidate is associated with which address. Thus, selectively blocking unfavourable vote transactions would be infeasible.

4.6.3 Sybil attacks

The Sybil attack (whereby an attacker attempts to have an honest node connect only to hostile nodes) should not have any serious consequences in the case of blockchain voting. While it would theoretically leave one open to double-spend attacks, as discussed in Section 4.6.2, this should not be an issue for blockchain voting. Furthermore, the Sybil attack is extremely difficult to implement on the blockchain network due to the requirement that a node always select the chain with the longest proof of work as the

main chain (see Section 3.2.4). Thus, if a client is able to connect to even one honest node, the Sybil attack would fail.

4.6.4 Linking candidates to their respective blockchain address

Since voters are normally unable to link a candidate's address to the identity of the respective candidate, they would not know which address belongs to their preferred candidate and hence would be unable to vote for their chosen candidate remotely.[39] If, however, the identities of candidates are linked to their respective addresses prior to the close of the election (for example, via a security breach of the voting machines or databases that store this link), it would theoretically be possible for voters to vote for their preferred candidate remotely (and thus possibly under duress) as there is no way of verifying where a vote transaction was generated. Furthermore, adversaries could try to selectively prevent vote transactions to certain candidates from being mined through a denial of service attack (discussed previously in Section 3.4.2) against specific transactions.

A known breach of the link between a candidate's identity and their associated blockchain address should therefore require the election be repeated, as it would not be possible to guarantee free and fair elections – voting under duress or through coercion becomes possible as a result of remote voting.

4.6.5 Voter authentication

One potential attack vector would be to compromise the voter register. This would be possible by having the voting authority incorrectly add entries to (or remove entries from) the voting register that is used to store authorised public keys. This should not prove problematic if the voter authentication step described in Section 4.5.2.1 is carried out correctly, as the voting authority would only allow registered voters to proceed to the voting terminal. However, this attack, coupled with a breach of the identities associated with candidates' blockchain addresses, would allow malicious users to vote remotely by using the keys that were fraudulently inserted into the list of authorised public keys. Thus, if a breach of these links between candidates and their assigned address is discovered, it may be necessary to re-do the election.

4.6.6 Coercion resistance

As voting occurs at authorised voting stations, voting under direct duress should not be possible unless a breach of the link between candidates and their blockchain addresses has occurred (as discussed in Section 4.6.4). However, the problem with this election system is that it is not receipt free and voters are theoretically able to reveal how they have voted to another party. This is achieved trivially by having the voter prove ownership of the private key used to sign their vote transaction. Given the open nature of this voting system, this problem is hard to circumvent and is left as future work.

4.6.7 Voting devices

It would theoretically be possible for an attacker to compromise the devices used for generating and submitting the vote transactions. The compromised system could then fool the user into thinking his vote is being captured correctly and thus the user could be tricked into signing a transaction to a candidate that is not their choice candidate. As the

[39] This is why, as discussed in Section 4.5.1.4, this link is stored securely on the voting machines and is not revealed to the voters until the close of voting.

blockchain is open, a user would be able to later inspect the blockchain to ensure that the vote was cast as intended. However, as discussed previously in Section 4.5.3.5, an incorrectly captured vote is not a trivial problem to solve.

4.7 A COMPARISON OF BLOCKCHAIN BASED VOTING SYSTEMS

In this section, a comparison between this paper's suggested approach to blockchain based voting and some prior approaches (introduced previously in Section 2.3) is provided. Table 1 summarises several implementations of blockchain based voting and compares them to the system suggested in this paper. The approach taken throughout this paper has been to analyse prior work and attempt to develop a system that improves upon these prior implementations. Thus, as each component of the system developed in this paper has been thoroughly researched and compared to alternatives, the suggested approach should prove to be the current best implementation.

Table 1 - A comparison between several blockchain based voting implementations.

		Follow My Vote	**TIVI**	**Lee et al.**	**The Open Vote Network**	**This paper**
	Year introduced	2012	2016	2016	2017	2017
	Implementation	In trial	In use	Academic	Academic (with trial)	Academic
Blockchain	**Framework**	BitShares	Unknown	Bitcoin	Ethereum	Ethereum
	Privacy	Public	Public	Public or private	Public	Public
	Function	Ballot box[i]	Bulletin board[ii]	Candidate wallets[iii]	Enforce protocol execution	Candidate wallets
Security	**Third party[iv]**	Yes	Yes	Yes	No	No
	Self-tallying	No	No	No	Yes	No[v]
	Anonymity[vi]	Anonymous public key assignment	Vote mixing[vii]	Third party	Vote mixing	Blind signatures or zero-knowledge proofs
	Remote voting[viii]	Yes	Yes	Possible	Possible	No

[i] Ballots are submitted via proprietary voting software and stored on the blockchain.
[ii] A digital ballot box is used to store votes, with a blockchain being used to timestamp and securely commit each event, thus providing an immutable audit trail.
[iii] Each candidate is assigned a blockchain address, with the number of votes for that candidate being defined as the number of transactions, originating from unique and authorised voters, to the candidate's blockchain wallet.
[iv] Assuming a voting authority is always required, are any other third parties required for the electoral process?
[v] However, the tally can be publicly verified.
[vi] While all implementations aim to guarantee anonymity, the methods employed differ.
[vii] Votes are cryptographically mixed to destroy any correlation with the order in which they were cast and to maintain voter anonymity.
[viii] Whether or not the system is designed to allow remote electronic voting.

5 CONCLUSIONS AND FUTURE WORK

5.1 RECAP OF METHODOLOGY AND APPROACH

The aim of this paper was to determine the viability of utilising the blockchain in electronic voting, as well as to provide a secure and efficient implementation of such a system. This paper first discussed the rather complex and critical requirements of electronic voting and why, despite considerable research in this field, it has proven to be extremely difficult to implement a secure and viable electronic voting system. The reader was presented with an in-depth analysis of blockchain technology and several open questions were introduced. These questions formed a key aspect of the development of the electronic voting system introduced in this paper and ultimately governed many of the design decisions during the development of the voting protocol and system.

5.2 SUMMARY OF FINDINGS

A key question this paper looked to answer was whether or not blockchain based voting could guarantee the same level of security and anonymity as physical voting. It was shown that sufficient security and anonymity are ultimately very difficult to implement in an electronic voting system. The suggested blockchain based voting approach does satisfy many of the security requirements discussed in Section 2.1. Anonymity is guaranteed through the use of either blind digital signatures or zero-knowledge proofs, at the expense of adding additional complexity to the system. However, while the system appears to be theoretically secure, there is no way of guaranteeing that the system will be implemented correctly by the voting authority, nor that it is free from malicious users who could compromise the implementation of the system. Thus, while this system offers strong guarantees of security (in particular, authentication, integrity and anonymity), one cannot guarantee that the system, when implemented, is completely secure.

The second question this paper asked was how one could prevent multiple votes from being cast by a single voter or individuals with power and influence from manipulating the vote or blockchain. A public blockchain was chosen to reduce the likelihood that a consensus based attack would succeed. Furthermore, the election process proposed in Section 4.5 guarantees that only authorised voters can cast a single ballot. The security concerns of remote voting were discussed in Section 4.3, and ultimately the decision to prevent this from being possible was made and achieved by hiding the link between a candidate's identity and his assigned blockchain address during the election process.

The third and final question asked at the beginning of this paper was how voters could be sure that the voting machines have not been compromised in any way. The use of a public blockchain attempted to provide some form of guarantee to the voter that their vote has been cast and counted as intended, by allowing voters to inspect the blockchain once the election has closed. A concern with the developed system is that while voters would not knowingly sign an incorrect vote transaction, a compromised voting machine could theoretically trick a user into signing an incorrect vote. Unfortunately, the risk of compromised voting machines will always be present when electronic (or even mechanical) devices are used to capture or tally votes.

Ultimately, the initial objectives of this paper that were introduced in Section 1.2 were satisfied: Prior work and literature surrounding electronic voting, along with the security requirements of such schemes, was effectively and succinctly summarised in Section 2, and an effective and novel approach to electronic voting was introduced in Section 4. In so doing, this paper satisfied its key aim of thoroughly investigating the feasibility of blockchain based electronic voting. While blockchain based voting is theoretically feasible, the inability to offer a full guarantee of system security would imply that such a system should be used on a subjective basis. Unfortunately, the developed system is unable to provide the same level of security guarantees provided by traditional physical voting. However, such a system could serve as an effective springboard for further research into blockchain based voting, which can potentially be expanded upon and improved through future work.

5.3 Does electronic voting have a future?

Ultimately, the research conducted in this paper points to a much larger and critical problem with electronic voting in general: even if a protocol is theoretically secure, there is no guarantee or way to effectively prove that the system used for voting is in fact implementing the protocol correctly and has not been compromised. As such, while electronic voting offers numerous advantages over its physical counterpart, a flaw in such a system may make it just as easy for an attacker to compromise a single vote as it is for him to compromise a considerable number of votes. This is perhaps the primary reason for the continued ubiquity of physical voting systems, especially when the outcome of the vote has considerable consequences. The most secure form of voting remains the tried and tested physical ballot approach.

5.4 Future work

From the work presented in this paper, it is clear that electronic voting offers several advantages over traditional ballot box voting mechanisms. However, it also brings with it additional implementation requirements and introduces numerous new attack vectors. Whether electronic voting will ever progress to a point where it will be widely accepted as the optimal means of voting, especially in critical national elections, remains to be seen. Despite these concerns, it is important to continually expand the research in this field and work on furthering electronic voting protocols. While blockchain voting may, at this early stage of research, not be a viable option for security critical elections, it certainly is an idea worth exploring further.

Throughout this paper, several sections were explicitly mentioned as areas for future work. These include:

- Adapting the implementation to handle remote electronic voting. This would require one to develop a means to prevent coercive voting or voting under duress, which will undoubtedly prove to be non-trivial.
- Methods and techniques to secure the blockchain voting machines need to be developed as, similarly to most electronic voting systems, it will likely be the main target of attack.

- The methods of issuing and securing the voting cards that hold the voter's public/private key pair needs to be expanded upon, as this was not discussed in this paper.
- The public nature of the blockchain allows a voter to confirm that their vote has been recorded successfully and provides E2E verifiability, but this also allows voters to reveal their vote to others; ideally this should not be possible.
- Given the abundance of different blockchain technologies, a more in-depth comparison between them would be ideal. This may in turn yield a blockchain technology that is superior to Ethereum for conducting blockchain based voting.
- Formal proofs of the mathematical and cryptographic correctness of the protocols presented in this paper could be provided.

While electronic voting, in general, still has serious security concerns associated with it, it is important that research in this field continues. By continually experimenting and expanding the latest cryptographic and technological research, it is hoped that an electronic voting system will one day emerge that offers complete[40] security guarantees.

[40] Hopefully 'complete' will not have any qualifiers or footnotes associated with it.

6 REFERENCES

[1] F. Lehoucq, “Electoral Fraud: Causes, Types, and Consequences,” *Annual Review of Political Science,* vol. 6, no. 1, pp. 233-256, 2003.

[2] E. Nakashima, “Russian hackers targeted Arizona election system,” The Washington Post, 29 August 2016. [Online]. Available: https://www.washingtonpost.com/world/national-security/fbi-is-investigating-foreign-hacks-of-state-election-systems/2016/08/29/6e758ff4-6e00-11e6-8365-b19e428a975e_story.html?utm_term=.bf829516c13a. [Accessed 8 August 2017].

[3] M. Zapotosky and K. Demirjian, “Homeland Security official: Russian government actors tried to hack election systems in 21 states,” The Washington Post, 21 June 2017. [Online]. Available: https://www.washingtonpost.com/world/national-security/homeland-security-official-russian-government-actors-potentially-tried-to-hack-election-systems-in-21-states/2017/06/21/33bf31d4-5686-11e7-ba90-f5875b7d1876_story.html?utm_term=.64579eab3652. [Accessed 8 August 2017].

[4] Follow My Vote, “The online voting platform of the future - Follow My Vote,” 2017. [Online]. Available: https://followmyvote.com. [Accessed 19 March 2017].

[5] B. Schoenmakers, “A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting,” in *Advances in Cryptology (CRYPTO ’99)*, Santa Barbara, CA, USA, 1999.

[6] T. Okamoto, “Receipt-Free Electronic Voting Schemes for Large Scale Elections,” in *Proceedings of the 5th International Workshop on Security Protocols*, Paris, 1997.

[7] D. A. Gritzalis, Secure Electronic Voting, Springer US, 2003.

[8] A. L. Tsilidou and G. Foroglou, “Further Applications of the Blockchain,” in *12th Student Conference on Managerial Science and Technology*, Athens, 2015.

[9] P. McCorry, S. F. Shahandashti and F. Hao, “A Smart Contract for Boardroom Voting with Maximum Voter Privacy,” in *The 21st International Conference on Financial Cryptography and Data Security (FC'17)*, Malta, 2017.

[10] K. Lee, J. I. James, T. G. Ejeta and H. J. Kim, “Electronic Voting Service Using Block-Chain,” *The Journal of Digital Forensics, Security and Law,* vol. 11, no. 2, pp. 123-135, 2016.

[11] K. Hegadekatti, “Democracy 3.0: Voting Through the Blockchain,” 23 December 2016. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2889291. [Accessed 23 April 2017].

[12] United Nations General Assembly, “Universal Declaration of Human Rights,” Paris, 1948.

[13] P. G. Neumann, “Security Criteria for Electronic Voting,” in *16th National Computer Security Conference* , Baltimore, Maryland, 1993.

[14] R. G. Saltman, “Accuracy, Integrity, and Security in Computerized Vote-Tallying,” *Communications of the ACM,* vol. 31, no. 10, pp. 1184-1191, 1988.

[15] Internet Policy Institute, “Report on the National Workshop on Internet Voting,” in *Proceedings of the 2000 Annual National Conference on Digital Government Research*, CA, USA, 2000.

[16] A. D. Rubin, “Security Considerations for Remote Electronic Voting,” *Communications of the ACM,* vol. 45, no. 12, pp. 39-44, 2002.

[17] J. N. Wand, “The Butterfly Did It: The Aberrant Vote for Buchanan in Palm Beach County, Florida,” *American Political Science Review,* vol. 95, no. 4, pp. 793-810, 2001.

[18] W. R. Mebane, “The Wrong Man is President! Overvotes in the 2000 Presidential Election in Florida,” *Perspectives on Politics,* vol. 2, no. 3, pp. 525-535, 2004.

[19] R. Cramer, G. Rosario and B. Schoenmakers, “A Secure and Optimally Efficient Multi-authority Election Scheme,” *Transactions on Emerging Telecommunications Technologies,* vol. 8, no. 5, pp. 481-490, 1997.

[20] A. Juels, D. Catalano and M. Jakobsson, “Coercion-Resistant Electronic Elections,” in *Proceedings of the 2005 ACM workshop on Privacy in the electronic society (WPES '05)*, Alexandria, VA, USA, 2005.

[21] D. Jefferson, A. D. Rubin, B. Simons and D. Wagner, “Analyzing Internet Voting Security,” *Communications of the ACM,* vol. 47, no. 10, pp. 59-64, 2004.

[22] L. F. Cranor and R. K. Cytron, “Sensus: A Security-Conscious Electronic Polling System for the Intemet,” in *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Wailea, HI, USA, 1997.

[23] M. Clarkson, S. Chong and A. Myers, “Civitas: Toward a Secure Voting System,” in *IEEE Symposium on Security and Privacy (SP 2008)*, Oakland, CA, USA, 2008.

[24] D. DeSilver, “On Election Day, most voters use electronic or optical-scan ballots,” Pew Research Center, 8 November 2016. [Online]. Available: http://www.pewresearch.org/fact-tank/2016/11/08/on-election-day-most-voters-use-electronic-or-optical-scan-ballots/. [Accessed 18 June 2017].

[25] B. B. Bederson, B. Lee, R. M. Sherman, P. S. Herrnson and R. G. Niemi, “Electronic Voting System Usability Issues,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA, 2003.

[26] S. Kumar and E. Walia, “Analysis of Electronic Voting System in Various Countries,” *International Journal on Computer Science and Engineering,* vol. 3, no. 5, pp. 1825-1830, 2011.

[27] Ü. Madise and T. Martens, “E-voting in Estonia 2005. The first practice of country-wide binding Internet voting in the world,” in *2nd International Workshop on Electronic Voting*, Bregenz, Austria, 2006.

[28] M. R. Alvarez, T. E. Hall and A. H. Trechsel, “Internet Voting in Comparative Perspective: The Case of Estonia,” *PS: Political Science & Politics,* vol. 42, no. 3, pp. 497-505, 2009.

[29] T. Kalvet, “Management of Technology: The Case of e-Voting in Estonia,” in *Proceedings of the International Conference on Computer Technology and Development (ICCTD '09)*, Kota Kinabalu, Malaysia, 2009.

[30] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine and A. J. Halderman, “Security Analysis of the Estonian Internet Voting System,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*, Scottsdale, Arizona, USA, 2014.

[31] T. Kohno, A. Stubblefield, A. D. Rubin and D. S. Wallach, “Analysis of an Electronic Voting System,” in *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, California, 2004.

[32] J. Bannet, D. W. Price, A. Rudys, J. Singer and D. S. Wallach, “Hack-a-vote: Security issues with electronic voting systems,” *IEEE Security & Privacy,* vol. 2, no. 1, pp. 32-37, 2004.

[33] A. J. Feldman, A. J. Halderman and E. W. Felten, “Security Analysis of the Diebold AccuVote-TS Voting Machine,” in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology (EVT'07)*, Boston, MA, 2007.

[34] R. Gonggrijp and W.-J. Hengeveld, “Studying the Nedap/Groenendaal ES3B voting computer: a computer security perspective,” in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology (EVT'07)*, Boston, MA, 2007.

[35] S. Wolchok, E. Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati and R. Gonggrijp, “Security Analysis of India’s Electronic Voting Machines,” in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, Chicago, Illinois, 2010.

[36] D. Chaum, “Secret-ballot Receipts: True Voter-verifiable Elections,” *IEEE Security & Privacy,* vol. 2, no. 1, pp. 38-47, 2004.

[37] B. Adida, “Helios: Web-based Open-Audit Voting,” in *Proceedings of the 17th USENIX Security Symposium*, San Jose, CA, 2008.

[38] D. Chaum, R. T. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. A. Ryan, E. Shen, A. T. Sherman and P. L. Vora, “Scantegrity II: End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes,” *IEEE Transactions on Information Forensics and Security,* vol. 4, no. 4, pp. 611-627, 2009.

[39] C. Konstantinos and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access,* vol. 4, pp. 2292-2303, 2016.

[40] G. Zyskind, O. Nathan and A. Pentland, “Decentralizing Privacy: Using Blockchain to Protect Personal Data,” in *Proceedings of the 2015 IEEE Security and Privacy Workshops (SPW '15)*, San Jose, CA, 2015.

[41] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, “MedRec: Using Blockchain for Medical Data Access and Permission Management,” in *International Conference on Open and Big Data*, Vienna, Austria, 2016.

[42] M. Crosby, P. Pattanayak, S. Verma and V. Kalyanaraman, “Blockchain Technology: Beyond Bitcoin,” *Applied Innovation Review,* no. 2, pp. 6-10, 2016.

[43] M. Swan, Blockchain: Blueprint for a New Economy., Sebastopol, CA: O'Reilly Media, 2015.

[44] A. Hertig, “The First Bitcoin Voting Machine Is On Its Way,” VICE, 5 November 2015. [Online]. Available: https://motherboard.vice.com/en_us/article/the-first-bitcoin-voting-machine-is-on-its-way. [Accessed 20 June 2017].

[45] M. Rockwell, “BitCongress - Process For Blockchain Voting & Law,” BitCongress, [Online]. Available: http://www.bitcongress.org/. [Accessed 20 June 2017].

[46] TIVI, “TIVI powered by Smartmatic and Cybernetica,” 2017. [Online]. Available: https://tivi.io/. [Accessed 20 June 2017].

[47] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[48] A. Narayanan, J. Bonneau, E. Felten, A. Miller and S. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton: Princeton University Press, 2016.

[49] G. Volpicelli, “Beyond bitcoin. Your life is destined for the blockchain,” WIRED, 8 June 2016. [Online]. Available: http://www.wired.co.uk/article/future-of-the-blockchain. [Accessed 5 June 2017].

[50] V. Buterin, “On Public and Private Blockchains,” Ethereum Blog, 7 August 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/. [Accessed 26 June 2017].

[51] A. M. Antonopoulos, Mastering Bitcoin - Unlocking Digital Cryptocurrencies, O'Reilly Media, 2014.

[52] A. Back, “Hashcash - A Denial of Service Counter-Measure,” 2002.

[53] C. Dwork and M. Naor, “Pricing via Processing or Combatting Junk Mail,” in *12th Annual International Cryptology Conference on Advances in Cryptology - CRYPTO' 92*, Santa Barbara, CA, 1992.

[54] M. Jakobsson and A. Juels, “Proofs of Work and Bread Pudding Protocols,” *Communications and Multimedia Security,* pp. 258-272, 1999.

[55] M. Peck, ““Hard Fork” Coming to Restore Ethereum Funds to Investors of Hacked DAO,” IEEE Spectrum, 19 July 2016. [Online]. Available: http://spectrum.ieee.org/tech-talk/computing/networks/hacked-blockchain-fund-the-dao-chooses-a-hard-fork-to-redistribute-funds. [Accessed 19 August 2017].

[56] L. Graham, “Blockchain fork will create new digital currency called Bitcoin Cash,” CNBC, 31 July 2017. [Online]. Available: https://www.cnbc.com/2017/07/31/blockchain-fork-will-create-new-digital-crypto-currency-bitcoin-cash.html. [Accessed 19 August 2017].

[57] Ethereum Foundation, “Etherum Project,” 2017. [Online]. Available: https://www.ethereum.org/. [Accessed 2017 June 9].

[58] T. Gerring, “Cut and try: building a dream,” Ethereum Blog, 9 February 2016. [Online]. Available: https://blog.ethereum.org/2016/02/09/cut-and-try-building-a-dream/. [Accessed 6 July 2017].

[59] V. Buterin, “Ethereum: A next-generation smart contract and decentralized application platform,” [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper. [Accessed 9 June 2017].

[60] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” *Ethereum Project Yellow Paper,* 2014.

[61] “Proof of stake instead of proof of work,” Bitcoin Talk, 11 July 2011. [Online]. Available: https://bitcointalk.org/index.php?topic=27787.0. [Accessed 20 June 2017].

[62] S. King and S. Nadal, “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake,” *Independantly published,* 2012.

[63] V. Zamfir, “Introducing Casper "the Friendly Ghost",” Ethereum Blog, 1 August 2015. [Online]. Available: https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/. [Accessed 20 June 2017].

[64] A. P. Ozisik and B. N. Levine, “An Explanation of Nakamoto’s Analysis of Double-spend Attacks,” *arXiv,* vol. 1701.03977, 2017.

[65] J. R. Douceur, “The Sybil Attack,” in *First International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, Cambridge, MA, USA, 2002.

[66] O. Spycher, R. Koenig, R. Haenni and M. Schläpfer, “A New Approach Towards Coercion-Resistant Remote E-Voting in Linear Time,” in *International Conference on Financial Cryptography and Data Security (FC 2011)*, Gros Islet, St. Lucia, 2011.

[67] S. Wolchok, E. Wustrow, D. Isabel and A. J. Halderman, “Attacking the Washington, D.C. Internet Voting System,” in *International Conference on Financial Cryptography and Data Security (FC 2012)*, Bonaire, 2012.

[68] B. Van Acker, “Remote e-Voting and Coercion: a Risk-Assessment Model and Solutions,” in *Electronic Voting in Europe: Technology, Law, Politics and Society*, Gesellschaft für Informatik, 2004, pp. 53-62.

[69] F. Hao, P. Y. A. Ryan and Z. P, “Anonymous voting by two-round public discussion,” *IET Information Security,* vol. 4, no. 2, pp. 62-67, 2010.

[70] D. Chaum, “Blind Signatures for Untraceable Payments,” in *Advances in Cryptology: Proceedings of Crypto 82*, Boston, MA, Springer, 1982, pp. 199-203.

[71] S. Goldwasser, S. Micali and C. Rackoff, “Knowledge Complexity of Interactive Proof-Systems,” *SIAM Journal on computing,* vol. 18, no. 1, pp. 186-208, 1989.

[72] J. Camenisch and A. Lysyanskaya, “ Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials,” in *Advances in Cryptology — CRYPTO 2002*, Santa Barbara, CA, 2002.

[73] J. Camenisch, R. Chaabouni and A. Shelat, “Efficient Protocols for Set Membership and Range Proofs,” in *Advances in Cryptology - ASIACRYPT 2008*, Melbourne, 2008.

[74] V. Buterin, “On Settlement Finality,” Ethereum Blog, 9 May 2016. [Online]. Available: https://blog.ethereum.org/2016/05/09/on-settlement-finality/. [Accessed 27 June 2017].

[75] N. Bitansky, R. Canetti, A. Chiesa and E. Tromer, “From Extractable Collision Resistance to Succinct Non-Interactive Arguments of Knowledge, and Back Again,” in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS '12).* , Cambridge, MA, 2012.

[76] N. Bitansky, A. Chiesa, Y. Ishai, R. Ostrovsky and O. Paneth, “Succinct Non-Interactive Arguments via Linear Interactive Proofs,” in *The Tenth Theory of Cryptography Conference (TCC 2013)*, Tokyo, Japan, 2013.

[77] D. Boneh, “The Decision Diffie-Hellman Problem,” in *Proceedings of the 3rd Algorithmic Number Theory Symposium (ANTS-III)*, Portland, Oregon, USA, 1998.

[78] A. Kiayias and M. Yung, “Self-tallying Elections and Perfect Ballot Secrecy,” in *5th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC 2002)*, Paris, France, 2002.

[79] L. Margot-Duclot, “97: A secure e-government infrastructure using Ethereum and zero-knowledge proof,” 3 April 2017. [Online]. Available: https://blog.97.network/97-a-secure-e-government-infrastructure-using-ethereum-and-zero-knowledge-proof-7b87f3b7397e. [Accessed 20 April 2017].

[80] B. Blackwelder, K. Coleman, S. Colunga-Santoyo, J. S. Harrison and D. Wozniak, *The Volkswagen Scandal. Case Study,* University of Richmond: Robins School of Business, 2016.

[81] D. Florence Nightingale, Games, Gods, and Gambling: A History of Probability and Statistical Ideas, New York: Hafner Publishing Company, 1962.

[82] J. Benaloh and M. de Mare, "One-Way Accumulators: A Decentralized Alternative to Digital Signatures," in *Advances in Cryptology — EUROCRYPT '93*, Lofthus, Norway, 1993.