MARS: Monetized Ad-hoc Routing System

Bernardo David* Rafael Dowsley† Mario Larangeira*

July 3, 2019

Abstract

A Mobile Ad-Hoc Network (MANET) automatically reorganizes itself, allowing moving nodes to join or leave the network at any point in time without disrupting the communication. An essential element of such systems is a routing protocol able to quickly reorganize existing routes when nodes leave and provide routes to previously unknown joining nodes. In most MANET routing protocols, nodes are assumed to be altruistic, that is, forward incoming packets to the next node in the route. However, as pointed out in a number of previous works, this is a big issue in real world scenarios where nodes are often selfish, *i.e.,* refusing to forward incoming packets from their peers but still using the network infrastructure to route their own packets. In this work, we propose MARS, a blockchain-based reputation system that acts as an overlay on top of existing MANET routing protocols (*e.g.* AODV and OLSR). The main goal of MARS is to keep a publicly available (and verifiable) record of node behavior that can be used to both select good routes and reward nodes that dedicate their resources to routing. As a building block, we propose a compact "proof-of-routing" that allows a node to prove that it has participated in the routing of a given (batch of) packet(s). Upon presenting such a proof, the node is assigned a reputation point that is publicly registered to the blockchain and that can be verified later. Such reputation points are modeled as coins in a cryptocurrency or (more generally) as assets in the blockchain, and as such can be traded for enhanced network services among MANET nodes or traded for other assets (*e.g.* bitcoin) with third parties.

1 Introduction

A mobile ad-hoc network (MANET) allows mobile devices to communicate without any pre-established infrastructure or centralized management. In a MANET, nodes cooperate among themselves to route messages, dynamically adjusting routes as they join, leave and physically move around the network

*Tokyo Institute of Technology. Emails: {bernardo,mario}@c.titech.ac.jp. This work was supported by the Input Output Cryptocurrency Collaborative Research Chair, which has received funding from IOHK.

†Aarhus University and IOHK. Email: rafael@cs.au.dk. This project has received funding from the European research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme (grant agreement No 669255). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731583 (SODA).

area. Such flexibility makes MANETs attractive for applications such as campus networks, disaster relief, emergency communications and providing internet access networks in areas without infrastructure. Moreover, novel *Internet-of-Things* (IoT) applications heavily rely on MANETs. However, a consequence of node mobility and lack of central infrastructure is a previously unknown and constantly changing network topology, which makes it unfeasible to employ traditional routing protocols in MANETs [8].

Providing efficient and reliable routing for MANETs is a challenging task for which a number of protocols has been developed [11, 20]. These protocols can be classified into two main categories [13]: reactive routing protocols, where nodes discover routes only when needed, and proactive routing protocols, where nodes perform a constant route discovery process by periodically exchanging topology information. The Ad-Hoc On Demand Distance Vector (AODV) protocol [19], a reactive routing protocol, and the Optimized Link State Routing Protocol (OLSR) [7], a proactive routing protocol, are well known examples of MANET routing protocols and will be the focus of this work when discussing our proposed solutions.

As it is the case with most networking protocols, MANET routing protocols were not designed with security in mind and are indeed subject to several threats and attacks [14, 23, 11, 13]. Apart from malfunctioning due to broken software or resource overload, nodes that intentionally misbehave in a MANET routing protocol can be classified into two main categories: *malicious* nodes or *selfish* nodes [16]. Malicious nodes deviate from the protocol to mount a number of attacks aiming at disrupting routing operations by causing messages to be routed in dysfunctional ways or by overloading the network. On the other hand, selfish nodes do not purposefully disrupt network operations but refuse to dedicate their resources to route incoming messages while using other nodes' resources to route their own messages.

Many malicious attacks (*e.g.* blackhole, link spoofing and replay attacks) are based on injecting specially crafted (possibly invalid) messages into the routing protocol to cause malfunction [13]. The main reason why these attacks succeed is that MANET routing protocols lack mechanisms to verify message authenticity, leading to the development of a number of solutions based on cryptographic message authentication [11, 10, 12, 21]. However, solely adding authenticity (or even integrity) guarantees to routing protocol messages does not help mitigating selfish behavior, since selfish nodes do not actively conduct attacks against the protocol but refuse to route messages.

A number of heuristics for detecting and isolating selfish nodes have been proposed [17, 6, 5, 16, 2, 1]. Most of them are *reputation*-based solutions, basically providing ways for nodes to measure how much their peers are contributing to routing and keep local records of each other's reliability (*i.e.* reputation). Given this data, nodes can choose which peers to cooperate with. Notice that, in addition to observing the behavior of peers in their vicinity, nodes also rely on external advice for building their reputation records (specially for peers that cannot be reached directly). However, in current reputation-based schemes, each node keeps its own reputation records locally and dishonest nodes can claim that their peers are misbehaving without providing any publicly verifiable proof. These issues affect the accuracy and effectiveness of current reputation systems, which employ complicated heuristics to mitigate false claims of misbehavior and build a cohesive view of reputation among honest altruistic nodes.

1.1 Our Contributions

In this work, we introduce MARS, a system that uses cryptographic tools to build a *publicly verifiable* record of nodes' reputations in MANET routing protocols that can be accessed and verified by any third party (including new nodes that join the network). Moreover, MARS allows nodes to trade their reputation points for other assets, such as (improved) network services and cryptocurrencies. MARS works as an overlay extension to any MANET routing protocol. It stores reputation information in a blockchain-based public ledger [18], which allows nodes to publicly share data with the added guarantee that data becomes immutable once written [9]. In order to provide publicly verifiable reputation information, we introduce the concept of *Proof-of-Routing*, which allows any entity to verify whether a node has participated in the routing of a given message (or batch of messages) and prevents dishonest nodes from claiming extra reputation points or avoiding other nodes form claiming their points. MARS' main characteristics are the following:

- **Publicly verifiability:** Any node or third party can access the reputation record and verify that it is consistent with each node's actual behavior.
- **Resistance against dishonest nodes:** Dishonest nodes cannot abuse MARS to artificially increase their own reputation or decrease a node's reputation.
- **Financial transactions involving reputation:** Reputation in MARS is treated as a digital *asset* that can be traded for other assets, services or digital currencies (*e.g.* Bitcoin [18]).

It is natural to assume that Proof-of-Work based blockchain protocols (*e.g.* Bitcoin) can be too resource intensive for MANET nodes, but there exists lightweight alternatives based on Proof-of-Stake that achieve the same security guarantees [15]. MARS works by awarding a number of reputation points to a node every time a proof-of-routing showing that this node participated in routing a message (or batch of messages) is posted to the public ledger. A proof-of-routing cannot be forged by a node who didn't participate in the protocol nor can a dishonest node disavow another node's proof-of-routing. Hence, well-behaved nodes are guaranteed to receive their reputation points while dishonest nodes cannot artificially receive more points without participating in the protocol. Besides providing a unified (but decentralized) source of reputation for current network nodes, MARS can also be used by external nodes to check which nodes are the most reliable ones before joining the network.

As in a cryptographic currency (*e.g.* Bitcoin), any entity can determine how many reputation points a node has received by inspecting the blockchain and counting how many proofs-of-routing including that node have been posted. Moreover, using standard cryptocurrency techniques, a node can *transfer* reputation points to another party in exchange for cryptocurrency coins, services (*e.g.* increased network speed) or other assets. This flexibility of dealing in reputation points, allows the system to offer financial incentives for altruistic behavior. Moreover, it can also be used as part of a billing system where nodes can trade reputation points for discounts in using network services (*e.g.* an Internet gateway) if they are actively helping run the network.

Besides the blockchain-based public ledger, another core component of MARS is a secure proof-of-routing scheme, which we construct based on *composite signatures* [22]. Similarly to aggregate signatures [4], composite signatures allow many signatures under different secret-keys to be *composed* into a compact representation (the size of a single signature) with the added security guarantee that a signature under a given secret-key cannot be removed from this compact representation once it is composed. The basic idea for building a proof-of-routing is to have each node in the route of a message (including the source and destination nodes) sign the message along with the address of the source and destination nodes, compose its signature with the composite signature received from the previous node and forward both the message and the new composite signature to the next node in the route. Once the message reaches its destination, the destination node posts the final composite signature to the public ledger (*i.e.* blockchain) as a proof-of-routing. The proof-of-routing is considered valid if includes signatures under both the source and destination nodes' secret keys, awarding reputation points to each of the intermediate nodes in that route.

1.2 Related Works

Out of the many works proposing heuristics for mitigating selfish behavior in MANETs, e.g. [17, 6, 5, 16, 2, 1], Ad hoc-VCG [1] stands out due to its use of financial compensation between nodes. Ad hoc-VCG incentivizes nodes to actively participate in routing messages through financial rewards, which are accrued for each message that is routed. A game-theoretic analysis shows that the reward strategy of Ad hoc-VCG results in rational players behaving correctly. However, Ad hoc-VCG only considers rational nodes that either participate or not in order to save energy, offering no security guarantees against dishonest nodes that might actively subvert the financial compensation scheme regardless of energy costs. Moreover, it specifies optimal reward strategies to ensure that rational nodes cooperate but does not describe any concrete scheme for reward distribution. In fact, MARS can be used to concretely instantiate Ad hoc-VCG with added security guarantees against dishonest adversaries, since it provides a mechanism for distributing financial rewards to nodes that is resistant to dishonest behavior.

Another interesting scheme that employs blockchain-based public ledger for rewarding nodes in semi-decentralized networks was suggested in the context of the Tor anonymous routing network by Biryukov and Pustogarov [3]. In this scheme, Tor relays (powerful nodes that aggregate large volumes of traffic) are compensated by users through block mining. Basically, the nodes connected to a given Tor relay act as members of a mining pool, trying to solve proof-of-work puzzles and sending their results to the Tor relay. Since solutions to proof-of-work puzzles are anonymous, no information about the nodes' identity is leaked while the Tor relay is compensated by either using this information to mine blocks itself or participating itself of a large mining pool. However, this scenario is fundamentally different from MANETs since anonymity is a central concern and certain pre-established infrastructure is assumed to exist along with a degree of central management, *i.e.* the Tor relays are known beforehand and nodes are assigned to a specific Tor relay for a long period of time, instead of constantly changing routes as in a MANET.

2 Preliminaries

In this section we present definitions and constructions that will be used throughout the paper. We denote concatenating a string x with a string y by $x \mid y$. When the concatenation operations is used with algebraic objects we assume their binary representation is concatenated.

2.1 Composite Signatures and their Security Definition

A cryptographic building block used in our solution is the *composite signature scheme*, as defined by Saxena et al. [22]. This primitive was derived from aggregate signature scheme [4]. The main similarity between these primitives is that in both cases multiple signatures can be combined into one single and short (aggregated) signature. The feature introduced by [22] is that the aggregation process is *one-way*, namely, given the aggregated signature, it is very hard to compute the individual signatures (or the signatures on any proper subset).

Now, we formally review the definition.

Definition 2.1 (Composite Signature Scheme [22]) *Let a message-descriptor ℓ consist of pairs of message/verification key, i.e., $\ell = \{(m_1, vk_1), \ldots, (m_i, vk_i)\}$. A composite signature scheme* $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Verify}, \mathsf{SIG.Compose})$ *works as follows:*

- $\mathsf{SIG.Gen}$ *gets as input the security parameter λ and output a pair of signing sk and verification vk keys.*
- $\mathsf{SIG.Sign}$ *takes as input a pair of keys (sk, vk) and a message m and outputs a signature σ on the message-descriptor $\ell = \{(m, vk)\}$.*
- $\mathsf{SIG.Verify}$ *takes as input a message-descriptor $\ell = \{(m_1, vk_1), \ldots, (m_i, vk_i)\}$ and a signature σ and outputs a decision bit about the validity of the signature. If a single message is signed under multiple signing keys, the message-descriptor is denoted as $\ell = \{m, vk_1, \ldots, vk_i\}$.*
- $\mathsf{SIG.Compose}$ *takes as input two pairs of message-descriptor/signature, (ℓ_1, σ_1) and (ℓ_2, σ_2). If $\ell_1 \cap \ell_2 \neq \emptyset$ or any of the message-descriptor/signature pairs is invalid according to* $\mathsf{SIG.Verify}$*, it outputs $\perp$; otherwise, it outputs a composite signature σ on the message-descriptor $\ell = \ell_1 \cup \ell_2$.*

The correctness requirement that the composite signature scheme needs to satisfy is the straightforward one. The compactness requirement is that the composite signature has the same size as a single signature. Next, we recall the security definition for composite signature schemes. The key security property is that given the set of valid composite signatures that are available to an adversary, he should be able to compute valid composite signatures only on unions of their message-descriptors. We now review the formal security definition of composite signature[1].

Definition 2.2 (Secure Composite Signatures) *The security game consists of an interaction between the forger $\mathcal{A}$ and a challenger:*

[1] Our formulation of the security definition is more simple and clear than the one in [22], but they are equivalent.

1. **Setup:** *The forger $\mathcal{A}$ chooses n and sends it to the challenger. The challenger runs $(sk_i, vk_i) \xleftarrow{\$} \mathsf{SIG.Gen}(1^\lambda)$ for $i = (1, \ldots, n)$ and then sends the set $VK = \{vk_i\}_{i=(1\ldots n)}$ to $\mathcal{A}$.*

2. **Sign Queries:** *The forger $\mathcal{A}$ makes a polynomial number of sign queries. Each sign query i consists of ℓ_i, a message-descriptor with verification keys from VK. If the pairs in ℓ_i are unique, the challenger responds with a composite signature on ℓ_i, otherwise with the error symbol $\perp$. The challenger adds the message-descriptor ℓ_i to L, the set of message-descriptors in all sign queries.*

3. **Output:** *The forger $\mathcal{A}$ outputs $(\ell_\mathcal{A}, \sigma_\mathcal{A})$ and he wins if the following conditions hold:*

   - $1 \leftarrow \mathsf{SIG.Verify}(\ell_\mathcal{A}, \sigma_\mathcal{A})$;
   - *For $\ell'_\mathcal{A} = \{(m, vk) |\ (m, vk) \in \ell_\mathcal{A} \wedge vk \in VK\}$, it holds that $\ell'_\mathcal{A} \notin \mathcal{P}(L)$, where $\mathcal{P}(L)$ denotes the power set of L.*

A composite signature scheme SIG *is secure if there is no probabilistic polynomial-time forger $\mathcal{A}$ that wins this game with non-negligible advantage in λ.*

It was proven by Saxena et al. [22] that the aggregate signature scheme of Boneh et al. [4] can be transformed into a secure composite signature scheme by appending the verification-key and a random string to the message.

2.2 Blockchain and Public Ledger

Blockchain-based systems had been considered for applications in several different environments beyond its original financial realm. In a nutshell, what these applications have in common is that they rely on a distributed database kept by a network of users. In a peer-to-peer (P2P) network, each user keeps a copy of a single data structure, *i.e.,* the blockchain data-structure, and follows a protocol which rules how the structure will be updated and by whom. Once the participant is chosen, it releases its update information on the network and all the others following the protocol would update its own copy of the data-structure.

The ledger and its properties. Arguably the most famous blockchain-based protocol is the Bitcoin Cryptocurrency which achieves global reach. Namely, anyone in the globe can trade using the system, as long as one has a Bitcoin address and network access. Several other cryptocurrencies exist however all of them follow similar blockchain-P2P design. Naturally, the safety of any cryptocurrency relies on the "quality" of the P2P network kept data-structure. By "quality", the research community observed three fundamental properties. Notably these properties were first formalized by Garay et al. [9], and their respective intuition are as follows:

- **Common-Prefix:** The sequence of blocks has exactly the same blocks among all the copies kept by the participants of the P2P network, but some small part in the newest blocks;
- **Chain-Quality:** The total number of blocks generated by the corrupted parties are a small fraction of the whole sequence of blocks;

- **Chain-Growth:** The blockchain is guaranteed to grow at a minimal rate regarding the rounds of the protocol.

More interestingly in [9], the authors proved that the Bitcoin blockchain implements a *public transaction ledger* by fulfilling the above mentioned properties. In a global scale, this system works as a robust database which keeps a set of *records*, which in the case of a cryptocurrency are transactions. More formally, a *ledger* has two properties:

- **Liveness:** Given a record created by an honest participant and enough time, this record will be inserted into the ledger;
- **Persistence:** Given a record old enough in the copy of a honest participant of the system, we can be assured, *i.e.*, with great probability, that this record will also be in all other copies of the remaining honest players.

From records to reputation. Generalizing the idea from virtual currency to a simple *record* on a database allowed the ledger to be used in a process called *tokenization*. In this process, the blockchain is used to keep track of real assets in the real world, which is particularly interesting for verification purposes by all the participants of the system. In other words, the history of ownership of a given asset can be verified by anyone. The trustful nature of the ledger is particularly interesting in this scenario because it prevents malicious parties from changing the history and therefore stealing or influencing the tracking of the assets.

Such corruption-free set of records is the core tool we use to implement a reputation system, since every transaction can be verified and it is known to be correct by the participants of the system. With such tool, players can scrutinize and argue about the past behavior of each other, consequently reputations can be reasoned, and therefore built, in a public way.

2.3 MANET Routing Protocols

As discussed previously, MANET routing protocols can be classified into two main categories [13]: reactive routing protocols, where nodes discover routes only when needed, and proactive routing protocols, where nodes perform a constant route discovery process by periodically exchanging topology information. Several MANET routing protocols have been developed and are discussed in surveys such as [20, 11, 13]. For the sake of completeness, we will give a brief description of the Ad-Hoc On Demand Distance Vector (AODV) protocol [19] and the Optimized Link State Routing Protocol (OLSR) [7].

- **AODV:** a reactive protocol that requires a source node S who wants to send a packet to a destination node D (but does not know a route) to initiate route discovery by broadcasting a route request $RREQ$ to its neighbors. Each neighbor who receives the $RREQ$ broadcasts it to its own neighbors until $RREQ$ reaches the destination node D. Once D receives $RREQ$ it sends a reply $RREP$ back to S through the same route through which $RREQ$ arrived and ignores future copies of the same $RREQ$. If a route is not used in a while, it expires and route discovery has to be executed again.

- **OLSR:** a proactive protocol that constantly assesses the current network topology to determine the best routes. In OLSR, each node periodically broadcasts to its neighbors a $HELLO$ message containing its own address and the list of its single-hop neighbors. After receiving $HELLO$ messages from all of its single-hop neighbors, each node learn the network topology up to two hops. Next, each node selects a set of single-hop neighbors large enough to provide routes to all of its two hop neighbors to act as its Multipoint Relays (MPR). After MPRs are selected, only the MPR nodes start broadcast Topology Control messages TC containing the list of all other nodes that selected that node as a MPR. Upon receiving a TC message from other MPRs, an MPR node broadcasts the TC message to its neighbors. Through this process, all nodes learn the topology of the network and can compute routes to all other nodes.

3 Threat Model

We consider that the nodes can be selfish in relation to the execution of the routing protocol. This means that they do not disrupt the network operations on purpose. However, each node potentially tries to avoid using its own resources to route messages from other nodes while using the resources from other nodes' to route its own messages. This is the case of traditional selfish behavior where the nodes have no wish to disrupt communications but do wish to save their own resources, as defined in [16].

In addition, we do consider that the nodes can try to act dishonestly in the execution of the reputation system in order obtain advantages without being flagged as selfish. The nodes are modeled as probabilistic polynomial-time (PPT) Turing machines. With these added adversarial powers, we augment the usual model of selfish behavior [16] to capture real world situations where dishonest nodes still do not wish to disrupt communications but will adopt adversarial strategies to subvert systems that detect and isolate/punish selfish behavior [17, 6, 5, 16, 2, 1]. We call such adversarial nodes *dishonest*, in contrast to fully malicious or simply selfish nodes, as defined in [16].

We consider the case in which there is no collusion among the different dishonest nodes, meaning that dishonest nodes act individually to gain advantages but do not perform coordinated attacks. We argue that this restriction on collusion seems natural in highly mobile scenarios where nodes join and leave the network (or move around) very frequently, without having sufficient time to discover other dishonest nodes and coordinate. While there are real world situations where collusion can be achieved, we leave these scenarios as a future work.

We wish to build a reputation system that awards reputation points to nodes that truly cooperate in the routing protocol with security against dishonest nodes as defined above. The first property we wish to obtain in a secure reputation system is that no dishonest node can gain reputation points without presenting a valid proof-of-routing in which he was part of the route. The second property is that no dishonest node can decrease another node's reputation (*e.g* by presenting false data).

4 Building MARS

The main goal of MARS is to keep a publicly verifiable record of reputation information for MANET nodes while resisting attacks by dishonest nodes who aim at maliciously increasing their own reputation or decreasing other nodes' reputations. Reputation points are stored in a blockchain-based public ledger in such a way that they can be used in transactions between users of the ledger as in a cryptocurrency. To that end, we introduce a proof-of-routing scheme (detailed later in this section) that allows for nodes (and third parties) to verify whether a given node has participated in the routing of a given message (or batch of messages). Each node receives a number of reputation points proportional to the amount of valid proofs-of-routing posted on the blockchain showing that it has participated in routing messages (*i.e.* proportional to the number messages it has helped route).

MARS is run in conjunction with an arbitrary MANET routing protocol, requiring that each node adds extra information to the message (*i.e.* adding proof-of-routing information) before proceeding with the regular routing operations. Once a proof-of-routing showing participation is posted in the public ledger, the nodes who participated in the routing (*i.e.* are included in the proof-of-routing) are awarded reputation points. We assume a Public-Key Infrastructure as setup, that is, we assume that MANET nodes know each other's public-keys. This is necessary in order to verify proofs-of-routing. This assumption might seem far fetched in a MANET environment where nodes constantly join and leave the system and where there is no central certificate authority to keep track of keys. However, notice that we have access to a public ledger where nodes can publicly register their public-keys upon joining the network. The basic blue print for MARS is the following:

- **Proof-of-Routing:** The proof-of-routing is basically a composite signature by each node that participates in the route (including the destination node) on the message that is being routed, along with a signature by the source node on the message, its own address and the destination node's address. In order to save space and achieve the desired security guarantees, each node in the route composes its signature with the initial signature generated by the source node, forming a small composite signature that contains each signature of the proof-of-routing. Each node forwards both the message and the composite signature containing the previous signatures along with its own. Additionally, each node checks the validity of the composite signature it has received from the previous node, aborting if it is not valid. By the security properties of composite signatures and since there is no collusion among the nodes, no node can remove the signatures of the previous nodes from the composite signature it receives, and thus cannot decrease the number of reputation points received by honest nodes. For a given route from node N_0 to node N_d, we denote the source node as N_0, the destination node as N_d and the intermediary nodes in the route as $N_1, \ldots, N_{d-1}$. Abusing notation, we also denote each nodes address the same way. More specifically a proof of routing is generated and verified as follows:
  - **All nodes:** Each N_i runs $\mathsf{SIG.Gen}(1^\lambda)$ obtaining a pair of signing sk_i and verification vk_i keys (sk_i, vk_i), and posts vk_i to the public

ledger.

- **Source node:** N_0 signs a hash of its message $H(m)$ concatenated with its address N_0 and the address of the destination node N_d, obtaining a signature $\sigma_{0,m} = \mathsf{SIG.Sign}(sk_0, vk_0, N_0 \mid N_d \mid H(m))$. N_0 sends both $N_0 \mid N_d \mid Hm$ and $\sigma_{0,m}$ to N_d through the underlying routing protocol (*i.e.* following the procedures of the routing protocol for forwarding this message to node N_1).
- **Intermediary nodes:** For $i = 2, \ldots, d-1$, each node N_i verifies previous nodes' signature by computing $\mathsf{SIG.Verify}(N_0 \mid N_d \mid H(m), vk_0, \ldots, vk_{i-1}, \sigma_{i-1,m})$, computes a signature $\sigma'_{i,m} = \mathsf{SIG.Sign}(sk_i, vk_i, N_0 \mid N_d \mid H(m))$, and composes it with the previous signature $\sigma_{i-1,m}$, obtaining a composite signature $\sigma_{i,m} = \mathsf{SIG.Compose}(N_0 \mid N_d \mid H(m), vk_{i-1}, \sigma_{i-1,m}, vk_i, \sigma'_{i,m})$. N_i proceeds with the underlying routing protocol forwarding $N_0 \mid N_d \mid m$ along with its fresh composite signature $\sigma_{i,m}$.
- **Destination node:** N_d performs the same actions as the intermediary nodes. Additionally it posts its final composite signature $\sigma_{d,m}$ along with $N_0 \mid N_d \mid H(m)$ to the public ledger. N_d is also required to post a list of the nodes that participated in routing the message, which can be accomplished in different ways as discussed later in this section.
- **Verifier:** Any party V who wants to verify a proof of routing uses the list of nodes that participated in the routing of the message along with $N_0 \mid N_d \mid H(m)$ to verify the validity of the composite signature $\sigma_{d,m}$ (verification keys for each node can be retrieved from the public ledger itself).

- **Registering Reputation Points:** Once the destination node receives the message, it signs it as well and composes its signature with the composite signature received from the previous node. The destination node posts the final composite signature on the public ledger (*i.e.* blockchain) along with the list of nodes that are part of the route. Any other node (or third party) can later verify that a given node has indeed participated in the routing by verifying that the composite signature is valid. Current and future nodes can verify in how many message routing procedures each node was involved by reading the proofs-of-routing in the blockchain, determining the most reliable nodes in the network. The amount of points awarded per proof-of-routing can be adjusted proportionally to the amount (or size) of messages concerned in the proof.

- **Transactions with Reputation Points:** Each node's reputation points are linked to their public-keys since they are represented as composite signatures. In other words, a node is considered to have reputation points connected to a given proof-of-routing if that proof-of-routing is valid and contains its public-key as part of the message-descriptor associated with the composite signature. Hence, transactions with reputation points can be carried out using the same transaction scheme as Bitcoin. In order to transfer a reputation point to another entity, a node uses his public-key to sign a Transfer Transaction specifying both the public-key of the

entity who will receive the points and the proof-of-routing that awarded those points. This transaction is considered valid if the proof-of-routing specified as the source of reputation points is valid and contains that user's public-key in the message-descriptor; and the transaction is signed under the same key.

Security Analysis and Collusion: First we argue that a dishonest node cannot prevent a honest node (who participated in routing a message) from receiving its reputation points. It is clear that dishonest nodes cannot remove an honest node's signature from the intermediate composite signature they receive during the routing process or from the final composite signature posted in the public ledger. This follows immediately from the security properties of composite signatures presented in Section 2.1 and from the fact that there are no collusion among nodes. Notice that this also means that a dishonest node cannot decrease the number of reputation points another node has received in the future, since the proofs-of-routing awarding points to a node become immutable once posted to the public ledger (by the Persistence propety of public ledgers presented in Section 2.2).

Next, we argue that dishonest nodes cannot gain reputation points without participating in routing messages. In our threat model we assume that dishonest nodes cannot collude and perform coordinated attacks, having to act alone in trying to subvert the reputation system. In this scenario, all nodes are forced to generate a correct proof-of-routing and send both this proof and the message to the next node in the route. If a dishonest node fails to send the message or provides an invalid partial proof-of-routing (*i.e.* composite signature including signatures of all nodes up to the curent node in the route), the transmission is aborted. Hence a node has to follow the protocol so that the message reaches the destination and points are awarded. In case of collusion, a group of dishonest nodes who coordinate an attack would be able to simply share their secret-keys and sign all of the messages routed by each of them in the names of each other, effectively adding all of them to the proof-of-routing. However, this strategy requires that the secret-keys of all of these nodes are revealed to each other, allowing one of the dishonest nodes to transfer all of their reputation points to an arbitrary address, which it can do by signing transactions under each colluding node's secret-key.

Selecting a Blockchain Protocol: MARS relies on a blockchain-based public ledger to store reputation data. MANET nodes executing MARS are responsible for executing the underlying blockchain consensus protocol so that data can be added to the ledger. Notice that any blockchain consensus protocol can be used to construct the public ledger as long as it fulfills the security guarantees laid down in [9] and summarized in Section 2.2. Since this blockchain is maintained by the MANET nodes, which are usually resource constrained, it is a clear issue to use Proof-of-Work based blockchains (such as Bitcoin). Notice, however, that there are alternatives, such as Proof-of-Stake based blockchain protocols that achieve the same security guarantees [15].

Improving Efficiency in Congested Networks: It is clear that generating individual proofs-of-routing for every message (or packet) transmitted in

the network would generate large communication and computational overheads since the number of messages can be very high. Moreover, each proof-of-routing added to the public ledger has to be broadcast by the blockchain protocol. A simple way to solve this problem is to generate proofs-of-routing for batches of messages (or packets) that are sent through the same route between the same source and destination nodes. In this case, a minimum number of routed messages is set and a proof-of-routing for that batch of messages in generated after that number is achieved.

Deploying MARS with Reactive and Proactive Routing Protocols: The proof of routing mechanism requires any parties who want to verify a proof to know the nodes that participated in routing the message (or batch of messages) associated to that proof. Hence, this information must be posted along with the proof itself in the public ledger. When using MARS on top of reactive routing protocols (*e.g.* AODV), where routes might change for every message that is transmitted between a source node and a destination node, it might be necessary to post full route information describing all nodes that participated in routing to the public ledger, which can cause a high sotrage overhead. However, if MARS is deployed on top a proactive routing protocol (*e.g.* OLSR), a routing table is known for each node. In this case, the protocol can be modified to require each node to first post its routing table to the public ledger, later including a reference to specific routes in the table in each message that it sends. Intermediary nodes will then only route a message if they are part of the route referenced in the message. When the routing table changes, the node can simply post an update to the public ledger specifying only the new route and the old route that it supersedes, instead of posting the full new routing table. Such a modification amortizes the storage overhead in the public ledger, since individual route information does not have to be posted for every single proof of routing.

5 Conclusion

We have introduced MARS, a publicly verifiable reputation system to account for node participation in routing messages in a MANET. Differently from previous schemes, MARS keeps public records by means of a blockchain-based public ledger. Moreover, we show that dishonest nodes cannot influence the reputation system, *i.e.* they cannot maliciously increase their own reputation or decrease the reputation of other nodes by presenting false data. An interesting feature of MARS is that it allows nodes to conduct transactions involving transaction points as in a cryptocurrency. MARS reputation points registered in the public ledger can be traded by other assets such as cryptocurrency coins and network services.

In order to achieve these results, we introduce the notion of Proof-of-Routing, a cryptographic scheme that allows nodes to prove that they have participated in the routing of a given message in a publicly verifiable way. We introduce a construction of proof-of-routing based on composite signature schemes and show that dishonest nodes cannot generate a fake proof-of-routing to trick nodes into believing they have more reputation points. Furthermore, we show that

dishonest users cannot tamper with proofs-of-routing in order to prevent honest nodes from earning their points.

5.1 Open Problems

Our results are analysed in a threat model that considers dishonest adversaries who do not disrupt the routing protocol on purpose but avoid employing their on resources in routing other nodes' messages, possibly subverting the reputation system in order not to be identified as selfish and earn more points. We argue that this model captures real world scenarios. However, we consider that dishonest nodes do not collude, that is, they do not perform coordinated attacks, only attempting to subvert the reputation system individually. This restriction on collusion seems to be reasonable in highly mobile scenarios where network nodes constantly join and leave the network. Nevertheless, it is an interesting open problem to propose a reputation system with similar guarantees as ours in a threat model with colluding adversaries that can perform coordinated attacks.

We have introduced a theoretical description of MARS along with possible applications, highlighting the fact that reputation points can be traded as in a financial system. This setting gives rise to a number of possible applications of mechanism design in designing optimal financial incentive strategies that result in rational nodes cooperating in the routing protocol, as in the Ad hoc-VCG protocol [1]. Further investigations on designing optimal incentive mechanisms are left as an open problem.

Even though MARS employs public-key cryptographic primitives, it is reasonable to assume that current devices (*e.g.* mobile phones) are able to handle the processing loads involved in the required computations. However, experimentally evaluating the performance of our schemes in a real world scenario is an interesting future work. It is specially interesting to measure the performance of blockchains operating in MANETs and other possible applications of such distributed "mobile" consensus protocols.

References

[1] Anderegg, L., Eidenbenz, S.: Ad hoc-vcg: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking. pp. 245–259. MobiCom '03, ACM, New York, NY, USA (2003), http://doi.acm.org/10.1145/938985.939011

[2] Balakrishnan, K., Deng, J., Varshney, V.K.: Twoack: preventing selfishness in mobile ad hoc networks. In: IEEE Wireless Communications and Networking Conference, 2005. vol. 4, pp. 2137–2142 Vol. 4 (March 2005)

[3] Biryukov, A., Pustogarov, I.: Proof-of-work as anonymous micropayment: Rewarding a tor relay. In: Böhme, R., Okamoto, T. (eds.) FC 2015. LNCS, vol. 8975, pp. 445–455. Springer, Heidelberg (Jan 2015)

[4] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (May 2003)

[5] Buchegger, S., Boudec, J.Y.L.: A robust reputation system for mobile ad-hoc networks. Tech. rep., Proceedings of P2PEcon (2003)

[6] Buchegger, S., Le Boudec, J.Y.: Performance analysis of the confidant protocol. In: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing. pp. 226–236. MobiHoc '02, ACM, New York, NY, USA (2002), http://doi.acm.org/10.1145/513800.513828

[7] Clausen, T., Jacquet, P.: Optimized link state routing protocol (olsr) (2003)

[8] Corson, S., Macker, J.: Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations (1999)

[9] Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 281–310. Springer, Heidelberg (Apr 2015)

[10] Hu, Y.C., Johnson, D.B., Perrig, A.: Sead: secure efficient distance vector routing for mobile wireless ad hoc networks. In: Proceedings Fourth IEEE Workshop on Mobile Computing Systems and Applications. pp. 3–13 (2002)

[11] Hu, Y.C., Perrig, A.: A survey of secure wireless ad hoc routing. IEEE Security and Privacy 2(3), 28–39 (May 2004), http://dx.doi.org/10.1109/MSP.2004.1

[12] Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: A secure on-demand routing protocol for ad hoc networks. Wireless Networks 11(1), 21–38 (Jan 2005), https://doi.org/10.1007/s11276-004-4744-y

[13] Kannhavong, B., Nakayama, H., Nemoto, Y., Kato, N., Jamalipour, A.: A survey of routing attacks in mobile ad hoc networks. IEEE Wireless Communications 14(5), 85–91 (October 2007)

[14] Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. In: Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003. pp. 113–127 (May 2003)

[15] Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. Cryptology ePrint Archive, Report 2016/889 (2016), http://eprint.iacr.org/2016/889

[16] Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. pp. 255–265. MobiCom '00, ACM, New York, NY, USA (2000), http://doi.acm.org/10.1145/345910.345955

[17] Michiardi, P., Molva, R.: Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and

Multimedia Security: Advanced Communications and Multimedia Security. pp. 107–121. Kluwer, B.V., Deventer, The Netherlands, The Netherlands (2002), http://dl.acm.org/citation.cfm?id=647802.737297

[18] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)

[19] Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (aodv) routing (2003)

[20] Royer, E.M., Toh, C.K.: A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications 6(2), 46–55 (Apr 1999)

[21] Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: A secure routing protocol for ad hoc networks. In: Proceedings of the 10th IEEE International Conference on Network Protocols. pp. 78–89. ICNP '02, IEEE Computer Society, Washington, DC, USA (2002), http://dl.acm.org/citation.cfm?id=645532.656326

[22] Saxena, A., Misra, J., Dhar, A.: Increasing anonymity in bitcoin. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) FC 2014 Workshops. LNCS, vol. 8438, pp. 122–139. Springer, Heidelberg (Mar 2014)

[23] Yang, H., Luo, H., Ye, F., Lu, S., Zhang, L.: Security in mobile ad hoc networks: challenges and solutions. IEEE Wireless Communications 11(1), 38–47 (Feb 2004)