

1 Background

1.1 Evaluation of Previous Research

There has been some research conducted in this area already and several protocols for blockchain based voting have been proposed. *Pierre Noizat* proposes a system [3'noizat'2016] where each candidate provides a unique public key, KeyC, to each individual voter along with a singular bitcoin address, AddressC, which the final sum of the voting transactions will be sent to. Each voter is also assigned an individual public key, KeyA, by the election organizers and, either they generate a key pair themselves (this could be done by the voting software for better usability), KeyB, or be assigned the KeyB by the organization. From these three public keys, the voter can generate a 2-of-3 multi signature address which represents the vote of B in favour of C. This address is then funded with a bitcoin micropayment (around the price of a postage stamp), either by the voter or organisation, and this is the voters confirmation of their vote. After a few hours this ballot is securely logged on the blockchain and, as the multisignature address was funded, a voter can check that this address is represented in the blockchain and that their vote was registered. It should be noted that there is no way to guess neither the voter (B) nor the candidate (C) from a multisignature address without knowing all three public keys (KeyA, KeyB, and KeyC) and knowing to whom they belong. Once the election is concluded the organization is able to, via the 2-of-3 multi signature address, spend the coins the voter gave to the candidate to fund the address of the candidate, AddressC. This provides an unequivocal link between the vote and the candidate which can be seen and validated by anyone.

While the proposed method does provide a valuable alternative to current proprietary electronic voting systems and has the benefits of; protecting the secrecy of the ballots, allowing free, independent audits of the results and minimizing the trust level required from the organizers [3'noizat'2016]. It does come with several drawbacks; the independent validation of votes before the organization funds a candidate's public address, AddressC, can only be done by the voting individual on their ballot choice. The protocols dependence on the Bitcoin blockchain could pose problems with subsequent elections as there is no definitive boundary between one election and the next. The currency units, although in very small denominations, could be transferred out of the election to private addresses for an individual's gain (though even if 100% of the vote currency was lost, the cost of an election would likely be less than if done using current methods).

Another proposal is that of Universal Cast-as-Intended Verifiability [49'escala'guasch'herranz'morillo'2015] which allows any party (not only the voter) to publicly verify that an encrypted cast vote really matches the selection of a voter. Their proposal allows a voter who's eligible to vote to register with a registrar, who then generates a pair of public-secret values for each voting option in the election. These secret values

are sent to the voter, while the public ones are published, linked to the voting options they are related to. During the voting phase, the voter provides her selected voting options and a subset of the secret values she received during registration to the voting device. The voting device then encrypts the voters selections and creates a non interactive zero-knowledge proof, which will be valid only in the case that the voting device encrypted what the voter selected. Due to the zero-knowledge property of the proofs, they can be publicly verified while maintaining the voters privacy.

While this may seem like a vastly superior proposal on the surface, the additional complexity of the underlying system, that is the inclusion of zero-knowledge proofs, should not be underestimated. Furthermore, zero-knowledge protocols, despite being proposed in the late 1970s, are still in their relative infancy when compared to Blockchain technology (e.g. zCash whitepaper [50'ben-sasson'chiesa'garman'green'miers'tromer'virza'2014], 2014) and therefore have not been through the same level of scrutiny nor do they have the same level of development or adoption. The protocol also requires the voter to supply a secret value for each of the voting options they did not choose which, may require considerable effort if the ballot is large enough, and is counter intuitive to what would usually be expected to cast a vote.

1.2 Ethereum

1.2.1 What is Ethereum

In summary, Ethereum is an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications. The block chain is a decentralized network of computers who, at the most basic level, all maintain a ledger in consensus with each other. One block is added at a time, each block contains a mathematical proof that verifies it's addition to the chain and the transactions within are protected by a strong cryptography.

Ethereum is poised to become the next greatest innovation based on block chain technology. So is Ethereum similar to Bitcoin? It does have similarities, but not really. Like Bitcoin, Ethereum is a distributed public blockchain network, however there are some significant technical differences between the two. The most important distinction to note is that Bitcoin and Ethereum differ substantially in purpose and capability. Bitcoin offers one particular application of blockchain technology, a peer to peer electronic cash system that enables online payments. While the bitcoin blockchain is used to track ownership of digital currency (bitcoins), the Ethereum blockchain focuses on running the programming code of decentralized application [55'what'is'ethereum'a'step-by-step'beginners'guide'2017].

Ethereum enables developers to build and deploy decentralized applications. A decentralized application or 'Dapp' serves some particular purpose to its users, for example Bitcoin, is a Dapp that provides its users with a peer to peer payment system. Because decentralized applications are made up of code that runs on a blockchain network, they are not controlled by any individual or central entity. Running these Dapps on a decentralized platform, the blockchain, means they benefit from all of its properties [54'ethereum'explained'2017]:

- Immutability, a third party cannot make any changes to data.
- Corruption and tamper proofing, as apps are based on a network formed around the principle of consensus, making censorship impossible.
- No central point of failure, as Dapps can be run on every node in the network.
- Secured using cryptography, applications are well protected against hacking attacks and fraudulent activities.
- Zero downtime, Dapps never go down and can never be switched off.

1.2.2 Ethereum Blockchain

The concept of the blockchain was originally outlined in a white paper [12'nakamoto'2008] authored under the pseudonym Satoshi Nakamoto in November of 2008 and was quickly followed by an open source release of the Bitcoin proof-of-concept source code in January 2009 [13'nakamoto'2009]. This is the distributed ledger which underpins the entirety of the Bitcoin and Ethereum systems. A distributed ledger is a consensus of replicated, shared, and synchronized digital data geographically spread across multiple sites, countries, and/or institutions [24'distributed'ledgers'and'blockchain'technology'2016]. This ledger is stored locally on every node in the network which is running the full version of the blockchain software [14'bitcoin'2009] and records every transaction sent and confirmed on the network (the current size of the Ethereum Blockchain is around 21GB, March 2017[25'blockchain'size'2016]). This complete history, coupled with the fact that it is an open network, means that anyone can see what is happening in the network, not just now but during all periods in the past. This is extremely powerful as it allows an individual to fully audit the entire contents of the Blockchain without relying on external parties. This process is, in fact, what happens when you first download the full version of many blockchain reliant software [20'developer'guide'bitcoin'2016].

While the Ethereum Blockchain is not the only most mature distributed ledger in existence, it does have several years of being a publicly proven method to achieve distributed consensus and does this via the 'proof-of-work mining' process [24'distributed'ledgers'and'blockchain'technology'2016]. This is how new information gets added to the blockchain, by nodes in the network running a special 'mining' variant of the Ethereum software which uses considerable computing resources to win the right to add another block to the Blockchain, which is accompanied by a reward for the winning user. The concept of 'proof-of-work' is a method of ensuring that the information being added to the Blockchain was difficult (in terms of cost and time) to be made, though is easy for others to validate that the requirements were met [26'blockchain'mining'-distributed'ledgers'and'blockchain'technology'2016]. This means that the expenditure of computing power serves to secure the integrity of the Blockchain, while the miners themselves verify through public-private key cryptography the validity of each transaction they include in a block.

Blocks are chained together making it impossible to modify transactions included in any one block without modifying all following blocks; as a result, the cost to modify a particular block increases with every new block added to the block chain, magnifying the effect of the proof of work [20'developer'guide'bitcoin'2016][38'proof of work'. This is why, although a transaction is deemed clear upon its inclusion in a block on the Blockchain, best practices dictate that a user considers a transaction confirmed after its inclusion in a block and the addition of five subsequent blocks to the Blockchain [27'confirmation'-bitcoin'wiki'2016].

The difficulty of the proof-of-work mining needs to be controlled, so that an

average mining time of around 12 seconds per block is maintained. This time is somewhat arbitrary but is an attempt to find a balance between accepting transactions quickly and minimizing instability and waste in the network, as, while a new block is being distributed other miners will be working on an obsolete problem. As more miners join the network the block creation rate will increase due to the greater collective computational power. Therefore, every 2,016 blocks the difficulty of the mathematical challenge is recalculated so that the average mining time returns to normal [20'developer'guide'bitcoin'2016][26'blockchain'mining'-distributed'ledger'2016].

Despite the media often suggesting that bitcoin (and the Blockchain technology behind it) is an anonymous payment system, the Blockchain is in fact a transparent record of all user transactions on the network. Blockchain transactions are in fact pseudonymous, and your transactions in the network are like writing under a pseudonym. If an author's identity is ever linked to their pseudonym then everything written under that pseudonym will be revealed [28'anonymity'2016]. This is particularly poignant when considering the Blockchain as every transaction is stored forever, therefore a compromised address could lead to all transactions being linked to a person. There are however ways to reduce the amount of statistical analysis which can be done on transactions that a person is a part of which help to achieve reasonable anonymity [76'chan'liu'xeu'2013].

1.2.3 Mining and Ether

Ether is the fuel of the Ethereum system. It is the currency of the Ethereum network with which the payment of computation is achieved. Ethereum, like all blockchain technologies, uses an incentive-driven model of security where transaction consensus is based on a “proof-of-work” criterion of a given difficulty.

The block chain on which the Ethereum executes certain environment is known as the Ethereum Virtual Machine (EVM) [54'ethereum'explained'2017]. Each participating node within the network runs the EVM and performs the proof of work algorithm called Ethash which involves finding a nonce input to the algorithm so that the result is below a certain threshold (depending on the difficulty) [57'introduction'ethereum'frontier'guide'2017]. There is no better strategy to find such a nonce than enumerating the possibilities while verification of a solution is trivial and cheap. If outputs have a uniform distribution, then we can guarantee that on average the time needed to find a nonce depends on the difficulty threshold, making it possible to control the time of finding a new block just by manipulating difficulty [57'introduction'ethereum'frontier'guide'2017].

This is the process of how transactions are validated, new transactions are forwarded around the network and placed into a pool of unconfirmed transactions. These are not considered ‘accepted’ yet but are available for all to see almost instantaneously. Miners draw from this pool to create a candidate next set

of transactions to be officially accepted which will form the next block. The full text of all of these candidate transactions, along with the hash of the previous block and a nonce, are input into the the hash function (Ethereum) and miners will try different values for the nonce until the resulting hash is below a certain value. Because it's a cryptographic hash, there's no way to find a nonce that satisfies the output hashes condition other than attempting to guess [20'developer'guide'bitcoin'2016]. At this point, all of the miners are in a competition to find the hash first, each with a potentially different set of transactions to confirm. Once a miner succeeds they announce their solution to the rest of the network, their block becomes the next block in the Blockchain, and the transactions therein become confirmed. This strategy means that one miner will choose the next set of confirmed transactions, but the hash function effectively makes the miner a random one. All other mines then validate this new block, and the transactions held within, and can choose to accept it and start work on the next block. As the new block contains the hash of the previous block, this forms a chain of confirmed blocks securing the order of the transactions held within.

Occasionally, two miners may find a solution to the problem at the same time creating two potential next blocks in the chain. When miners produce simultaneous blocks at the end of the block chain, each node individually chooses which block to accept, this is usually the first block they see. Eventually another miner finds the solution to another block which attaches to only one of the competing blocks. This makes that side of the fork stronger and, as the general consensus is to use the strongest chain, other nodes will switch to this longer Blockchain [20'developer'guide'bitcoin'2016]. While this is statistically unlikely to happen, it is even more unlikely for the subsequent blocks to be solved at the same time, meaning that one fork will grow quicker than the other and the fork will resolve itself quickly. Transactions that were in the fork that wasn't chosen are not lost and are placed back into the unconfirmed transactions pool [4'driscoll'2016]. The fact that the end of the chain can be forked and rearranged means you shouldn't trust transactions at the end of the chain as much as ones further back. In Ethereum, a transaction is not considered confirmed until it is part of a block in the longest fork, and at least five blocks follow it. In this case we say that the transaction has "5 confirmations". This gives the network time to come to an agreed-upon the ordering of the blocks [35'nielsen'2013].

The successful miner of a block receives a reward for the 'winning' block, consisting of exactly 5.0 Ether along with all of the gas expended within the block, that is, all the gas consumed by the execution of all the transactions in the block. Over time, it's expected the gas reward will dwarf the block reward and become the main incentive for miners to continue working [57'introduction'ethereum'frontier'guide'2017].

1.2.4 Transaction Costs and Gas

Ethereum does have a small transaction fee, just like Bitcoin, where users pay a relatively small amount to the executor of your transaction. The sender has to pay the fees at each and every step of the activated program, this includes the memory, storage and computation [54'ethereum'explained'2017]. The size of the fee paid is equivalent to the complexity of the transaction, i.e. the more complex the commands you wish to execute, the more gas (and Ether) you have to pay. For example if "Alice" wants to send "Bob" 1 Ether unit, there would be a total cost of 1.00001 Ether to be paid by Alice. However if A wanted to deploy a contract or run a contract function, there would be more lines of code executable, therefore more energy consumption placed on the distributed Ether network and she would have to pay more than the 1 Gas done in the first transaction [56'ethereum'2017]. Some computational steps cost more than others, either because they are more computationally expensive or because they increase the amount of data that has to be stored in the state.

Gas is the internal pricing for running a transaction or contract in Ethereum. The gas system is not very different from the use of kilowatts in measuring electricity except that the originator of the transaction sets the price of gas, to which the miner can or not accept [56'ethereum'2017]. Ether and Gas are inversely related say for instance if the Ether price increases, than Gas price should decrease to maintain the concept of real cost [54'ethereum'explained'2017]. There is also a blocksize limit, so the more space your transaction takes up the more you have to pay to get it validated. With Bitcoin, miners prioritise transaction with the highest mining fees. The same is true of Ethereum where miners are free to ignore transactions whose gas price limit is too low.

The reason for the inclusion of a gas price per transaction or contract is to deal with the Turing Complete nature of Ethereum and its EVM essentially to guarantee that code running in the network will terminate. So for example, 0.00001 Ether or 1 Gas can execute a line of code or some command. If there is not enough Ether in the account to perform the transaction or message then it is considered invalid. This aims to stop denial of service attacks from infinite loops, encourage efficiency in the code and make any potential attacker pay for the resources they use (whether that be bandwidth, CPU cycles or storage) [56'ethereum'2017].

1.2.5 Smart Contracts

Smart contracts are the key element of Ethereum. In them, any algorithm can be encoded, they can carry arbitrary state and can perform any arbitrary computations even being able to call other smart contracts. This gives the scripting capabilities of Ethereum tremendous flexibility [59'peyrott'senanayaka'2017]. When run a smart contract becomes like a self-operating computer program that automatically executes when specific conditions are met and because they

run on the blockchain, they run exactly as programmed without any possibility of censorship, downtime, fraud or third party interference. While all blockchains have the ability to process code, most are severely limited. Ethereum is different in this respect as rather than giving a set of limited operations, Ethereum allows developers to create whatever operations they want allowing developers to build thousands of different applications that go far beyond anything seen previously [55'what'is'ethereum'a'step-by-step'beginners'guide'2017].

The Ethereum Virtual Machine is where smart contracts are run. It provides a more expressive and complete language than bitcoin for scripting and is Turing Complete. A good metaphor is that the EVM is a distributed global computer where all smart contracts are executed [58'the'hitchhikers'guide'to'smart'contracts'in'ethereum'2017]. There are several higher level languages used to program smart contracts, but Solidity is the most mature and widely adopted. Its syntax is similar to that of JavaScript, its statically typed, supports inheritance, libraries and complex user-defined types among other features.

Smart contracts are run by each node as part of the block creation process and, just like in Bitcoin, this is the moment where transactions actually take place. An important part of how smart contracts work in Ethereum is that they have their own unique address in the blockchain. In other words, contract code is not carried inside each transaction that makes use of it. Instead contracts are “deployed” to the blockchain in a special transaction that assigns an address to a contract. This transaction can also run code at the moment of creation. After this initial transaction, the contract becomes forever a part of the blockchain and its address never changes. Whenever a node wants to call any of the methods defined by the contract, it can send a message to the address of the contract, specifying data as input and the method that must be called. The contract will then run as part of the creation of newer blocks up (subject to the gas limit or completion) and can return a value or store data [59'peyrott'senanayaka'2017].