Towards Mobile Distributed Ledgers

Dimitris Chatzopoulos, Anurag Jain, Sujit Gujar, Boi Faltings, and Pan Hui

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). 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 and analyse Mneme, a DAG-based distributed ledger that can be maintained solely by mobile devices. Mneme utilizes two novel 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 Mneme's security 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, D2D ecosystems

1 Introduction

The popularity of permissionless distributed ledgers (DLs) increased with the development of Bitcoin in 2009 [1]. They store replicated data and are maintained by interconnected *nodes* that exchange messages, which can join and leave the system at any time and are self-interested or even malicious. Although the most popular data type for DLs are transactions between users of cryptocurrencies, but they can also be used to store healthcare data [2], collected data from IoT devices [3], votes [4], 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* [5].

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. Persistence is essential to ensure that credits are final and that they happened at a particular "time" in the system's timeline (implicitly defined by the DL itself). The Liveness property ensures that the ledger makes progress by including new transactions. Consensus nodes eventually need to reach an agreement to ensure validity of 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. In *Proof-of-Stake* (PoS), another well-known protocol, 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.

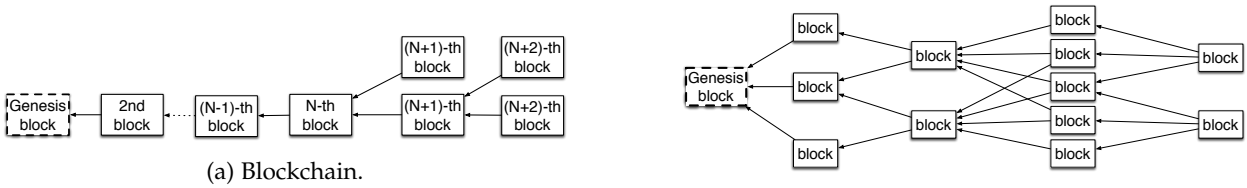
Motivation. Mobile devices can not operate as consensus nodes in existing distributed ledgers due to their *(i)* poor processing capabilities compared to miners, *(ii)* unstable connectivity, and *(iii)* limited storage capacity. Motivated by the advancement in mobile devices capabilities, the plethora of network interfaces [6]–[9], the energy efficient link layer protocols, neighbor discovery protocols, and broadcasting on infrastructure-less networks [10]–[12], 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 features of mobile devices.

Contributions. We introduce Mneme, a DAG-based DL that is maintained solely by mobile devices, called *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 *regenesis blocks*. Both protocols are based on unforgeable proofs and terminate when a big fraction of the corroborators is informed. Analysis using random geometric graphs and cryptographic protocols show that the persistence and liveness properties are guaranteed in Mneme.

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 in 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 [13], [14]. The reduced connectivity in remote and rural areas has motivated the development of digital payment applications between mobile users, but

- *Dimitris Chatzopoulos (dcab@cse.ust.hk) is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.*
- *Anurag Jain (anurag.jain@research.iiit.ac.in) and Sujit Gujar (sujit.gujar@iiit.ac.in) are with the International Institute of Information Technology, Hyderabad, India.*
- *Boi Faltings (boi.faltings@epfl.ch) is with the Ecole Polytechnique Federale de Lausanne, Switzerland.*
- *Pan Hui (panhui@cse.ust.hk) is with with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong and with the Department of Computer Science at the University of Helsinki, Finland.*

Manuscript received: date; revised: date



(a) Blockchain.

(b) Directed Acyclic Graph.

Fig. 1: Representation of a blockchain with two forks (left) and of a DAG (right).

none of the existing proposals focuses on the security of the ledger that stores the transactions. For example, the authors of [15] 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 [16] proposed a payment scheme that functions only with mobile devices, but they do not analyze the security properties of the underlying distributed ledger. Solutions similar to Mneme are already in use in mobile cloud computing architectures [13], [17]. In crowdsourcing, Mneme can be used to store the data generated by the mobile devices together with the additional information of the participants [18]. Furthermore, a popular technique to increase the anonymity of the users on cryptocurrencies is a laundry service [19]. Considering Internet of Things (IoT) applications, Mneme has extensive applicability. Representative examples are data marketplaces [20] and data truthfulness [21].

2 Background

The breakthrough of Bitcoin is the use of the blockchain and PoW as a solution to the Byzantine Generals problem [22], 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 the same 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 [23] by introducing incentives and randomization. The two most commonly employed protocols on permissionless DLs 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 propose a block in the blockchain. In PoW, the election is based on solving cryptographic inequalities [24], while in PoS it is based on a cryptographic protocol that creates randomness [25].

Consensus protocols designed for mobile ad-hoc networks (MANETs) 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 [26], [27]. These make the performance of such protocols very poor since they require many messages to reach consensus. The design of both PoC and PoE overcomes these constraints by interlinking their termination with the topology of the MANETs and introduced incentives. Their agreement is based on the created proofs.

Distributed Ledgers. The design of every distributed ledger has to guarantee persistence and liveness via two consensus protocols. The first protocol adds data to the ledger and the second verifies it and allows its use. The blockchain is the most popular distributed ledger and is composed of blocks of data stored in sequence. A less popular type is directed acyclic graph (DAG), where the nodes of the graph can be transactions [3], [28]–[30] or blocks [31], [32]. DAGs are employed to increase the processing capacity of the ledger and are part of the so-called second-generation of distributed ledgers [33], [34]. 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 protocols, 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 [25]. Algorand uses byzantine agreement protocol to achieve consensus among the users and mine a new block [35].

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 miner 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 [36]. 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 store all the generated blocks (or transactions) in the graph, but they employ more sophisticated protocols of higher complexity to verify them [32]. Each node of the DAG, as depicted in Figure 1b, is connected to at least 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

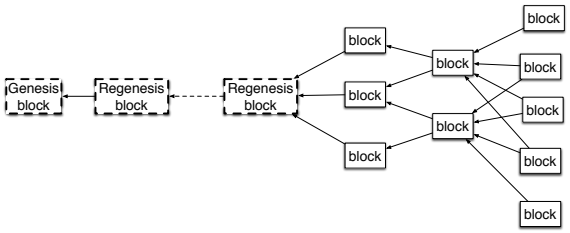


Fig. 2: Structure of Mneme.

are guaranteed by the voting between the DAG nodes that depends on how well each block is connected to the others. B. Cao et al. [37] provide an overview of challenges faced by DAG-based consensus in IoT.

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 2, 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 *regenesis blocks*, and use them as synchronization points. Regenesis blocks trigger the deletion of conventional blocks for releasing storage space. The blocks between regenesis blocks are deleted, and the included transactions are reorganized in new blocks hash pointers to the last regenesis block. Similar ideas to release storage space and reduce the time needed for synchronising new nodes have been incorporated by popular blockchains such as Ethereum [38].

3 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 needs 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 regenesis 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. This allows us to reduce the requirement of storing the entire blockchain by storing only the recent regenesis blocks and headers of previous blocks. PoE runs periodically to produce the next regenesis 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.

3.1 System Model

We consider a set of $\mathcal{N}$ corroborators who are browsing in an 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 [39]. 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 regenesis 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 regenesis 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 [40]. 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 $f = |\mathcal{M}|/|\mathcal{N}|$.

As an underlying network that allows corroborators to exchange messages, we consider a MANET [41], where mobile users are communicating opportunistically in a device-to-device manner [42]. Practically, mobile users broadcast

messages to their neighbors motivated by the implemented incentives and without following a specific routing protocol. We do not make any assumption regarding the wireless links between users since Mneme is agnostic to underlying D2D technologies. We assume a partially synchronous network where at any time users may have a different view of the ledger but with a common subset that includes the oldest blocks. I.e., there exist a $t_{sync} < t$ 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 [43]–[45]. 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], [25], [46], [47]. 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 with such infrastructure.

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. However, a transaction cannot be added to the blockchain unless it is acknowledged by the receiver. This induces a meaningful delay in the acceptance of the transaction that allows the entire network to receive the transaction and detect any possible double spends happening simultaneously.

The receiver of one transaction will accept the transaction if she receives the transaction signed by at least *a minimum number of trusted users* of her trusted network, denoted by mTr and atleast δ time has elapsed since she had first received the transaction. Where δ can be chosen by the user that is going to accept the transaction. Letting the users pick δ introduces flexibility since the user may decide to accept a transaction quickly if it is of relatively low value or perceived as less risky (coming from a trusted user). [1]

The authors of [16] show that such an acceptance rule is sufficient to prevent simultaneous double spending attacks as long as there are no disjoint components in the network. In our analysis, we show that the probability of such an event is negligible. Therefore, once the receiver acknowledges the transaction, he/she can be confident that no honest user has accepted a conflicting transaction.

Each user stores acknowledged transactions locally in her pool of pending transactions $\mathcal{P}_i(t)$. Any user can propose a new block b by adding B transactions in it and triggering PoC. PoC is described in detail in Section 4. 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, $\mathcal{K}_\tau^m$ reach an agreement. 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 [48]. 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.

1. $\delta = \Delta$ assures complete safety against a double spending attack.

3.2 Incentives

Incentives are required to balance the energy loss through the participation in the maintenance of the Mneme [49]. Users should be allowed to collaborate (e.g., like the miners of the Bitcoin mining pools), while their incentives are aligned with the protocols [50]. Although the focus of this manuscript is the security guarantees of Mneme and not the engineering of the incentives required to guarantee Mneme's function, we discuss the incentives needed without further analysing them. We list four types of incentives and 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 Mnene [51]. 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 regenesis 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.

3.3 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 [52], [53], she uses the cryptographic protocol presented below:

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

2) Tag production. Use $m_i(t)$ to produce a tag of fixed size, $tag_i = f\left(m_i(t)\right)$, via a pseudo-random function [54] stored

in the genesis block. A popular example is HMAC [55] that is used in SSL, SSH, etc. that produces tags of 256 bits.

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

$$\mathcal{C}omm\left(m_i(t), tag_i\right) \xrightarrow{sk_i} \Upsilon_i(t), \tag{1}$$

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

$$\mathcal{V}er\left(m_i(t), tag_i, \Upsilon_{ij}(t)\right) \xrightarrow{sk_j} \Omega_{ji}(t) \in \{\texttt{yes}, \texttt{no}\}. \tag{2}$$

$\Omega_{ji}(t)$ 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 [56]. 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.

$$\Pi_i^C\left(\Upsilon_i(t), \bigcup_{j \in c_i^{\mathcal{N}}(t)} \Omega_{ji}(t), \sum_{j=1}^{|c_i^{\mathcal{N}}(t)|} \frac{1_{\{\Omega_{ji}(t)==\text{“yes”}\}} r_j(t)}{|c_i^{\mathcal{N}}(t)|}\right) \tag{3}$$

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.

3.4 Proofs-of-Equivalence

Every T time units PoE is triggered to create a regenesis block and at the τ-th round $\mathcal{K}_\tau$ users 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, $\mathcal{D}_\tau$ and create a smaller number of blocks with equivalent outcome $\mathcal{L}_\tau^i$ and produces a PoE:

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

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: $tr(Alice \rightarrow Bob) = 5$, $tr(Alice \rightarrow Carol) = 2$, $tr(Alice \rightarrow David) = 2$, $tr(Bob \rightarrow David) = 1$, $tr(David \rightarrow Carol) = 2$, $tr(Bob \rightarrow Alice) = 1$, $tr(Carol \rightarrow Alice) = 1$ and$tr(Carol \rightarrow David) = 1$. The balance changes are $Alice \searrow 7$, $Bob \nearrow 3$, $Carol \nearrow 2$, $David \nearrow 2$, 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 regenesis block in the same way as a conventional blocks using PoC. 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.

4 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 [57]. 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 regenesis block [58], [25].

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 [31] 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:

$$\mathcal{D}_b = \{< i, \Pi_i^C >, < j, \Pi_j^C >, \ldots\}. \tag{5}$$

At this phase, $\mathcal{D}_b$ contains 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.

Any user that receives a verified block will accept the block if and only if none of the new transactions conflict with the set of transactions it had accepted earlier. 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 [31] and by replacing PoW by PoS [25], persistence and liveliness are still guaranteed. Note that, we also have extra security measures by incorporating PoC, that is, every verifier of the block attaches its proof of context 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.

5 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 contained in the blocks that have been produced since the last regenesis block. Hence, the participating mobile devices would be able to delete all blocks since the last regenesis block and replace them with a new regenesis block that contains all non-redundant data from the deleted blocks. 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 $\mathcal{K}_\tau$ users out of $\mathcal{N}$ users with high reputation to validate the regenesis block, and the regenesis block is accepted only if at least $\mathcal{K}_\tau^m$ of the $\mathcal{K}_\tau$ 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 3 depicts the whole process.

Accounting Phase $P_0^{PoE}]$ Every selected user $i \in \mathcal{K}_\tau$ at round τ actively monitors for blocks $\mathcal{D}_\tau$ that are added in Mneme, detect all the involved accounts, summarise their balance change, and produce $\mathcal{L}_\tau^i$. The summary contains the balance of each user along with their public key, hence, the pending transactions would not get invalidated since the signatures could still be verified against the same public keys. A different public key is treated as a separate user. Via protocols like Spectre [31] each user is able to detect double spending attacks on the DAG. In parallel, the users calculate the collected fees in order to update the reputation scores.

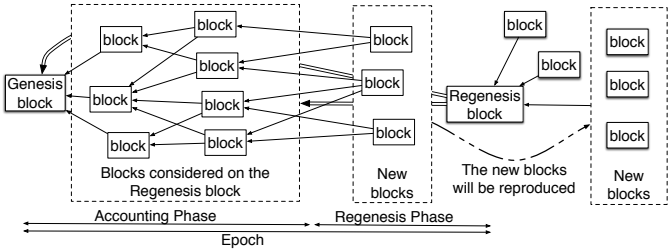


Fig. 3: Schematics of PoE consensus protocol.

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.
T	Time between two regenesis blocks.
δ	Time receiver waits before accepting a transaction.
Δ	Maximum delay for the transfer of a block to all honest nodes.

TABLE 1: Relevant Notation

Regenesis Phase $P_1^{PoE}]$ 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 order to minimize the generated traffic, only a subset of the selected users are able to initiate this process and verify the new blocks.

6 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* [59]. 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. 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.

PoC Termination. 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: $\rho_b^u = \Pi_{j=1}^{B} Prob\left[\mathbf{tr}_j \in \mathcal{P}_u(t)\right]$, where $Prob\left[\mathbf{tr}_j \in \mathcal{P}_u(t)\right]$ 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 users who are not able to sign a block creation message 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 [60], where R is the fraction between the coverage area of the used communication technology 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:

$$\min_{\mathbf{n}} \quad \frac{1}{\binom{|\mathbf{n}|}{2}} \sum_{\substack{u \neq v, \\ u,v \in \mathbf{n}}} dist(u,v) \tag{6}$$

$$\text{subject to:} \quad \frac{1}{\binom{|\mathbf{n}|}{2}} \sum_{\substack{u \neq v, \\ u,v \in \mathbf{n}}} dist(u,v) \geq mD$$

$$\sum_{u=1}^{|\mathbf{n}|} \sum_{j=1}^{B} 1_{\mathbf{tr}_j \in \mathcal{P}_u(t)} = |\mathbf{n}| \cdot B, |\mathbf{n}| \geq mRS$$

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 provided incentives. High values of mRS and mD will delay PoC and cause the generation of many forwarding messages but they will provide higher security guarantees.

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

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

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 coroborators who will sign the new regenesis 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 $\mathcal{K}_\tau^m$.

7 Threat Model

We denote the fraction of Byzantine corroborators by $f = |\mathcal{M}|/|\mathcal{N}|$ where $\mathcal{M} \geq mRS$ that may strategically deviate from the protocol. Their 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 [61] 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 [62], where a Byzantine corroborator creates fake messages to misinform honest corroborators a new block/transaction.

7.1 Characteristics of Adversaries

To justify the robustness of Mneme, we consider adversaries with capabilities that normal participants do not have. We consider adversaries that have the following advantages over honest corroborators by deviating from the protocol.

Wormhole Attack. In a *Wormhole Attack*, the attackers locate themselves strategically across the network and communicate across a secret channel. This might allow the adversary to transfer a message between geographically separated attackers without broadcasting the same to the network.

Unbounded Block Creation. Notice that since $\mathcal{M} \geq mRS$, the adversary can produce a valid PoC by signing a block with only its corroborators. The adversary could potentially distribute the corroborators geographically so that the average distance between them becomes greater than mD. The adversary could then collect both the private and public keys of the corroborators at a single location and produce the mRS signatures required, without any network delays. Although, such an attack is not possible on other distributed ledgers like Bitcoin or SPECTRE, we show that this does not lead to a loss of security in Section 7.3.

7.2 Analysis of Possible Attacks

Attacking PoC via the Wormhole Attack. 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 [63], [64]: 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:

$$P[Double\_spending] \leq \frac{1}{\mathcal{N}_a^2} \tag{8}$$

An active adversary may try to split a connected component by controlling 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. The probability would reduces even further in a setting with mobile users that are actively moving around, with the possibility that a user that was part of one component moves to another component. Thus, if we can achieve a certain level of security with n static users, we could achieve it too with $m(< n)$ mobile users.

7.3 Analysis of Double Spending Attacks

Producing Acknowledgement for Double Spends. We consider a double spending attack successful if an adversary manages to spend the same input in transactions to two honest nodes. In order to do this, the adversary would require mTr users from both the nodes trusted set to sign the transaction without any of the recipients of the transactions receiving the other transaction. In case of a double spending attempt, the adversary would need to propose her transaction to mTr users of another user before they receive the first transaction. If a trusted user detects the double spending attempt, she would alert the respective user who would then reject the transaction. We show that once a delay of δ has elapsed and mTr trusted users have signed and forwarded the transaction, this transaction would have been sufficiently propagated throughout the network (Figure 6a). Thus, either the trusted user will register the transaction or detect a double spend. Therefore, the double spending attempt would be foiled by social engineering.

As observed in Figure 6a, by setting $\delta = 5$ times the time taken to transfer the message between two users, the user can ensure that 95% of other honest users would receive the message. Hence, the probability of a double spending attack being successful would be negligible since it would not only require the other recipient to belong to the remaining 5% but also atleast mTr trusted user of that recipient to also belong to this set of users.

Faking an Acknowledgement. Let us consider if in the previous scenario, one of the transaction is addressed towards the recipient and another recipient is the adversary itself. In this case, he/she would generate an acknowledgement for the transaction forcefully. In this case, even if the adversary manages to produce a block using PoC that contains the transaction addressed to herself, the block would be rejected by rest of the honest users that contain the original transaction in their $\mathcal{P}_i(t)$ or they had already accepted a block containing the original transaction.

Attacking PoE via Collusion: 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 $\mathcal{K}_\tau^m$ are needed to steal credit. If the selected malicious corroborators are more than $\frac{\mathcal{K}_\tau}{2}$ but less than $\mathcal{K}_\tau^m$, they can prevent honest corroborators from building a new regenesis block. There are $\binom{|\mathcal{N}|}{|\mathcal{K}_\tau|}$ ways of nodes getting selected to validate a regenesis block. Out of which favorable to the adversary are: $\binom{\mathcal{M}}{\lfloor \frac{\mathcal{K}_\tau}{2}+1 \rfloor} + \binom{M}{\lfloor \frac{\mathcal{K}_\tau}{2}+2 \rfloor} + \ldots + \binom{\mathcal{M}}{\mathcal{M}}$ instances. Since the total number of instances is less than $2^{\mathcal{M}}$, the probability of a successful attack is:

$$P[Credit\_stealing] \leq \frac{\binom{\mathcal{N}_a}{\mathcal{K}_\tau}}{2^{\mathcal{M}}} \approx \frac{\mathcal{K}_\tau!}{2^M \mathcal{N}_a^{\mathcal{K}_\tau}} \tag{9}$$

This probability is high when $\mathcal{K}_\tau$ is small and M is large and it is decreasing dramatically by the use of reputation scores in the selection of the $\mathcal{K}_\tau$ corroborators.

Example. For $|\mathcal{N}_a| = 100, |\mathcal{K}_\tau| = 10$, and $|\mathcal{M}| = 10$, the probability of successfully attacking PoE is $< 2^{-28}$. while for $|\mathcal{N}_a| = 1000$, and $|\mathcal{K}_\tau| = \mathcal{M} = 100$, it is $< 2^{-475}$.

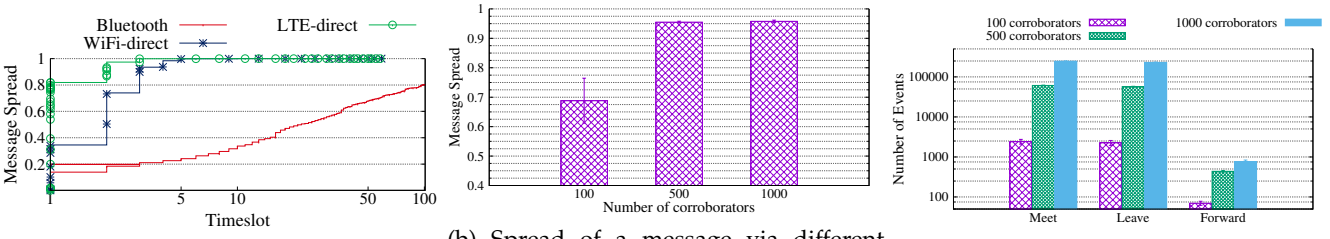
Impact of Mobility. The probability of a successful attack on PoC is decreasing dramatically by users' mobility since an active adversary will need to employ a higher fraction of the users in order to stop message forwarding. Additionally, Eclipse attacks, designed for peer-to-peer networks, are not 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 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 ($\mathcal{N}_a$), the communication technology (e.g., Wifi-direct), the fraction of the malicious users (f), and four parameters (mRS, mD, $\mathcal{K}_\tau$ 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 number of users, small deployment area) or the higher the coverage radius of the communication technology (e.g., LTE-direct instead of bluetooth), 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 $\mathcal{K}_\tau$ 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. In a deployment area where users can form disconnected components, mRS and mD need to be carefully selected in order to make sure that a block is validated by most of the components.

Example. For a coverage radius of 100 meters (which could be achieved by LTE-direct), note that having more than 500 users per sq km guarantees that any communication would reach all users in 10 minutes with a probability $> 99\%$. For comparison, the average population density of an urban settlement is typically more than 10,000 people per sq km.

8 Evaluation

In order to depict the conditions under which PoC and PoE terminate and show that Mneme has the properties

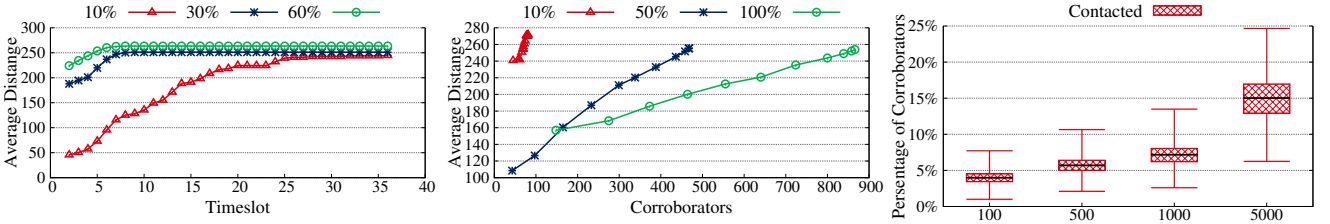


(a) The spread of a message over time.

(b) Spread of a message via different numbers of corroborators.

(c) Number of generated events.

Fig. 4: Experimentation with the parameters of Mneme to characterise the interactions of the corroborators.



(a) Average distance between the corroborators that can sign a new block over time.

(b) Average distance between the corroborators that can sign a new block for different numbers of corroborators.

(c) Distribution of the unique meets of corroborators.

Fig. 5: Impact of the parameters of Mneme in the termination of PoC and PoE consensus protocols.

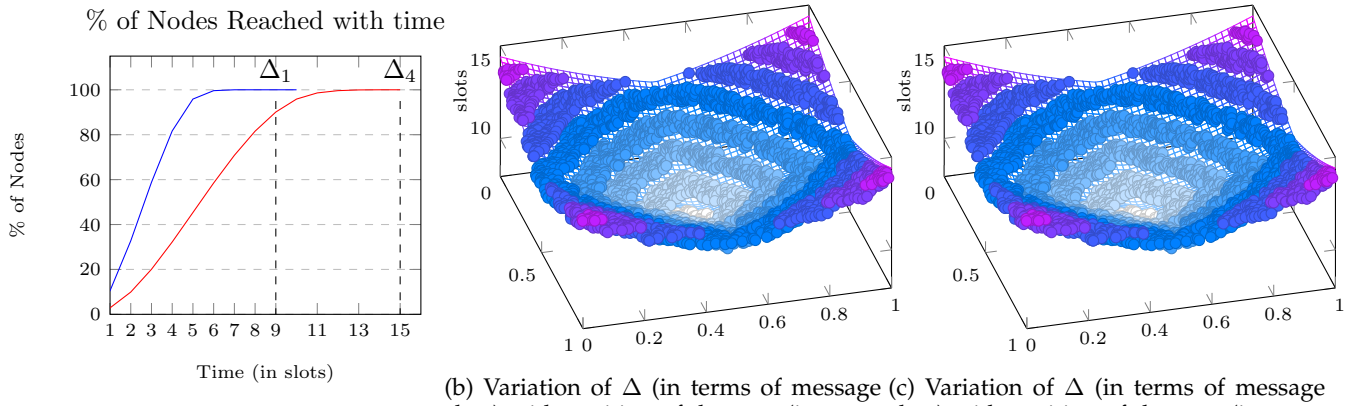
of persistence and liveness, we implement an event-driven simulator in JAVA. The input to the simulator includes: the size the deployment area, the number of the mobile users, the employed communication technology, the duration of the simulation and the fraction of the malicious users. We consider an 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 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 and using LTE-direct if it is less than 100. 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 user[2]. 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 4a, 4b and 4c show the 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 4b 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 4c 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.

Approximating Δ. Since Δ is not in common knowledge of every user, each user would need to calculate the approximate upper bound Δ by itself. The Δ depends upon two parameters: (a) The location of the user which he can determine using GPS (Figure 6b and 6c demonstrate the variation of Δ with the location of the user) (b) The network characteristics which may vary with time and cannot be determined deterministically. We describe a procedure to approximate Δ as follows:

1) The user sends a message containing a timestamp (a_i) to some of his trusted users.
2) The trusted users reply with the message along with the timestamp of when they received the message (b_i), timestamp of when they send their reply(c_i) and their location (x_i, y_i).
3) The user calculates the distance between the trusted user and himself as $d_i = \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2}$ and the time average time difference as $t_i = \frac{(b_i - a_i) + (d_i - c_i)}{2}$.
4) Let matrix $A = \begin{bmatrix} d_1 & 1 \\ d_2 & 1 \\ \dots & \dots \end{bmatrix}$, $x = (pq)$, $y = \begin{bmatrix} t_1 \\ t_2 \\ \dots \end{bmatrix}$
5) Let $x = (A^T A)^{-1} A^T y$ then we can set $\Delta = p \times (\max(x_o^2, (1-x_o)^2) + \max(y_o^2, (1-y_o^2)) + q$

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



(a) Spread of a message in a RGG for 1000 Users spread over 1 Sq. Km.

(b) Variation of Δ (in terms of message slots) with position of the user (in a setting with static users)

(c) Variation of Δ (in terms of message slots) with position of the user (in a setting with mobile users)

Fig. 6: Variation of Δ (in terms of message slots) along with parameters of Mneme and the position of a user

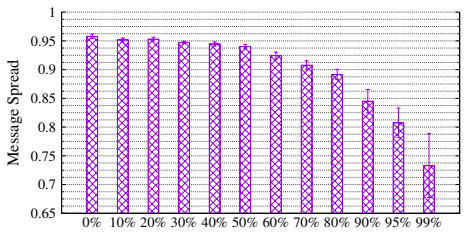


Fig. 7: Impact of non-assisting users to forwarding.

Friendly Relaxation. Since Mneme would be deployed in a local setting where all users are in vicinity of each other, it is likely that users know each other socially as well. In that case, if the recipient trusts the sender of the transaction that he/she will not double spend or the transaction is of nominal value, the recipient could skip or reduce δ for the particular transaction at his/her convenience to accommodate quicker commerce.

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 5a shows how the average distance between the users increases over time for $\rho_b^u = 10\%$, $\rho_b^u = 30\%$ and $\rho_b^u = 60\%$. Figure 5b depicts how the increase in ρ_b^u slows down the increase of the average distance between the corroborators. This is expected 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 $mD > 300$ meters).

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_τ^u and determine the minimum number of corroborators needed to execute PoE. Figure 5c 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 25^{th} percentile and 75^{th} percentile. Using Figure 5c we can argue that if $\mathcal{K}_\tau^m$ 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 7 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 5c and 7 show that PoE will terminate if the epoch duration and the number of the selected users are carefully selected.

Given Figures 4, 6 and 7, 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.

9 Discussion

In this work, we implement the scheme proposed in Localcoin to replace the *computational* hardness that is at the root of Bitcoin's security with the *social* hardness of ensuring that all witnesses to a transaction are colluders (users assisting the malicious user to double spend) [16]. Our design requires users to sacrifice anonymity to maintain reputation scores, which is a threat to privacy due to 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 [65] 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 [66]. 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.

Whenever Mnene 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 occurences. 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.

10 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. Last, advancing on [67] we extended the adversarial analysis and detailed an analysis against double spending attacks.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[2] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "Medblock: Efficient and secure medical data sharing via blockchain," *Journal of medical systems*, vol. 42, no. 8, p. 136, 2018.

[3] S. Popov, "The tangle," p. 131, 2016.

[4] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *Int. Conf. on Financial Cryptography and Data Security*, 2017, pp. 357–375.

[5] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," Cryptology ePrint Archive, Report 765, 2014, https://eprint.iacr.org/2014/765.

[6] 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.

[7] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE wireless comm.*, vol. 20, no. 3, pp. 96–104, 2013.

[8] 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.

[9] 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.

[10] A. Asadi and V. Mancuso, "Wifi direct and lte d2d in action," in *Wireless Days (WD), 2013 IFIP*. IEEE, 2013, pp. 1–8.

[11] 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.

[12] C. Funai, C. Tapparello, and W. Heinzelman, "Enabling multi-hop ad hoc networks through wifi direct multi-group networking," in *IEEE ICNC*, 2017, pp. 491–497.

[13] 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.

[14] ——, "Openrp: a reputation middleware for opportunistic crowd computing," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 115–121, 2016.

[15] Y. Hu, A. Manzoor, P. Ekparinya, M. Liyanage, K. Thilakarathna, G. Jourjon, A. Seneviratne, and M. E. Ylianttila, "A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain," *ArXiv e-prints*, Jan. 2018.

[16] 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.

[17] 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 IoT Journal*, 2018.

[18] M. H. Moti, D. Chatzopoulos, P. Hui, and S. Gujar, "Farm: Fair reward mechanism for information aggregation in spontaneous localized settings," in *IJCAI*, 7 2019, pp. 506–512.

[19] 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.

[20] V. Koutsos, D. Papadopoulos, D. Chatzopoulos, S. Tarkoma, and P. Hui, "Agora: A privacy-aware data marketplace," in *IEEE ICDCS*, 2020, pp. 1211–1212.

[21] C. Niu, Z. Zheng, F. Wu, X. Gao, and G. Chen, "Achieving data truthfulness and privacy preservation in data markets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 1, pp. 105–119, 2018.

[22] 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.

[23] 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.

[24] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Annual International Cryptology Conference*. Springer, 1992, pp. 139–147.

[25] 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.

[26] H. Moniz, N. F. Neves, and M. Correia, "Turquois: Byzantine consensus in wireless ad hoc networks," in *IEEE/IFIP Dependable Systems and Networks*, 2010, pp. 537–546.

[27] 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.

[28] X. Boyen, C. Carr, and T. Haines, "Blockchain-free cryptocurrencies: A framework for truly decentralised fast transactions," Cryptology ePrint Archive, Report 871, 2016.

[29] A. Churyumov, "Byteball: a decentralized system for transfer of value," 2016.

[30] S. D. Lerner, "Dagcoin: a cryptocurrency without blocks," 2015.

[31] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol," Cryptology ePrint Archive, Report 1159, 2016.

[32] Y. Sompolinsky and A. Zohar, "Phantom: A scalable blockdag protocol," Cryptology ePrint Archive, Report 104, 2018.

[33] Y. Li, B. Cao, M. Peng, L. Zhang, L. Zhang, D. Feng, and J. Yu, "Direct acyclic graph-based ledger for internet of things: Performance and security analysis," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1643–1656, 2020.

[34] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[35] S. Micali, "ALGORAND: the efficient and democratic ledger," *CoRR*, vol. abs/1607.01341, 2016. [Online]. Available: http://arxiv.org/abs/1607.01341

[36] H. Finney, "The finney attack(the bitcoin talk forum)."

[37] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When internet of things meets blockchain: Challenges in distributed consensus," *IEEE Network*, vol. 33, no. 6, pp. 133–139, 2019.

[38] P. Szilágyi, "Ethereum p2p protocol, 2020," github.com/ethereum/devp2p/blob/master/caps/snap.md.
[39] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain," in *IEEE MASS*, 2018, pp. 442–450.
[40] 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.
[41] J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," 1999.
[42] D. Chatzopoulos, C. Bermejo, E. u. Haq, Y. Li, and P. Hui, "D2d task offloading: A dataset-based Q&A," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 102–107, 2019.
[43] 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.
[44] 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.
[45] 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.
[46] 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 406, 2017.
[47] 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.
[48] "Proof of burn," https://bitcointalk.org/index.php?topic=1141676.0.
[49] N. Banerjee, M. D. Corner, and B. N. Levine, "An energy-efficient architecture for DTN throwboxes," in *IEEE INFOCOM*, 2007, pp. 776–784.
[50] 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.
[51] Earn.com, "Predicting bitcoin fees for transactions." bitcoinfees.earn.com, 2018.
[52] 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.
[53] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Cluster Computing*, pp. 157–167, 2002.
[54] 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.
[55] H. Krawczyk, M. Bellare, and R. Canetti, "Hmac: Keyed-hashing for message authentication," Tech. Rep., 1997.
[56] 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.
[57] C. Newport, "Gossip in a smartphone peer-to-peer network," *arXiv preprint arXiv:1705.09609*, 2017.
[58] I. Cascudo and B. David, "Scrape: Scalable randomness attested by public entities," Cryptology ePrint Archive, Report 216, 2017.
[59] 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.
[60] M. R. Brust and S. Rothkugel, "Small worlds: Strong clustering in wireless networks," *CoRR*, vol. abs/0706.1063, 2007.
[61] A. Singh *et al.*, "Eclipse attacks on overlay networks: Threats and defenses," in *IEEE INFOCOM*. Citeseer, 2006.
[62] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *IEEE Security and Privacy*, 2017, pp. 375–392.
[63] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2000.
[64] M. Penrose, *Random geometric graphs*. Oxford University Press, 2003, vol. 5.
[65] 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 S&P 2014*.
[66] 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.
[67] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Mneme: A mobile distributed ledger," in *IEEE INFOCOM*, July 2020, pp. 1897–1906.

Dimitris Chatzopoulos is a research assistant professor at The Hong Kong University of Science and Technology (HKUST). Dimitris received his PhD in Computer Science and Engineering from HKUST and his Diploma and Msc in Computer Engineering and Communications from the Department of Electrical and Computer Engineering of University of Thessaly, Volos, Greece. His main research interests are in the areas of mobile computing, device–to–device ecosystems and cryptocurrencies.

Anurag Jain is an undergraduate researcher at the Machine Learning Laboratory@IIITH. As a young researcher, he has diverse interests including Blockchains, Game Theory, Mechanism Design and Economics. He has also co-authored a paper at the AAMAS 2021 and 2 papers in the workshop on Game Theory in Blockchain at WINE 2020.

Sujit Gujar is an Assistant Professor at the Machine Learning Laboratory@IIITH. Prior to this, he was a Sr. Research Associate at Indian Institute of Science. He worked as a postdoctoral researcher at Ecole polytechnique federale de Lausanne (EPFL). He also worked as a research scientist with Xerox Research Centre India where he contributed in developing a technology that enables enterprises to use crowdsourcing as a complimentary workforce. His research interests are Game Theory, Mechanism Design, Machine Learning, and Cryptography applied to modern web and AI applications such as Auctions, Internet Advertising, Crowdsourcing, and multi-agent systems. His doctoral thesis was awarded alumni medal for best doctoral thesis in the Department of Computer Science and Automation at Indian Institute of Science. He was a recipient of Infosys fellowship for his doctoral research. He has co-authored 7 journal publications, 1 book chapter and 50 peer reviewed international conference/workshop papers. He has 11 patents on his name.

Boi Faltings is a full professor of computer science at the Ecole Polytechnique Federale de Lausanne (EPFL), where he heads the Artificial Intelligence Laboratory, and has held visiting positions at NEC Research Institute, Stanford University and the HongKong University of Science and Technology. He has co-founded 6 companies in e-commerce and computer security and acted as advisor to several other companies. Prof. Faltings has published over 300 refereed papers and graduated over 30 Ph.D. students, several of which have won national and international awards. He is a fellow of the European Coordinating Committee for Artificial Intelligence and a fellow of the Association for Advancement of Artificial Intelligence (AAAI). He holds a Diploma from ETH Zurich and a Ph.D. from the University of Illinois at Urbana-Champaign.

Pan Hui (IEEE Fellow and ACM Distinguished Scientist) received his Ph.D degree from Computer Laboratory, University of Cambridge, and earned his MPhil and BEng both from the Department of Electrical and Electronic Engineering, University of Hong Kong. He is currently a faculty member of the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology where he directs the HKUST-DT System and Media Lab. He also serves as a Distinguished Scientist of Telekom Innovation Laboratories (T-labs) Germany and an adjunct Professor of social computing and networking at Aalto University Finland.