

Mneme: A Mobile Distributed Ledger

Dimitris Chatzopoulos*, Sujit Gujar[&], Boi Faltings[§], and Pan Hui*[#]

*HKUST, [&]IIT-Hyderabad, [§]EPFL, [#]University of Helsinki

dcab@cse.ust.hk, sujit.gujar@iiit.ac.in, boi.faltings@epfl.ch, panhui@cse.ust.hk

Abstract—Advances in mobile computing have paved the way for new types of distributed applications that can be executed solely by mobile devices on device-to-device (D2D) ecosystems (e.g., crowdsensing). More sophisticated applications, like cryptocurrencies, need distributed ledgers to function. Distributed ledgers, such as blockchains and directed acyclic graphs (DAGs), employ consensus protocols to add data in the form of blocks. However such protocols are designed for resourceful devices that are interconnected via the Internet. Moreover, existing distributed ledgers are not deployable to D2D ecosystems since their storage needs are continuously increasing. In this work, we introduce Mneme, a DAG-based distributed ledger that can be maintained solely by mobile devices and operates via two consensus protocols: Proof-of-Context (PoC) and Proof-of-Equivalence (PoE). PoC employs users’ context to add data on Mneme. PoE is executed periodically to summarize data and produce equivalent blocks that require less storage. We analyze the security of Mneme and justify the ability of PoC and PoE to guarantee the characteristics of distributed ledgers: persistence and liveness. Furthermore, we analyze potential attacks from malicious users and prove that the probability of a successful attack is inversely proportional to the square of the number of mobile users who maintain Mneme.

Index Terms—Distributed Ledgers, Consensus Protocols

I. INTRODUCTION

The popularity of permissionless distributed ledgers (DLs) increased by the development of Bitcoin in 2009 [1]. They store replicated data and are maintained by interconnected *consensus nodes* that exchange messages, can join and leave the system at any time and are self-interested or even malicious. The most popular data type is transactions between users of cryptocurrencies. Apart from transactions, ledgers have been used to store healthcare data, collected data from IoT devices, votes, ownership titles and others. The two most popular types of DLs are the blockchain and the directed acyclic graph (DAG). DLs need to have two properties to be functional: *persistence* and *liveness* [2]. Persistence measures how common is the view of the DL among the consensus nodes while liveness is associated the ability of the consensus nodes to add new data on the DL. Consensus nodes need to reach an agreement to add data in a DL. Depending on the design of a DL, they are rewarded for dedicating resources for its function. Bitcoin, Ethereum, and most cryptocurrencies, are using *Proof-of-Work* (PoW) consensus protocol. Consensus nodes in PoW protocols (*miners*) are rewarded proportionally to the processing power they contribute. Another well-known protocol is *Proof-of-Stake* (PoS), where consensus nodes (*validators*) are rewarded based on the stake they own. Miners and validators store locally the DLs whose storage needs are increasing since data can only be appended.

Mobile devices can not operate as consensus nodes in existing distributed ledgers due to (i) their poor processing capabilities compared to miners, (ii) their unstable connectivity, and (iii) their limited storage capacity. Motivated by the increase on the capabilities of mobile devices, the plethora of network interfaces [3]–[6], the advancements on energy efficient link layer protocols, neighbor discovery protocols and broadcasting on infrastructureless networks [7]–[9], we argue that a distributed ledger can be maintained solely by mobile devices if its function is guaranteed by protocols that are based on the characteristics of mobile devices.

We introduce Mneme, a DAG-based distributed ledger that is maintained solely by mobile devices, named *corroborators*, and is based on two consensus protocols that utilise the characteristics of mobile devices: *Proof-of-Context* (PoC) and *Proof-of-Equivalence* (PoE). PoC is using users’ context and reputation in its function to add blocks in Mneme. PoE detects blocks that can be deleted from Mneme and produces *regeneses blocks*. Both protocols are based on unforgeable proofs and terminate when a big fraction of the corroborators is informed about the addition of a block. Mathematical analysis using random geometric graphs and cryptographic protocols show that the properties of persistence and liveness are guaranteed by the design of Mneme.

The breakthrough of Bitcoin is the use of the blockchain and PoW as a solution to the Byzantine Generals problem [10], a classic problem in the distributed consensus literature. Formally, the problem of distributed consensus is: “Given a number of nodes and assuming that each node has an input value and a subset of the nodes may be malicious, a distributed consensus protocol has two properties: *termination* and *agreement*”. The first property implies that the protocol terminates with all the honest nodes agreeing on value and the second that the value the nodes have been agreed on is generated by an honest node. Satoshi Nakamoto, via PoW, violated the traditional assumptions upon which the impossibility results were built [11] by introducing incentives and randomization. The two most commonly employed protocols on permissionless distributed ledgers are PoW and PoS and are designed for nodes that can exchange messages via the Internet. In both of them, an election determines the next node to add a block in the blockchain. The nature of this election is different. In PoW, it is based on solving cryptographic inequalities [12], while in PoS it is based on a cryptographic protocol that creates randomness [2].

PoC and PoE are designed for ecosystems where mobile devices exchange messages opportunistically forming a par-

tially synchronous system. Their termination depends on the topology of the ecosystems and the introduced incentives. Their agreement is based on the created proofs. Consensus protocols designed for mobile ad-hoc networks cannot be utilised to maintain a distributed ledger because (i) they assume that every node is aware of the considered inputs and (ii) are based on message exchange [13], [14]. These make the performance of such protocols very poor since they require many messages to reach consensus.

Distributed Ledgers: The design of every distributed ledger has to guarantee persistence and liveness via two distributed consensus protocols. The first protocol adds data to the ledger and the second one verifies and allows their use. The blockchain is the most popular distributed ledger and is composed of blocks of data that are stored in sequence. A less popular type is directed acyclic graph (DAG), where the nodes of the graph can be either transactions [15]–[18] or blocks [19], [20]. DAGs are employed to increase the processing capacity of the ledger and are part of the so-called second-generation of distributed ledgers. Both ledgers require a genesis block, which is the first block in the ledger, as a point of reference for what is a correct ledger. Depending on the employed consensus protocol, the genesis block may contain additional information. Ouroboros PoS protocol, for example, considers a genesis block that contains the public-keys of the stakeholders, their respective stakes and auxiliary information that is used to seed a leader election process [2].

Blockchain: PoW and PoS require an honest majority to be functional. The property of persistence is guaranteed via the longest chain rule. In PoW it is probable for more than one miners to produce a block that extends the same block and create a fork. Forks are caused by the absence of coordination between the miners. Figure 1a depicts three blockchains with two forks. The property of liveness is guaranteed by the fact that malicious nodes can not prevent normal users from generating blocks. A block is verified if it is in a chain with at least six consecutive blocks. This rule prevents race and Finney attacks [21]. The security of a blockchain that uses PoW relies on honest nodes being sufficiently connected so that when one miner extends the chain with a new block, it propagates it in time to all honest nodes before the next one is created. To guarantee this property, the creation of blocks is regulated via the difficulty parameter in PoW. In Bitcoin, for example, a block is created every 10 minutes, on average. The low scalability of blockchain is the primary motivating factor for using DAG-based distributed ledgers.

Directed Acyclic Graphs: DAGs allow any generated block (or transaction) to be included in the graph, but they employ more sophisticated protocols of higher complexity to verify them [20]. Each node of the DAG is connected to more than two other nodes and each directed link from one node to another implies that the former verifies the later. Invalid blocks can be added on the DAG, but the other nodes will not verify them. Persistence and liveness are guaranteed by the voting between the DAG nodes that depends on how well each block is connected to the others. Figure 1b presents an example.

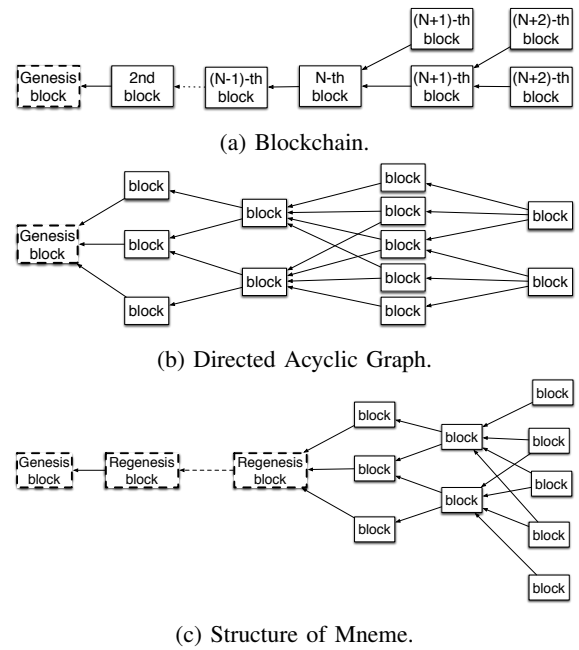


Fig. 1: Representation of a blockchain with two forks (Figure 1a), of a DAG (Figure 1b) and of Mneme (Figure 1c).

The main weakness of blockchains is the need for the corroborators to have the same view of the ledger. Given that we focus on an ecosystem where mobile device exchange messages opportunistically, this requirement slows-down the addition of data to the ledger. Mneme, as depicted in Figure 1c, has the form of a particular type of DAG that is composed of two types of blocks. The conventional blocks that are created via PoC and the blocks that are generated by a subset of users via PoE. We call them *regensis blocks*, and use them as synchronization points. Regenesis blocks also allow the deletion of conventional blocks to release storage space from the mobile devices. The blocks between regenesis blocks are deleted, and the included transactions are reorganized in new blocks hash pointers to the last regenesis block.

Applications of Mneme: The first application of Mneme is as a cryptocurrency in “underbanked” areas where people do not have financial footprint and Internet connectivity is limited. Advances on opportunistic computing highlight the need for automated payment schemes between mobile devices that assist each other in the execution of tasks and need to be compensated. The reduced connectivity on remote and rural areas has motivated the development of digital payment applications between mobile users (e.g., M-Pesa), but, to the best of our knowledge, none of the existing proposals focuses on the security of the ledger that stores the transactions. For example, the authors of [22] propose a blockchain-based payment scheme in a similar way to Mneme, but they assume intermittent connectivity to a bank. Their proposal is based on a system that is composed of smart contracts while their blockchain is not maintained by mobile users. The authors of [23] proposed a payment scheme that functions only with mobile devices, but they do not analyze the security proper-

ties of the underlying distributed ledger. Solutions similar to Mneme are already in use in mobile cloud computing architectures [24], [25]. In crowdsourcing, Mneme can be used to store the data generated by the mobile devices together with the additional information of the participants [26]. Furthermore, a popular technique to increase the anonymity of the users on cryptocurrencies is a laundry service [27].

II. OVERVIEW

We design Mneme to store blocks of transactions between mobile users and data produced by them without the assistance from remote servers. Mobile users, named *corroborators*, share their resources for the maintenance of Mneme to collect fees that are either generated by Mneme, via the PoE protocol, or are exchanged by mobile users who want to add data to the ledger. We introduce four types of fees that can be collected by the corroborators: *transaction fees*, *block creation fees*, *block deletion fees* and *block forwarding fees*.

Every distributed ledger that stores transactions of credit, need to guarantee that malicious users can not perform *double-spending attacks* while all the users can confirm the ownership of their credit. The defence against double-spending attacks is based on the consensus protocols that are responsible for the addition of new blocks in the ledger. The confirmation of credit ownership is demonstrated via hashpointers to verified blocks in the ledger. Mneme’s operation is based on PoC and PoE consensus protocols. PoC forces corroborators to produce proofs of their context before processing and sharing blocks with their neighbors. PoC terminates when a block has been accepted by enough mobile users that are scattered in the area of deployment. The idea behind the design of PoC is to not accept a block as verified before a big fraction of the corroborators is aware of its existence. PoC is designed in such a way to satisfy persistence and liveness. The parameters of PoC determine the rate with which blocks are added to Mneme and the difficulty a malicious user will face in adding a conflicting block. PoE is executed periodically and produces genesis blocks that can be used by users to synchronize with others and delete blocks from Mneme to released storage. PoE is based on *proofs of equivalence* that are composed of two sets of blocks whose transactions produce equivalent balances to the involved accounts. PoE runs periodically to produce the next genesis block. A subset of the corroborators is randomly selected, based on the fees they have collected, to execute the protocol. The design of Mneme is not based on any assumption regarding the capabilities of the devices. We only assume that mobile users are browsing within a predetermined geographic area and that there is an upper bound in the time needed for a message to be transferred between any two devices.

A. System Model

We consider a set of \mathcal{N} corroborators who are browsing in a geographical area \mathcal{A} . Each corroborator, $i \in \mathcal{N}$, has a public/private key pair (pk_i, sk_i) and is identified by her public key. Mobile devices can transact with each other to transfer credit and add data to Mneme. Each transaction contains fees

that will be allocated to the corroborators who assisted on the process. Non-financial applications, like crowdsourcing, can use Mneme to exchange data between mobile users [28]. In such scenarios, the collected credit by the corroborators should be exchangeable by the application.

The corroborators maintain a distributed ledger \mathcal{D} and, at time t , each one of them has her own view $\mathcal{D}_i(t)$. We expect corroborators to join and exit \mathcal{A} arbitrarily. A mobile user i is considered *active* and can participate in the maintenance of \mathcal{D} whenever she is in \mathcal{A} , otherwise the user is *inactive*. Active users at time t are denoted by $\mathcal{N}_a(t)$. Inactive users that become active can get synchronized by requesting the last genesis block and collecting accepted transactions that will be added to new blocks. For each user i , we denote her context by $c_i(t)$ and her reputation by $r_i(t)$, at time t . The context of a mobile user depends on her surroundings and can be measured via the sensors of her mobile device. The location of a mobile user i is part of her context and is denoted by $c_i^l(t)$. Similarly, her neighbors are denoted by $c_i^{\mathcal{N}}(t)$. Two users are neighbors if the distance between them is less than a communication threshold to allow a message exchange. The reputation of a user is calculated periodically and added to the genesis blocks. We measure users’ reputation based on the fees that they managed to collect since they are a clear measure of how much they have helped others. Trusted beacon devices and fingerprinting methods can also be employed to offer this functionality [29]. PoC and PoE are designed in such a way to tolerate a subset of the users to be *malicious* (also known as Byzantine users). We denote the malicious users by \mathcal{M} and their fraction by $f = |\mathcal{M}|/|\mathcal{N}|$.

B. Network Model

As an underlying network that allows corroborators to exchange messages, we consider a mobile ad hoc network [30], where mobile users are communicating opportunistically. We assume a partially synchronous network where at any time different users can have a different view of the ledger but with a common subset that includes the oldest blocks. Formally, there exist a timestamp $t_{sync} < t$ such that:

$$\mathcal{D}_i(t_{sync}) = \mathcal{D}_j(t_{sync}), \forall i, j \in \mathcal{N}, \{c_i^l(t), c_j^l(t)\} \in \mathcal{A}, i \neq j,$$

t_{sync} can be calculated via mobility prediction models [31]–[33]. It is worth mentioning that the usual assumption on network synchronicity is to assume that the network is synchronous (i.e., if an honest user broadcasts a message, then all honest validators receive the message within a known maximum delay) [1], [2], [34], [35]. We can not make this assumption in the examined setting because mobile devices, depending on the size of \mathcal{A} , become inactive more frequently than miners and validators in conventional Internet-connected networks that maintain distributed ledgers. Also, depending on the fraction of the malicious users and their actions a mobile user may fail to receive a transaction. We assume that every message is delivered to every active user in less than Δ . Any user who has not receive a message within Δ of its broadcast is considered inactive. We do not consider any fixed

infrastructure, but the proposed protocols are readily adaptable to areas where infrastructure is deployed.

C. Transactional Model

Mobile users can create transactions and broadcast them to their neighbors who will forward them until they reach the recipient. Every transaction, apart from the public keys of the sender and the receiver and the exchanged amount, has as input hash pointers to the blocks that the sender wants to use to justify the ownership of her funds. Honest users are motivated to forward transactions because they collect transaction fees. The broadcast medium allows honest mobile users to collect non-conflicting accepted transactions with the purpose to use them to create blocks. A corroborator can classify a transaction as accepted if she has collected an acknowledgment message from the recipient. The popularity of an accepted transaction that has not been added to a block depends on the transaction protocol. A recipient of a transaction is incentivized to broadcast the accepted transaction to speed up its addition to a block. Each mobile user stores accepted transactions locally in her pool of pending transactions $\mathcal{P}_i(t)$. Users can exchange pending transactions to participate in the block creation process. A transaction is verified and can be used as an input to another transaction if it is included in a block that is verified and added to Mneme.

Whenever Mneme is employed to non-financial applications, the exchanged transactions transfer data and fees. The fees have a central role since they are needed to motivate the corroborators to share their resources. In order to enable the merging of non-financial transactions, each transaction includes a field that indicates a function over the data and the number of the occurrences. PoE will be based on this field. For example, if one mobile user created three transactions to transfer a sensor reading to another mobile user and the function over the data is the average, PoE can produce a new transaction with the average data value and set the occurrences field to three. To demotivate malicious users to attack Mneme, we define a special type of transaction named *conditional self transaction*. These transactions are based on Proof-of-Burn policies [36]. Every user who participates either in PoC or PoE creates a conditional self-transaction with a number of credits that she will lose if she behaves maliciously.

Any user can propose a new block b by adding B transactions in it and triggering PoC. PoC is described in detail in Section III. PoE, on the other hand, is executed periodically every T time units (epoch), by a randomly selected set of users $\mathcal{K}_\tau \subset \mathcal{N}_a(\tau \cdot T)$. Where τ denotes the epoch index. The probability of a user to be selected in each epoch is proportional to her reputation. PoE terminates successfully if a minimum number of users, \mathcal{K}_τ^m reach an agreement.

D. Incentives

Incentives are required to balance the energy loss through the participation in the maintenance of the Mneme [37]. Users should be allowed to collaborate (e.g., like the miners of the Bitcoin mining pools), while their incentives are aligned with

the protocols [38]. Although the focus of this manuscript is the security guarantees of Mneme and not the engineering of the incentives required to guaranty Mneme's function, we use this paragraph to discuss the incentives needed without further analysing them. We list four types of incentives: transaction fees, block creation fees, block deletion fees and block forwarding fees. We introduce them motivated by the extensive literature on incentives in opportunistic networks and by their importance on the design of cryptocurrencies.

Transaction fees: Users who produce transactions add fees that will be collected by others. Similarly to conventional cryptocurrencies, the higher the fees, the faster a transaction is expected to be added to Mneme [39]. Every transaction is signed by users who forwarded it before its broadcast.

Block forwarding fees: Users who assist on the block creation or deletion processes share a fraction ϕ_c of the block creation fees and a fraction ϕ_d of the block deletion fees.

Block creation fees: To motivate users to store locally accepted transactions and add them to blocks, the user who produces it adds a block fee that will be collected by other users. A fraction $1 - \phi_c$ of block fees of the transactions that are included in a produced block are shared among the users who produced the block.

Block deletion fees: To motivate users to increase their reputation, every regensis block adds new credits to the system and assigns them to the users who run the PoE. Then these selected users produce a new block to share a fraction ϕ_d of their earnings with the users who forwarded the block deletion messages. The probability of a user to be selected to run PoE is proportional to the fees she has collected.

E. Proofs-of-Context

The functionality of Mneme depends on the users' context and primarily on their location that needs to be robustly estimated and verified by neighboring devices. To prove a user i that her location is as measured by her GPS or any localization method [40], [41], she uses the cryptographic protocol presented below:

Scanning: Scan for neighbours and produce a location message to interact with them: $m_i(t) = \{c_i^l(t), c_i^N(t)\}$

Tag production: Use $m_i(t)$ to produce a tag of fixed size, $tag_i = f(m_i(t))$, via a pseudo-random function [42] stored in the genesis block. A popular example is HMAC [43] that is used in SSL, SSH, etc. that produces tags of 256 bits.

Commitment: Use the secret key sk_i to produce a commitment for every neighbor $\Upsilon_i(t)$:

$$Comm(m_i(t), tag_i) \xrightarrow{sk_i} \Upsilon_i(t), \quad (1)$$

Validation: Every neighbour receives $\Upsilon_{ij}(t)$ and examines whether user i is at c_i^l at time t .

$$Ver(m_i(t), tag_i, \Upsilon_{ij}(t)) \xrightarrow{sk_j} \Omega_{ji}(t) \in \{\text{yes}, \text{no}\}. \quad (2)$$

$\Omega_{ji}(t)$ equals to "yes" if user j verifies that user i is her neighbour (i.e., their locations differ by less than a threshold) and "no" otherwise. Every user, after receiving $\Upsilon_i(t)$ can use the public key of i to extract the location user i claims to be

at time t and her neighbours together with tag_i . User i , by sending $\Upsilon_i(t)$ instead of $m_i(t)$ makes sure that her neighbors can only answer to her claim. Any malicious user is not able to change the location user i claims to be. The integrity of $\Upsilon_i(t)$ is guaranteed by the use of a pseudo-random function in the production of the the tag [44]. Practically, a malicious user can only produce a PoC for a location she is not currently in. By doing that, she will not be able to verify her fake location by normal users. Malicious users can still assist each other, and for that reason, we add a reputation weight to PoCs that depends on the reputation the neighbors.

Via this process user i can construct a *Proof-of-Context* that a set of her neighbors argues that are within a given distance threshold from her at time t .

$$\Pi_i^C \left(\Upsilon_i(t), \bigcup_{j \in \mathcal{C}_i^N(t)} \Omega_{ji}(t), \sum_{j=1}^{|\mathcal{C}_i^N(t)|} \frac{\mathbf{1}_{\{\Omega_{ji}(t) == \text{"yes"}\}} r_j(t)}{|\mathcal{C}_i^N(t)|} \right) \quad (3)$$

PoC is defined as the set of messages from the neighbouring devices of a user that the user is at a specific location. Each message is signed by the neighbouring users, and their worth is associated with their reputation.

F. Proofs-of-Equivalence

Every T time units PoE is triggered to create a genesis block and at the τ -th round users in \mathcal{K}_τ are randomly selected to produce the block. Each of the selected users i is responsible for collecting all the blocks that are added in the ledger during the τ -th round, \mathcal{D}_τ and create a smaller number of blocks with equivalent outcome \mathcal{L}_τ^i and produces a proof-of-equivalence:

$$\Pi_i^E(\mathcal{D}_\tau, \mathcal{L}_\tau^i) \quad (4)$$

Example: We consider two blocks with four transactions each between *Alice*, *Bob*, *Carol* and *David*. We assume that *Alice*, *Bob* and *David* are selected to summarise the two blocks and that they will share three credits as block fees. The eight transactions are: $tr(\text{Alice} \rightarrow \text{Bob}) = 5$, $tr(\text{Alice} \rightarrow \text{Carol}) = 2$, $tr(\text{Alice} \rightarrow \text{David}) = 2$, $tr(\text{Bob} \rightarrow \text{David}) = 1$, $tr(\text{David} \rightarrow \text{Carol}) = 2$, $tr(\text{Bob} \rightarrow \text{Alice}) = 1$, $tr(\text{Carol} \rightarrow \text{Alice}) = 1$ and $tr(\text{Carol} \rightarrow \text{David}) = 1$. The balance changes are $\text{Alice} \searrow 7$, $\text{Bob} \nearrow 3$, $\text{Carol} \nearrow 2$, $\text{David} \nearrow 2$, where the upright pointing arrow denotes an increase and the downright a decrease. We can now create one block and store four transactions where Alice will transfer to a *virtual user* $7 - 1$ coins and the virtual user will transfer $3 + 1$ coins to Bob, 2 coins to Carol and $2 + 1$ coins to David. Via this process we can delete the two blocks that contained the eight transactions and add one with four.

Alice, Bob, and David, will generate this block in the same way as a conventional blocks using PoC and add it to the genesis block. The only difference is that only the selected users are allowed to sign and verify it. Each of them is allowed to assign newly mined credits to herself. If the users do not have the same view, some of the blocks will not be created, and their creators will lose their credits. This motivates them to make sure that they are aware of every block needed before

the initiation of the process. The selected users should be more than the users needed to sign and verify one block.

III. PoC CONSENSUS PROTOCOL

Blocks are generated by mobile users and contain B transactions each. Every corroborator i can propose a new block b , at time t , by using B of the transactions she has witnessed and stored locally in $\mathcal{P}_i(t)$. We denote the transactions that are added to b by \mathbf{tr} , $|\mathbf{tr}| = B$ and the j -th transaction in \mathbf{tr} by \mathbf{tr}_j . The broadcast of one block requires $\mathcal{O}(\mathcal{N})$ messages if the topology is changing frequently [45]. Each block, after its creation, is connected to a number of existing blocks in Mneme via a set of hash pointers. These hash pointers are determined by the user i who initiated the creation of the block and are the hashes of the blocks, in i 's view, that does not have any incoming link in $\mathcal{D}_i(t)$. Each block after being verified by a set of corroborators, is added to Mneme. The users are sharing the same source of randomness that is determined in the genesis block and is updated on every regensis block [46], [2].

Honest users are expected to receive a new block within a given period, Δ . This means that the difference between the creation times of two contradictory blocks should be less or equal to a given period. If this difference is higher, there will be a path between the two blocks unless the second block was created by a malicious user who intentionally selects to not connect it to blocks that have a path to first one. Using a verification algorithm similar to the one proposed by [19] we can decide which of the two blocks to keep. The block creation process is composed of three phases and can be triggered by many mobile users at any time:

P_0^{PoC} **Block Creation and Broadcasting:** A user creates a block using B transactions and broadcasts it. Each block, apart from transactions and hashes to existing blocks, has a variable that stores the average distance between the users that will sign block b , \bar{l}_b and a set of user-context pairs:

$$\mathcal{D}_b = \{ \langle i, \Pi_i^C \rangle, \langle j, \Pi_j^C \rangle, \dots \}. \quad (5)$$

At this phase, \mathcal{D}_b contains only the public key of the user who created the block and her PoC, while \bar{l}_b is initialized to 0.

P_1^{PoC} **Block Signing and Forwarding:** Every corroborator that receives a block with a valid PoC, examines the included transactions and if she is familiar with every one of them she signs and adds her PoC to the set of user-context pairs and forwards the block.

P_2^{PoC} **Block addition to Mneme:** A block is considered verified if at least mRS users have signed it and the average distance between them is higher than mD . Whenever a user is the mRS -th user who receives a block, she examines whether the average distance between the users that have signed it is higher than mD . If this is true, she can use the locations of the users to construct a proof that verifies the block. If the users that have signed the block are not spread enough, the user rebroadcasts the block. This requirement ensures that each block is spread across the network. After the verification of the block, the users that have verified the block share a fraction of the block creation fees.

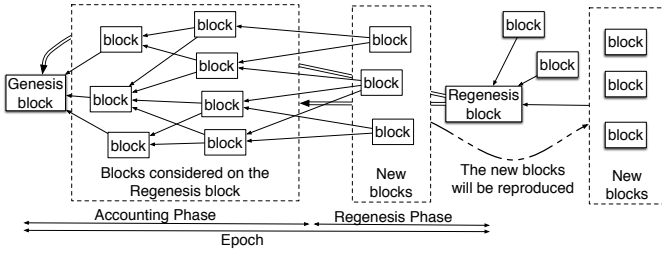


Fig. 2: Schematics of PoE consensus protocol.

Malicious users may try to sign a block using a fake PoC to speed up the block creation process or try to cooperate with each other to produce blocks that violate the proof-of-ownership rule. The required PoCs do not allow users to spoof their location while any block that violates the proof-of-ownership rule will be discarded and the malicious users who produced it will only waste their resources and credits from their conditional self-transactions.

Based on the analysis of SPECTRE [19] and by replacing PoW by PoS [2], persistence and liveness are still guaranteed. Note that, we also have extra security measures by incorporating Proof-of-Context, that is, every verifier of the block attaches its proof of location signed by her highly reputed neighbors. PoC scales with the number of the mobile users since the more the corroborators, the faster the messages are broadcasted, and the faster the blocks are created.

IV. POE CONSENSUS PROTOCOL

To reduce the storage overhead of Mneme we propose a solution to delete blocks and create *regenesis blocks* that summarise the transactions and the data of the deleted ones. Regenesis blocks are created in epochs by a randomly selected subset of users. The probability of one user to be selected is proportional to her reputation. That is, after every T , the protocol describe below is executed in order to produce a new regenesis block. In each round τ , the protocol selects \mathcal{K}_τ users out of \mathcal{N} users with high reputation to validate the regenesis block, and the regenesis block is accepted only if at least \mathcal{K}_τ^m of the \mathcal{K}_τ agree to it. These users will also share a fraction of the new coins Ξ_τ that will be produced via this process. In case of not reaching a consensus within the epoch, the regenesis block is not created and the selected users of the next epoch are responsible to produce a regenesis block that summarizes all the blocks of the last two epochs. The regenesis block creation process is triggered periodically and the users that will be responsible are selected at the beginning of each epoch. The source of shared randomness and the reputation of each user allows the participating users figure out if they are selected or not. Figure 2 depicts the whole process.

P_0^{PoE}] **Accounting Phase:** Every selected user $i \in \mathcal{K}_\tau$ at round τ actively monitors for blocks \mathcal{D}_τ that are added in Mneme, detect all the involved accounts, summarise their balance change and produces \mathcal{L}_τ^i . Via protocols like Spectre [19] each user is able to detect double spending attacks on the DAG. In parallel, the users calculate the fees each user collected in order to update the reputation scores.

| Symbol | Description |
|----------|---|
| B | Number of transactions per block |
| mRS | Minimum number of signatures that are required for a block to be added to the ledger. |
| mD | Minimum average distance between the corroborators that sign a block creation. |
| Δ | Maximum delay for the transfer of a block to all honest nodes. |
| T | Time between two regenesis blocks. |

TABLE I: Parameters of PoC and PoE.

P_1^{PoE}] **Regenesis Phase:** A regenesis block is created and is verified by the selected users. The reputation of each user is updated based on the fees that were allocated to her in the DAG. The regenesis block has a pointer to the previous regenesis block and to the header of every block in the DAG that was considered by the protocol. After its verification, the regenesis block is broadcasted and every user that receives it is able to delete the blocks that has stored locally and are included on the regenesis block. Depending on the number of the addresses involved on the examined DAG and the number of the transactions that can fit in a block, a number of blocks are created and are verified via PoC consensus protocol in the same way as being new blocks. In order to minimize the generated traffic, only a subset of the selected users are able to initiate this process and verify the new blocks.

V. ANALYSIS

To show that Mneme is functional, it suffices to show that it has two characteristic properties, which refer to the stored blocks: *liveness* and *persistence* [47]. These properties are guaranteed if the corroborators can add new blocks and if the produced blocks are broadcasted to every corroborator. Particularly, if PoC and PoE terminate and there exist an upper bound on the time a corroborator needs to get informed about a new event (i.e., if the network of corroborators is partially synchronous). In this section, we analyze the conditions under which PoC and PoE terminate, we introduced the considered threat model in the design of Mneme and we discuss Mneme's robustness against possible attacks.

Termination of PoC: Assuming that user u triggered PoC to produce block b at b_t , we prove that PoC terminates and b is included in \mathcal{D} . We define ρ_b^u as the probability of user $u \in \mathcal{N}_a(b_t)$ to be able to sign block b : $\rho_b^u = \prod_{j=1}^B Prob[\mathbf{tr}_j \in \mathcal{P}_u(t)]$, where $Prob[\mathbf{tr}_j \in \mathcal{P}_u(t)]$ is the probability of user u to have stored transaction \mathbf{tr}_j in her pool of pending transactions.

Under the assumption of a partially synchronous network, there exist a graph $\mathcal{G}(N, E)$ that shows how each user receives the block creation message. We assume that if one user received the same message by more than one users, she received it first by one who was able to sign it. This assumption is realistic if we force the users who are not able to sign a block creation message to back off for a few seconds before forwarding it. Each user is expected to have $\pi|\mathcal{N}_a(t)|\frac{R^2}{|\mathcal{A}|}$ neighbors [48], where R is the fraction between the coverage

area of the technology used from users to communicate over the deployment area (i.e., if the users communicate via WiFi-direct, that have a coverage radius of 50 meters, in an area 500 meters by 500 meters, $R = 0.1$). PoC terminates if the following problem has a feasible solution:

$$\begin{aligned} \min_{\mathbf{n}} \quad & \frac{1}{\binom{|\mathbf{n}|}{2}} \sum_{\substack{u \neq v, \\ u, v \in \mathbf{n}}} dist(u, v) \\ \text{subject to:} \quad & \frac{1}{\binom{|\mathbf{n}|}{2}} \sum_{\substack{u \neq v, \\ u, v \in \mathbf{n}}} dist(u, v) \geq mD \\ & \sum_{u=1}^{|\mathbf{n}|} \sum_{j=1}^B \mathbf{1}_{\text{tr}_j \in \mathcal{P}_u(t)} = |\mathbf{n}| \cdot B, |\mathbf{n}| \geq mRS \end{aligned} \quad (6)$$

where \mathbf{n} is any subset of $\mathcal{N}_a(t)$ that can sign the block and are spread enough. The existence of the feasible solution depends on $\{\rho_b^u\}_{u \in \mathcal{N}_a(t)}$, mRS and mD . ρ_b^u depends on the incentives provided to the mobile users. High values of mRS and mD will delay PoC and cause the generation of many forwarding messages but they will provide higher security guarantees.

Termination of PoE: Assuming that \mathcal{K}_τ corroborators are selected to produce the new genesis block. They need to collect all the created blocks of the epoch. We prove that PoE terminates and the new genesis block is produced. We define θ_τ^u as the probability of user $u \in \mathcal{K}_\tau$ to be able to produce the genesis block of the period τ : $\theta_\tau^u = \prod_{j=1}^D Prob[b \in \mathcal{D}_u(\tau)]$, where $Prob[b \in \mathcal{D}_u(\tau)]$ is the probability of user u to have stored block b in her local pool of Mneme. At least \mathcal{K}_τ^m out of \mathcal{K}_τ need to agree on the proposal of the genesis block and this will happen with probability:

$$P[PoE_termination] = \frac{\mathcal{K}_\tau}{\mathcal{K}_\tau^m} (\theta_\tau)^{\mathcal{K}_\tau} (1 - \theta_\tau)^{(\mathcal{K}_\tau - \mathcal{K}_\tau^m)} \quad (7)$$

assuming $\theta_\tau^u = \theta_\tau^v = \theta_\tau$, for any two corroborators u and v . In the case of $\theta_\tau^u \neq \theta_\tau^v$, the number of the corroborators who will sign the new genesis block can be approximated via a normal distribution with mean $\sum_{u \in \mathcal{K}_\tau} \theta_\tau^u$ and standard deviation $\sum_{u \in \mathcal{K}_\tau} \theta_\tau^u (1 - \theta_\tau^u)$. $P[PoE_termination]$ can be then calculated using the cumulative distribution function of the approximated distribution and \mathcal{K}_τ^m .

A. Threat Model

We denote the fraction of Byzantine corroborators by $f = |\mathcal{M}|/|\mathcal{N}|$, whose goal is to either steal credit or incapacitate Mneme. We expect them to act arbitrarily against PoC and PoE individually or in coordination. We assume that the resources of each of the Byzantine corroborators are equivalent to an honest corroborator. Practically, we expect Byzantine corroborators to: (i) not forward messages, (ii) generate fake transactions and blocks, (iii) forge PoCs and PoEs, and (iv) isolate honest corroborators. The most common attack is the double spending attack where malicious may assist each other to create two valid blocks with the same input transaction. By managing that, they will have successfully spent the same coins twice. Considering that Mneme allows the substitution

of blocks with less blocks of an equivalent impact on the users' balances, Byzantine corroborators may try to claim ownership of coins that do not belong to them. Malicious users may collaborate to perform Eclipse attacks [49] in order to isolate honest corroborators and spoof them to accept already spent transactions. Another type of a potential attack is the so-called Hijacking attack [50], where a Byzantine corroborator creates fake messages to misinform honest corroborators a new block/transaction.

Attacking PoC: There is a possibility that the corroborators are disconnected in disjoint components that are large enough to produce valid blocks. In such cases, a malicious user may try to double spend her coins, once in the first component and then in the second component. Using random geometric graph (RGG) theory, we can argue that such probability is inversely proportional to the square of the number of the corroborators. In more detail, given a 2-dimensional RGC that is composed of \mathcal{N}_a uniformly distributed users that can exchange a message if the distance between them is at most R , the following holds [51], [52]: For $\mathcal{N}_a R \geq 2 \log \mathcal{N}_a$, there is a path between any two users with probability $1 - \frac{1}{\mathcal{N}_a^2}$. Thus:

$$P[Double_spending] \leq \frac{1}{\mathcal{N}_a^2} \quad (8)$$

An active adversary may try to split a connected component by controlling mobile devices in a particular region such that messages are not transmitted from one component to another. For this, the width of such region under control should be at least $2R$, and again under uniform distribution model, the fraction of nodes that active adversary should control is non-practical.

Attacking PoE: A malicious user may try to form collusion of \mathcal{M} nodes to attack PoE. For a successful attack, at least $\lfloor \frac{\mathcal{K}_\tau}{2} + 1 \rfloor$ nodes should be selected from \mathcal{M} to participate in the process while \mathcal{K}_τ^m are needed to steal credit. If the selected malicious corroborators are more than $\frac{\mathcal{K}_\tau}{2}$ but less than \mathcal{K}_τ^m , they can prevent honest corroborators from building a new genesis block. There are $\binom{|\mathcal{M}|}{|\mathcal{K}_\tau|}$ ways of nodes getting selected to validate a genesis block. Out of which favorable to the adversary are: $\binom{\mathcal{M}}{\lfloor \frac{\mathcal{K}_\tau}{2} + 1 \rfloor} + \binom{\mathcal{M}}{\lfloor \frac{\mathcal{K}_\tau}{2} + 2 \rfloor} + \dots + \binom{\mathcal{M}}{\mathcal{K}_\tau^m}$ instances. Since the total number of instances is less than $2^{\mathcal{M}}$, the probability of a successful attack is:

$$P[Credit_stealing] \leq \frac{\binom{\mathcal{N}_a}{\mathcal{K}_\tau}}{2^{\mathcal{M}}} \approx \frac{\mathcal{K}_\tau!}{2^{\mathcal{M}} \mathcal{N}_a^{\mathcal{K}_\tau}} \quad (9)$$

This probability is high when \mathcal{K}_τ is small and \mathcal{M} is as large as possible. However, $P[Credit_stealing]$ is decreasing dramatically by the use of reputation scores in the selection of the \mathcal{K}_τ corroborators. **Example:** For $|\mathcal{N}_a| = 100$, $|\mathcal{K}_\tau| = 10$, and $|\mathcal{M}| = 10$, the probability of successfully attacking PoE is less than 2^{-28} . while for $|\mathcal{N}_a| = 1000$, and $|\mathcal{K}_\tau| = \mathcal{M} = 100$, it is less than 2^{-475} .

Impact of Mobility: The probability of a successful attack on PoC is decreasing dramatically by users' mobility since an active adversary will need to employ a higher fraction of

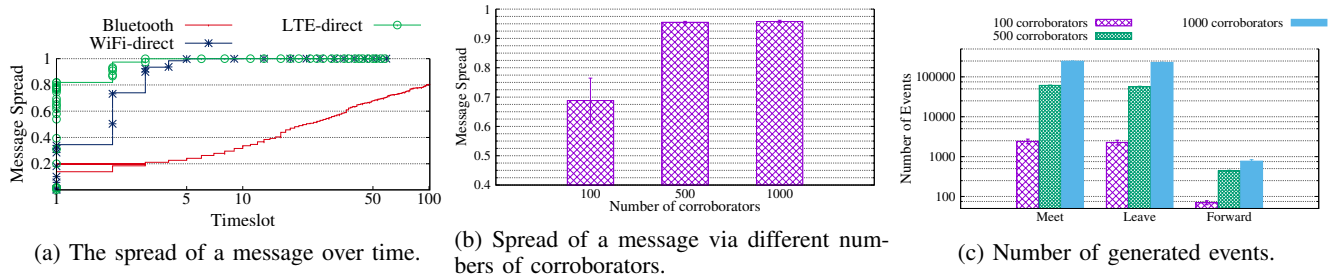


Fig. 3: Experimentation with the parameters of Mneme to characterise the interactions of the corroborators.

the users in order to stop message forwarding. Additionally, Eclipse attacks, designed for peer-to-peer networks, are not feasible in Mneme since corroborators, in difference to traditional consensus nodes, change their connected peers frequently over time. Similarly, Hijacking attacks are infeasible since the corroborators are communicating via broadcasting and not unicast protocols like the ones used in Internet-based ledgers. We do not consider specific mobility patterns for the movement of the users in order to robustly secure Mneme and guarantee persistence and liveness even in cases where corroborators are relatively static or move arbitrarily.

Parameters of Mneme: The operability and robustness of Mneme depends on the size deployment area ($|\mathcal{A}|$), the number of the active corroborators (N_a), the communication technology (e.g., Wifi-direct), the fraction of the malicious users (f), and four parameters (mRS , mD , \mathcal{K}_τ and T). Mneme has the properties of persistence and liveness if PoC and PoE terminate. PoC and PoE terminate if the network is partially synchronous (i.e., $\Delta < \infty$), which depends on the density of the corroborators in the deployment area and values of the four parameters. In more detail, the higher the density of the corroborators (i.e., a high coverage radius of the communication technology, high number of users, small deployment area), the lower the value of Δ , the faster PoC and PoE terminate. The higher the values of mRS and mD , the more time PoC needs to terminate and the more difficult it is for a malicious user to double-spend. The higher the value of \mathcal{K}_τ the more time it takes for PoE to terminate and the more difficult it is for a malicious user to claim coins he does not own. The longer the epoch T , the slower the storage needs of Mneme will be decreased, and PoE will produce less traffic.

VI. EVALUATION

In order to depict the conditions under which PoC and PoE terminate and show that Mneme has the properties of persistence and liveness, we implement an event-driven simulator in JAVA. The input to the simulator includes: the size the deployment area, the number of the mobile users, the employed communication technology, the duration of the simulation and the fraction of the malicious users. We consider an area of 500 meters by 500 meters where users are moving randomly towards any possible direction. We do not consider special mobility patterns or limitations on the users movement and we simulate mobile users who are moving with the same speed

(1 meter per slot). We use three communication technologies: Bluetooth, WiFi-direct and LTE-direct. We assume that two mobile users can exchange a message using Bluetooth if the distance between them is less than 20 meters, using WiFi-direct if it is less than 50 meters and using LTE-direct if it is less than 100 meters. If two users can exchange a message the simulator produces a “MEET” event and when they loose this ability it produces a “LEAVE” event.

Network Characterisation: Partially synchronous networks are characterised by a parameter Δ that represents an upper bound on the time needed for a message to be transferred to every active user¹. We analysed three scenarios with 1000 users in 100 slots and three communication technologies to measure Δ since it is directly related to the termination of PoC and PoE. Figures 3a, 3b and 3c show the produced results. Since LTE-direct has higher coverage radius, the messages are spread faster while in the case of Bluetooth time needed for a message to reach every corroborator is much higher. We also examine the impact of the number of users in the spread of a message in 100 slots and when the users communicate via WiFi-direct. Figure 3b shows how the message is spread in the cases of 100, 500 and 1000 users. 500 and 1000 users can guarantee the delivery of the message to every user while in the case of 100 users, 100 slots are not enough since less than 70% of the corroborators receive the message. This fact is also verifiable by the relatively high standard deviation of the fraction of the users who have received the message. Figure 3c depicts the number of the produced events (“MEET”, “LEAVE” and “FORWARD”) when a message is spread using WiFi-direct for different number of users. The more the users, the more events are produced. So, the higher the coverage radius of the communication technology, the faster PoC and PoE will terminate and the more the active corroborators the higher the probability for PoC and PoE to terminate within a given period.

Termination of PoC: We measure the average distance between the users that can verify a block for different values of ρ_b^u to examine whether PoC terminates. Figure 4a shows how the average distance between the users increases over time for three different values of ρ_b^u , 10%, 30% and 60%. Figure 4b depicts how the increase in ρ_b^u slows down the increase of the average distance between the corroborators. This is expected

¹In synchronous networks, all the users are active all the time while in asynchronous networks $\Delta = \infty$.

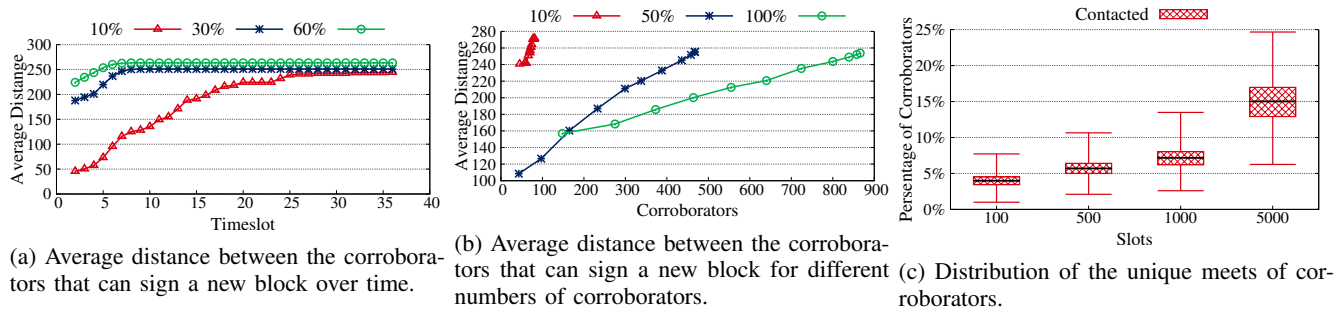


Fig. 4: Impact of the parameters of Mneme in the termination of PoC and PoE consensus protocols.

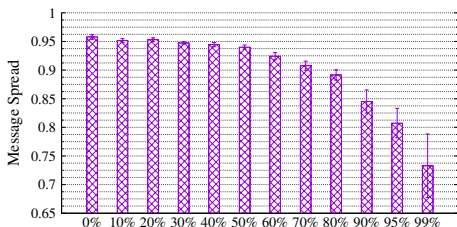


Fig. 5: Impact of non-assisting users to forwarding.

since the more probable it is for a corroborator to sign a block, the more corroborators will be found close to the one that initiated PoC and the more extra corroborators will be needed to reach an average distance between them, which is higher than mD . This result combined with the results of the previous paragraph shows that PoC terminates if mRS and mD are correctly selected (e.g., in the examined setting PoC will not terminate if mD is 300 meters or more).

Termination of PoE: We measure what fraction of the total users each user meets to examine whether PoE terminates. By measuring that quantity, we can estimate θ_r^u and determine the minimum number of corroborators needed to execute PoE. Figure 4c depicts four scenarios with 1000 users and four values of simulation duration. Each experiment is depicted via a candlestick where the black line shows the average, the lower line the minimum, the upper line the maximum and the boxed area contains the values between the 25th percentile and 75th percentile. Using Figure 4c we can argue that if \mathcal{K}_τ^m is less than 5% of the total population and the duration of τ is 500 slots, PoE terminates with probability 75% but if the slots are 5000, PoE terminates with certainty. Next, in Figure 5 we examine the impact of users who are not assisting in the delivery of messages. Via this plot, we can see that every message has more than 70% chances to be delivered even if 99% of the users are not helping with forwarding. Figures 4c and 5 show that PoE will terminate if the epoch duration and the number of the selected users are carefully selected.

Given Figures 3, 4 and 5, we can infer that PoC and PoE terminate when (i) there are enough users to maintain Mneme, (ii) mRS and mD are carefully selected to guarantee the termination of PoC and (iii) the minimum number of corroborators needed to execute PoE is high enough and the epoch duration of PoE is long enough to guarantee security.

VII. DISCUSSION

In this work, we sacrifice users' anonymity to keep reputation scores, and we do not discuss the energy needs of Mneme and the design of a fee scheme. The main privacy issue of public distributed ledgers is based on the fact that anyone can read its content of the blockchain, trace transactions and perform deanonymization attacks. Especially on the case we are focusing on, where corroborators are employing their context to verify blocks and maintain the ledger, their identity is more exposed. Unfortunately, most of the existing proposed solutions are based on zero-knowledge proofs that are computationally expensive [53]. The interconnectivity of the mobile users and their mobility patterns cannot be easily predicted by others, and this results in increases in traffic and energy consumption of the devices [54]. The performance of protocols that maintain a distributed ledger depends on energy-efficient neighbor discovery protocols that are complemented with bookkeeping functionalities that can be used on mobility prediction. Incentives are of high importance in the performance of Mneme. We introduced four types of fees to motivate corroborators to contribute resources by forwarding transactions and block messages, and storing transactions and blocks on their devices. However, a detailed analysis of the incentives is left for future work.

VIII. CONCLUSION

Distributed ledgers that are maintained solely by mobile devices can be employed by several mobile applications. The two most popular distributed ledgers are blockchain and directed acyclic graph. Both of them employ protocols that cannot function on mobile devices because they are designed for computationally powerful and well-connected nodes. In this work, we identified the need for a distributed ledger that can be maintained solely by mobile devices and designed Mneme to fill this need. Also we introduced two novel consensus protocols that are based on the users' context and serviceableness. The first one is responsible for data insertion in Mneme and the second one deletes blocks from Mneme to release storage resources. We prove that both protocols are secure against popular attacks and can guarantee the operability of Mneme. Experimental results provide a deeper understanding of the parameters of Mneme and the trade-off between performance and security.

REFERENCES

- [1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: <http://www.bitcoin.org/bitcoin.pdf>
- [2] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.
- [3] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.
- [4] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE wireless communications*, vol. 20, no. 3, pp. 96–104, 2013.
- [5] S. Mumtaz, K. M. S. Huq, and J. Rodriguez, "Direct mobile-to-mobile communication: Paradigm for 5g," *IEEE Wireless Communications*, vol. 21, no. 5, pp. 14–23, 2014.
- [6] D. Mark, J. Varma, J. LaMarche, A. Horovitz, and K. Kim, "Peer-to-peer using multipeer connectivity," in *More iPhone Development with Swift*. Springer, 2015, pp. 239–280.
- [7] A. Asadi and V. Mancuso, "Wifi direct and lte d2d in action," in *Wireless Days (WD), 2013 IFIP*. IEEE, 2013, pp. 1–8.
- [8] W. Shen, B. Yin, X. Cao, L. X. Cai, and Y. Cheng, "Secure device-to-device communications over wifi direct," *IEEE Network*, vol. 30, no. 5, pp. 4–9, 2016.
- [9] C. Funai, C. Tapparello, and W. Heinzelman, "Enabling multi-hop ad hoc networks through wifi direct multi-group networking," in *Computing, Networking and Communications (ICNC), 2017 International Conference on*. IEEE, 2017, pp. 491–497.
- [10] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.
- [11] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, 1985.
- [12] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Annual International Cryptology Conference*. Springer, 1992, pp. 139–147.
- [13] H. Moniz, N. F. Neves, and M. Correia, "Turquoise: Byzantine consensus in wireless ad hoc networks," in *IEEE/IFIP Dependable Systems and Networks*, 2010, pp. 537–546.
- [14] J. Brzezinski, M. Kalewski, and J. Kobusinski, "Providing uniform reliable broadcast delivery for mobile ad hoc networks with manet liveness property," in *IEEE SRDS*, 2012, pp. 237–242.
- [15] X. Boyen, C. Carr, and T. Haines, "Blockchain-free cryptocurrencies: A framework for truly decentralised fast transactions," *Cryptology ePrint Archive*, Report 2016/871, 2016.
- [16] A. Churymov, "Byteball: a decentralized system for transfer of value," 2016.
- [17] S. D. Lerner, "Dagcoin: a cryptocurrency without blocks," 2015.
- [18] S. Popov, "The tangle," *cit. on*, p. 131, 2016.
- [19] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol," *Cryptology ePrint Archive*, Report 2016/1159, 2016.
- [20] Y. Sompolinsky and A. Zohar, "Phantom: A scalable blockdag protocol," *Cryptology ePrint Archive*, Report 2018/104, 2018.
- [21] H. Finney, "The finney attack(the bitcoin talk forum)."
- [22] Y. Hu, A. Manzoor, P. Ekparinya, M. Liyanage, K. Thilakarathna, G. Jourjon, A. Seneviratne, and M. E. Yliantila, "A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain," *ArXiv e-prints*, Jan. 2018.
- [23] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "LocalCoin: An Ad-hoc Payment Scheme for Areas with High Connectivity," *ArXiv e-prints*, Aug. 2017.
- [24] J. Pan, J. Wang, A. Hester, I. AlQerm, Y. Liu, and Y. Zhao, "Edgechain: An edge-iot framework and prototype based on blockchain and smart contracts," *IEEE Internet of Things Journal*, 2018.
- [25] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "Flopcoin: A cryptocurrency for computation offloading," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1062–1075, May 2018.
- [26] M. H. Moti, D. Chatzopoulos, P. Hui, and S. Gujar, "Farm: Fair reward mechanism for information aggregation in spontaneous localized settings," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 7 2019*, pp. 506–512.
- [27] I. Miers, C. Garman, M. Green, and A. D. Rubin, "ZeroCoin: Anonymous distributed e-cash from bitcoin," in *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE, 2013, pp. 397–411.
- [28] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain," in *15th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2018, Chengdu, China, October 9-12, 2018*, 2018, pp. 442–450.
- [29] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94–104.
- [30] J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," 1999.
- [31] W.-J. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *IEEE INFOCOM 2007, 2007*, pp. 758–766.
- [32] J. Su, A. Chin, A. Popivanova, A. Goel, and E. De Lara, "User mobility for opportunistic ad-hoc networking," in *IEEE Workshop on Mobile Computing Systems and Applications*, 2004, pp. 41–50.
- [33] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International journal of network management*, vol. 11, no. 1, pp. 3–30, 2001.
- [34] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," *Cryptology ePrint Archive*, Report 2017/406, 2017.
- [35] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of ACM CCS*, 2016, pp. 17–30.
- [36] "Proof of burn," <https://bitcointalk.org/index.php?topic=1141676.0>.
- [37] N. Banerjee, M. D. Corner, and B. N. Levine, "An energy-efficient architecture for DTN throwboxes," in *IEEE INFOCOM, 2007*, pp. 776–784.
- [38] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proceedings of AAMAS*, 2015, pp. 919–927.
- [39] Earn.com, "Predicting bitcoin fees for transactions." [bitcoinfees.earn.com](https://www.earn.com), 2018.
- [40] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE personal communications*, vol. 7, no. 5, pp. 28–34, 2000.
- [41] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Cluster Computing*, pp. 157–167, Apr 2002.
- [42] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *ACM symposium on Theory of computing*, 1989, pp. 12–24.
- [43] H. Krawczyk, M. Bellare, and R. Canetti, "Hmac: Keyed-hashing for message authentication," *Tech. Rep.*, 1997.
- [44] S. G. Stubblebine and V. D. Gligor, "On message integrity in cryptographic protocols," in *Proceedings of IEEE Research in Security and Privacy*, 1992, pp. 85–104.
- [45] C. Newport, "Gossip in a smartphone peer-to-peer network," *arXiv preprint arXiv:1705.09609*, 2017.
- [46] I. Cascudo and B. David, "Scrape: Scalable randomness attested by public entities," *Cryptology ePrint Archive*, Report 2017/216, 2017.
- [47] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol with chains of variable difficulty," in *Annual International Cryptology Conference*. Springer, 2017, pp. 291–323.
- [48] M. R. Brust and S. Rothkugel, "Small worlds: Strong clustering in wireless networks," *CoRR*, vol. abs/0706.1063, 2007.
- [49] A. Singh *et al.*, "Eclipse attacks on overlay networks: Threats and defenses," in *IEEE INFOCOM*. Citeseer, 2006.
- [50] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *IEEE Security and Privacy*, 2017, pp. 375–392.
- [51] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [52] M. Penrose, *Random geometric graphs*. Oxford University Press Oxford, 2003, vol. 5.
- [53] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Practical decentralized anonymous e-cash from bitcoin," in *IEEE Symposium on Security and Privacy 2014*.
- [54] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.