A CONSENSUS-BASED APPROACH TO REPUTATIONAL ROUTING IN MULTI-HOP NETWORKS

Edward Staddon[1], Valeria Loscri[1], Nathalie Mitton[1]
[1]Inria, Lille, France

NOTE: Corresponding author: Edward Staddon, edward.staddon@inria.fr

Abstract – *When it comes to the security of the Internet of Things (IoT), securing their communications is paramount. In multi-hop networks, nodes relay information amongst themselves, opening the data up to tampering by an intermediate device. To detect and avoid such malicious entities, we grant nodes the ability to analyse their neighbours behaviour. Through the use of consensus-based validation, based upon the blockchain's miners, all nodes can agree on the trustworthiness of all devices in the network. By expressing this through a node's reputation, it is possible to identify malicious devices and isolate them from network activities. By incorporating this metric into a multi-hop routing protocol such as AODV, we can influence the path selection process. Instead of defining the best route based upon overall length, we can choose the most reputable path available, thus traversing trustworthy devices. By performing extensive analyses through multiple simulated scenarios, we can identify a decrease in packet drop rates compared to AODV by $\approx$ 48% and $\approx$ 38% when subjected to black hole attacks with 30 and 100 node networks respectively. Furthermore, by subjecting our system to varying degrees of grey holes, we can confirm its adaptability to different types of threats.*

Keywords – Consensus, cybersecurity, IoT, reputation, routing

1. INTRODUCTION

The Internet of Things (IoT) has become part of our everyday lives, providing services in multiple areas. From "Smart" equipment to wearable healthcare devices, the IoT processes a lot of important and sensitive data. Furthermore, as is the case with wearable healthcare devices such as a pacemaker, by allowing a connection with the open Internet, we also open the corresponding attack surface to new threats [1]. This can result in the loss of sensitive data and can even go as far as to cause significant health risks to the patient. In some use cases such as smart agriculture, IoT devices must operate in hostile environments where a direct connection with a base station or access point is not always available. To maintain communications, these devices employ the multi-hop paradigm, allowing intermediate nodes to transmit and relay passing packets to their destination. However, in doing so, we also increase the chance of attack, as any node in our network could compromise our routing activities [2].

One way to provide an extra layer of security is allowing nodes to only converse with neighbours that they trust. The notion of trust is deeply embedded in the human psyche and is a main contributor to how we form relationships. The parameters of how trust is defined varies from person to person, however, a fundamental element is the notion of reputation, where the higher the reputation, the more likely we are to trust said person or entity. Indeed, although the reputation influences the trust value, the opposite is also the case, where breaking someone's trust severely impacts that person's reputation. By rendering the reputation of someone or something common knowledge, any change will be perceived by everyone, meaning that any impact will have inevitable repercussions. This system can be applied to the digital networking world where nodes possess a known reputation value, allowing their neighbours to determine if they can be trusted. As a result, in a similar fashion to human interactions, if a node acts badly in the network, their reputation will decrease, allowing easy separation between malicious entities and good trustworthy nodes.

In multi-hop IoT networks, nodes are generally left to their own devices, operating as configured and routing data when needed. This means, there is no shared memory between devices, meaning that data must be actively provided to each node for them to know it. This is important since as we said previously, the reputation values are known by all nodes in the network. A well-known method for sharing data in a distributed manner whilst maintaining data integrity is through the use of a blockchain [3]. Made popular through its uses in many different cryptocurrencies, such as the infamous Bitcoin [4], a blockchain brings many elements to the table which can be of use. The blockchain employs devices known as "miners" which are responsible for the creation, validation and addition of new data in the form of blocks, into the chain itself. These miners employ a *Proof of Work* (*PoW*) technique for block validation, ensuring that only valid blocks get input into the blockchain, reducing the risk of incorrect data injection.

To allow data to traverse multi-hop networks, many routting protocols exist, each with their own advantages. By incorporating the newly acquired knowledge of node reputation thanks to blockchain, intermediate nodes are now capable of not only determining the trustworthiness of their neighbours, but also influencing their routing abilities. Many routing protocols use various metrics to determine the best route to take towards the destination which could be influenced by a nodes reputation. This is the case of the Ad hoc On-Demand Distance Vector (AODV) routing protocol, where the route with the lowest hop count is preferred [5]. Being a reactive protocol, route discovery is only performed when needed, meaning accurate up-to-date reputational values can be used. During route discovery, the source node broadcasts a Route Request (RREQ) packet, asking for a route towards the destination. This packet is relayed by each node it encounters, each one increasing the hop count by one, until the requested destination is reached. The destination then responds back via unicast towards the source with a Route Reply (RREP) using the shortest route available. By analysing the trustworthiness of each node, we can influence the *hop-count* to increase the corresponding ”length” the more malicious nodes are present. As a result, AODV would naturally select the shortest route, only here this doesn’t correspond to the least number of hops, but the highest trustworthiness overall.

In this paper, we propose a consensus-based module for routing protocols using reputation metrics to determine the most trustworthy route in the network. The main contributions are as follows:

- Firstly, we perform an analysis of previous work in the literature around the notion of ”reputation” as well as different uses of blockchain, in particular their applications to wireless routing activities. We also explore the different security improvements which have been proposed for AODV in recent years.
- Next, we define and propose updated metrics based on previous work for the computation of nodes’ reputation, as well as the addition of a *Reputation Decay* system, allowing nodes to be reintegrated into the network after a certain period of inactivity. We also explain how a consensus-based configuration inspired from the blockchain’s miners which allows us to grant the network the ability to adapt and determine these values without prior knowledge, before sharing the results throughout the network thanks to blockchain technology.
- We also present how our system can be incorporated into a reactive routing protocol, in this case AODV as well as a few updates to the existing protocol, allowing our system to function at peak efficiency.
- Finally, we analyse the performance of this new protocol, called *AODV-Miner*, by comparing it to basic AODV functionality in extensive simulations with networks of 30 and 100 nodes with varying network topologies. By pitching both protocols against black and grey holes with varying degrees of malicious presence and intentions, we demonstrate a reduction in packet drop rates by $\approx$ 48% and $\approx$ 38% with 30 and 100 nodes respectively.

The rest of this paper is organised as follows: Section 2 analyses previous work in the areas of reputation, blockchain and AODV security and presents the differences with our module. Section 3 defines our system model, before presenting our module and *AODV-Miner* in Section 4. Then, Section 5 explains our implementation and simulation parameters before analysing the results in Section 6. Finally, we discuss these results and future endeavours in Section 7 before concluding this paper in Section 8.

2. RELATED WORK

Our system is based around two distinct elements: reputation and blockchain; and it also uses a third in our analysis: AODV. Each of these notions are not new and have been extensively evaluated in the scientific literature. Furthermore, AODV has seen many new propositions to upgrade its functionality and security since its elaboration. However, as far as we are aware, none use a dynamically elected consensus-based reputation system, derived from blockchain’s miners. In this section we present these three elements as well as an analysis of some of the improvements they have received and their uses in routing activities before defining our system and its differences.

2.1 Behavioural reputation

Inspired from the human psyche, the notion of reputation can be applied to an IoT network, where nodes will choose a higher, more reputable neighbour over others. This is the case of [6] where the authors use trust-based methods to identify nodes in the network, based on their previous activities. By evaluating multiple types of activities based on node social interactions and QoS, the resulting trust profiles are evaluated by other nodes before being adopted. In a similar fashion, [7] integrates this functionality into their routing protocol for wireless sensor networks, where they compute a trust value per node, based upon their previous activities. By analysing their sincerity in forwarding data, acknowledging previous packets, as well as the nodes’ energy consumption, this value is then used to determine the most trustworthy candidate to relay the data throughout the network. However, reputation and trust metrics can be expressed in multiple fashions. For example, the authors of [8] evaluates neighbouring behavioural patterns using inter-node cooperation. On the other hand, the authors of [9] use a signature-based methodology, validating data integrity and confirming if data has reached the intended sink.

2.2 Blockchain-based sharing

The main advantage of the blockchain is its immutability [10], which has led it to being used in many other areas, such as that of IoT security [11]. However, they possess many challenges related to the specific context of the IoT, such as resource limitations and data management where power hungry *PoW* and block storage become a problem. That being said, the blockchain has seen its fair share of attention in the area of security, such as providing authentication and trust services to the IoT [12] and increasing data integrity and authenticity [13]. Since our interests revolve around routing, we concern ourselves with the different methods employed to increase routing security [14].

An example is the work performed by the authors of [15]. Here the blockchain stores information related to data transmission, allowing all nodes to participate in determining the ”legality” of the exchanges. In [16], the authors use blockchain to store and share the status of the network in real time to enhance the routing process. By checking the list of transactions, nodes can determine the most efficient route, thus avoiding congested areas and nodes. This technology has also been used in unmaned aircraft systems as in [17], improving both routing activities and authentication. Here, a lightweight blockchain deployment is used, providing each drone with identification and authentication information. The authors of [18] propose a novel routing protocol based on blockchain contractual methodology. By using the ledger to store smart contract addresses indicating when routing is needed, routes can be offered and determined when needed.

2.3 AODV routing protocol

AODV-related security has been an interest in the literature for some time since its original conception. Indeed, AODV is susceptible to multiple types of attacks [19] targeting packet control fields, such as source and destination IP or sequence numbers, as well as hop-count forging. As a result, the authors of [19] propose an intrusion detection system capable of detecting and countering these vulnerabilities by comparing the network's activities to predefined specifications where any deviation is considered malicious. The authors of [20] take a different standing point, directly targeting certain vulnerabilities in an effort to enhance the overall security. Their intrusion detection model allows the detection of multiple attacks, such as denial-of-service, impersonation or a compromised node, which is then isolated from network activities by the Intrusion Response Models. In all, their approach is capable of increasing the routing efficiency, rendering AODV more robust, as the slight cost of a higher overhead. In [21], the authors use advanced numerical analysis to increase the security of AODV during routing. By using methods such as cryptography or numerical sequences, they are able to increase the overall performance when subjected to black-hole attacks.

Reputation-based metrics and blockchain have also been used in line with AODV. Indeed, in [22], the authors extend the AODV-UU protocol to incorporate reputation-based metrics, identifying malicious and trustworthy nodes. By integrating the reputation value directly into the discovery process, it is possible to identify paths passing through malicious nodes, allowing them to be avoided. Regarding blockchain, the authors of [23] propose the protocol BAODV, using blockchain's hash chaining to authenticate nodes and confirm data integrity. By incorporating the IP address of malicious nodes in the discovery messages, BAODV can circumnavigate the malicious entities. Another approach used in [24] is the construction of a blockchain network, allowing the identification of routes towards the destination. Each path node is added to the blockchain network, avoiding malicious entities and identifying the most optimal route to take. In [25], the authors unite both elements, using reputation-based metrics to influence routing activities and the blockchain to distribute the reputation throughout the network. Their approach includes an extension to the reputation metric where the length of a route is manipulated depending on the node's reputation, lengthening it if they possess malicious tendencies. In regards to blockchain dissemination, the authors also define specific network grids in which miners are identified and are responsible for the computation of the reputation and blockchain distribution. This approach allows the type of node to be exploited, privileging powerful nodes for this role over weaker counterparts. However, once nodes have been defined as miners they cannot partake in routing activities, which reduces the number of potential relays in the network.

2.4 Our contribution

To define our system, we take inspiration from multiple approaches, in particular [25]. However, one major difference is that our module is not directly integrated into a specific routing protocol, but can be adapted to fit others, influencing and exploiting the route discovery and upkeep functionalities. By doing so, we allow the ability to dynamically build a route profile, meaning no prior knowledge of the network or nodes is needed. Furthermore, by updating the previously analysed reputation- based approaches to use this dynamic route profile, we allow nodes to identify activities which distinctly deviate from the expected, the main advantage of which is no need for any advanced or heavy techniques. We also define a lightweight version of blockchain, similar to [17], significantly reducing its role to that of a dissemination tool with lower weight and complexity. We also repurpose its miners to perform behavioural validation responsibilities, similar to [25], however, we include the addition of dynamic role selection, allowing nodes to take on the role of miners or routers at will. By not defining specific roles at the start, the network can, therefore, adapt to fluctuating typologies and also take advantage of new nodes with no user intervention needed. This paradigm also redefines the resource-intensive *PoW* process, into a consensus-based validation system, allowing nodes to select the best results to be shared throughout the network. As a result, our new *Validation Miners* differ significantly from their blockchain counterparts, all the while holding key positions in the network.

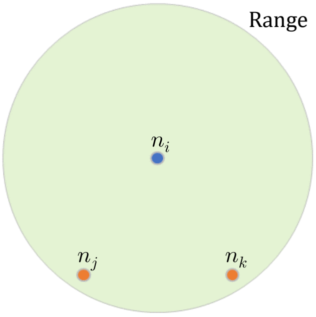


Fig. 1 – Communication range of node n_i

3. SYSTEM MODEL AND THREAT TAXONOMY

Our system is based around specific models and threat information. In this section we explore both our network and validations models, before taking a look at our threat taxonomy.

3.1 Network model

We consider an interconnected wireless network scenario with N static nodes, each possessing a fixed transmission range. Each node has at least one other node in communications range, called a neighbour, forming a partial mesh topology, an example is shown in Fig.1. We can see that node n_i possesses a fixed transmission range, encompassing two other nodes, its neighbours. These interconnections allow any one node to contact all others in the network, resulting in both stable connections and durable routes. As we can see in the figure, multiple nodes can be in range of multiple others. By using the wireless medium, we accept that it is possible for inevitable transmission overlaps to occur, resulting in areas of collision. Our choice of using AODV as a base for our system means that the nodes already take on certain characteristics which are useful to our system. For a reactive protocol to function correctly, all participating devices must be capable of receiving any routing-related traffic at any given time. As a result, we consider that all nodes remain in an active listening state, constantly analysing all passing packets waiting for a potential AODV discovery message. Our nodes also possess the ability to decide on their own role per participated route, making them either a routing node (forwarding information along the corresponding route), or a validation miner (observing and confirming the routing activities of neighbouring routing nodes for the same route). Both roles are mutually exclusive for each route, meaning a miner cannot participate in routing activities, as this would be a conflict of interest. With the additional ability of being able to participate in multiple routes simultaneously, the nodes can, therefore, take on multiple roles.

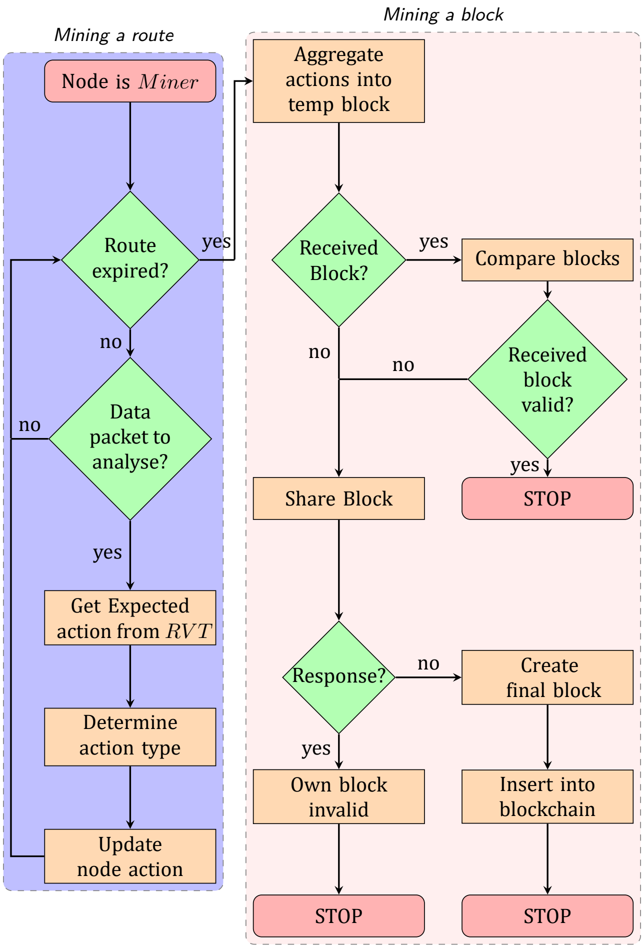


Fig. 2 – Validation flowchart

3.2 Validation model

As stated previously, each and every node has the ability to become a validation miner and, as a result, participate in validation activities. The role of these miners is twofold, illustrated in Fig.2:

1. They are responsible for validating routing behaviour between their neighbours, which we define as ”*mining a route*”.
2. They confirm and distribute the resulting behavioural analysis throughout the network in blockchain form, which we define as ”*mining a block*”.

To reach their first objective, *mining a route*, the miners must possess the ability to validate the behaviour of their neighbours. This is achieved by allowing all nodes to overhear and analyse passing RREP packets, from which each miner can extract the expected forward ($src \rightarrow dst$) and reverse ($dst \rightarrow src$) hops. These are then added to their respective *Route Validation Tables* (*RVTs*), allowing the miners to verify all passing data packets along the corresponding route, thus immediately detecting when a deviation occurs. Upon overhearing a network transaction, the miner classifies the resulting communication as either *Good* or *Bad*, depending if the activity was expected or not.

A more in-depth distinction between the two activities is presented below. Fig.1 depicts this process where since nodes n_j and n_k are in n_i' s transmission range, n_i is in a position to overhear all of their messages. All activities are accumulated and stored for each neighbouring node of the mined route. As stated previously, with wireless transmission comes the possibility of collisions or jamming attacks. As a result, it is possible that miners end up in the overlapping transmission zones, meaning they cannot correctly perform their activities. Since this is a general wireless issue, we address this problem for the miners to the best of our ability, through the possibility of multiple miners per route. This means that multiple miners can overhear and validate the same nodes, decreasing the chance of all being jammed, increasing the efficiency and resiliency of our system.

Once the route expires from the routing tables, the miners transition into their second activity: confirmation and dissemination, visible on the right of Fig.2. To begin, each miner aggregates all results for each node in communications range for that route into a temporary block. These blocks are shared amongst surrounding miners which all partake in the confirmation process. As a result, only blocks confirmed by consensus are deemed valid and disseminated throughout the network via the blockchain. We use blockchain here as it provides a secure means for both confirming and sharing the different blocks. However, our lightweight version, although following the basic blockchain principal, differs in certain aspects. The main difference is the adaptation of the *Proof of Work* for block confirmation, where here miners simply compare the received block with their own, only responding if a difference has been detected. This approach keeps the notion of consensus, where the most common block will be kept, all the while reducing network traffic between miners. As a result, a miner having transmitted their block and not received a response deems their own block valid, incorporating it into the blockchain and disseminating throughout the network. The resulting blocks permit all nodes to update the reputation for all participating nodes. It is, however, important to note that our current model omits possible threats towards the validation process itself. This choice was motivated by our desire to demonstrate the feasibility of our security module, before further analysing and proposing advanced security protocols to prop up this vulnerability.

3.3 Threat taxonomy

Threat detection in our system is reduced to a binary operation, since all miners possess the knowledge of the expected route. Explicitly, if a routing node transmits a valid data packet towards the correct next hop for its destination, then it has performed a *Good* action. Any other action is considered *Bad* and, therefore, identified as a malicious activity. As such, our system is capable of detecting multiple types of active threats, simply by their actions during forwarding. Table 1 presents a brief taxonomy of threats which can be fully, or partially detected. It is important to note that some threats also possess passive variants. Contrary to their active cousin, these threats hide in the background and do not impact day-to-day operations and are generally considered to be reconnaissance related, such as packet sniffing or eavesdropping [26]. Since these are impossible to detect in our context, only active threats are considered.

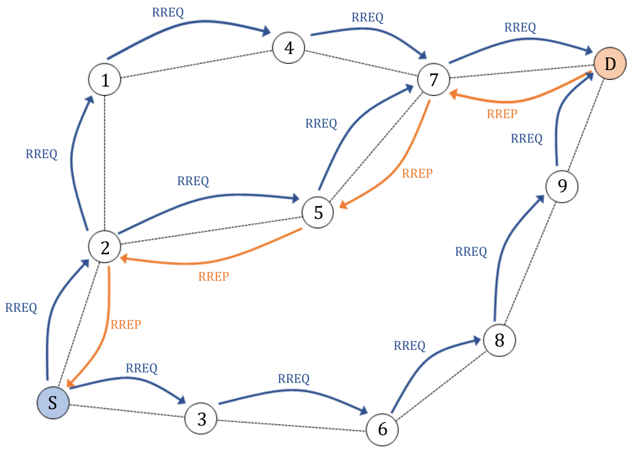


Fig. 3 – AODV discovery process

3.3.1 *Routing threats*

Possibly the most important action in a multi-hop network is the act of routing itself. As a result, it is important to reduce and eliminate any threat which seeks to impact network performance. By not transmitting towards the expected next hop, a malicious node can either transmit to the wrong next hop, or not transmit it at all. For example in Fig.3, node 7 can use Packet Redirect (*RTE07*) to deviate a packet from node 5 packet to node 4 instead of the destination. In the same idea, by destroying all packets with a black-hole attack (*RTE03*) or only some with a grey-hole-type attack (*RTE04*, *RTE01* & *RTE06*), data will never reach the destination. In either case, any deviation from the next expected hop will result in immediate detection by the miners. This also functions with other attacks, such as a sinkhole (*RTE02*) or wormhole (*RTE05*) attack, which can use another medium to reroute data, such as nodes 7 and 3 being connected using a cellular connection, thus elongating the route taken. In any case, since no corresponding transmission is detected by the miners, this activity is considered malicious. It is important to note, however, that some of these attacks can impact multiple aspects of the network. For example, a sinkhole attack manipulates routing tables to force traffic to transit through it, allowing it free access to the data. Although our system is capable of detecting deviations in expected routing, it is not currently specialised in detecting manipulations of AODV route discovery itself.

Table 1 – System active threat taxonomy

Threat Type	Threat ID	Threat	Description
Routing	RTE01	On-Off Attack	Random activation, dropping all or selectively dropping packets then randomly deactivate, causing periods of no attack where all packets are transmitted
	RTE02	Sinkhole	Trick other nodes to route traffic to a central point, allowing modification, dropping or forwarding at will to original destination or external device
	RTE03	Black-hole	All messages passing through a black-hole device are dropped, no exceptions
	RTE04	Grey-hole	Some messages passing through a grey-hole device are dropped, either randomly or by specific criteria
	RTE05	Wormhole	All messages passing through a wormhole device are captured and forwarded to another location inside/outside the network
	RTE06	Selective Forwarding	Similar to grey-holes, packets are forwarded or dropped based on specific criteria, or simply at random
	RTE07	Packet Redirect	Redirect passing traffic to wrong destination, or wrong next hop
Data	DTA01	Message Modification	Changing the content of passing messages, either at random or corresponding to specific criteria, changing the end result of the transmitted data
	DTA02	Replay	Capture a passing packet and replay it with or without modification at a later date
Node	NDE01	Byzantine	Multiple nodes are compromised and behave in an arbitrary manner causing network disruption
	NDE02	Node Capture	A node is compromised, granting ability to impact and control the network
	NDE03	Malicious Node	A node is compromised, transmitting false information to the network
	NDE04	False Node	A new node is added to the network, potentially replacing existing node, injecting false data, as well as disrupting routing or spreading malicious code to other nodes, taking over them or destroying them from the inside

3.3.2 Data threats

When sharing data, especially using the wireless medium, data integrity and privacy become an issue. Our taxonomy presents two data-based threats which can be detected. The first concerns Message Modification (*DTA01*) which directly impacts data integrity by modifying the packet's payload or even header. The second concerns the retransmission of previously sent messages, known as Replay (*DTA02*). To counter these threats, miners keep records of passing messages, allowing them to detect sudden changes to data integrity and resurfacing of previously encountered packets. Furthermore, since miners can only function when a route is present, if a packet is retransmitted after the route has expired and no other is active, it is immediately discarded and considered malicious.

3.3.3 Node-based threats

When nodes are left to their own devices without regular maintenance or surveillance, tampering becomes a threat. In many cases, gaining access to existing devices, or injecting a new node (*NDE04*) into a network provides surveillance capabilities to the malicious party. Although these threats are not detectable in our context, four active node-based threats are, however, they are only detectable in certain conditions. For example, if node 7 in Fig.3 aims to impact routing efficiency, then all deviations will be detected by the miners, which is the case of Byzantine attacks (*NDE01*). Captured, malicious or even new nodes (*NDE02*, *NDE03* & *NDE04*) can also be detected when acting upon the routing process or through modifying messages. However, if their goal is to legitimately inject invalid data into the network, then these threats are not detected.

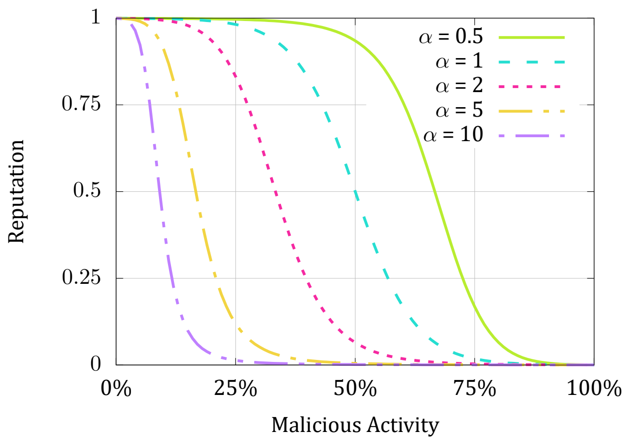


Fig. 4 – Reputation evolution

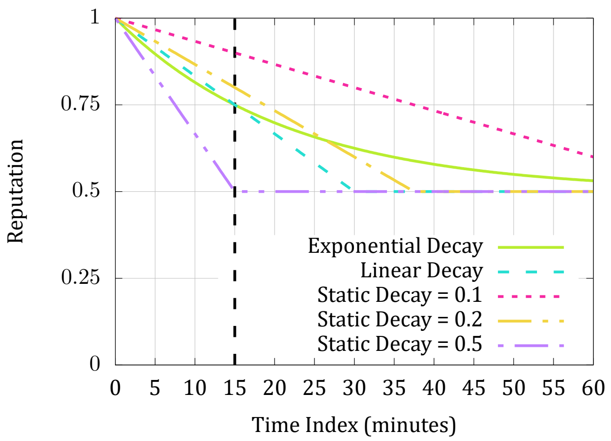


Fig. 5 – Reputation decay

4. CONSENSUS-BASED ROUTING

In this section, we present a consensus-based routing module using reputation metrics, implemented on top of the AODV protocol called *AODV-Miner*.

4.1 Behavioural analysis

To be able to accurately identify the activities of a routing node, their behaviour must be analysed. As explained previously, the miners possess the knowledge of the expected neighbouring hops for a specific route. By extracting and analysing the overheard transmissions, the miners are capable of detecting different threats. If a threat is detected, the transmission is labelled as malicious, thus impacting the reputation of the transmitting node.

4.1.1 Node reputation

The reputation of a node represents their trustworthiness in the network. As a result, it is calculated for the list of *good* and *bad* actions. These binary actions, are determined from the behavioural analysis, differentiating expected and non-modified transmissions as *good* and anything else as *bad*. As a result, the more actions there are in either category, the more the reputation will tend towards the corresponding value. In short, the greater the amount of *good* actions, the higher the reputation, and vice versa.

$$S_{good_n} = \sum_{i=1}^{W_n} good\, actions_{n_i} \qquad (1)$$

$$S_{bad_n} = \sum_{i=1}^{W_n} bad\, actions_{n_i} \qquad (2)$$

We define S_{good_n} and S_{bad_n} as the sum of *good* and *bad* actions respectively for node n, as computed in *(1)* and *(2)*. We also define W_n as the size of the action window time frame, corresponding to the number of previous actions taken into account during the calculation. By increasing or decreasing this value, we can influence the precision of the calculation. This allows the miner to take into account only the actions of the last exchange, or all actions during the last W_n exchanges. With this, we can open up the nodes history, allowing the network to have a longer or shorter memory when it comes to nodes' actions.
Armed with the quantity of *good* and *bad* actions during the time frame, we can calculate the nodes' reputation. Reputation $R_n \in [0, 1]$, is expressed as a sigmoid function, where the exponent $\delta_n \in [-1. 1]$ represents the weighted value of the relation between S_{good_n} and S_{bad_n}, calculated in *(1)* and *(2)*.

$$R_n = \frac{1}{1 + e^{-\delta_n}} \qquad (3)$$

$$\delta_n = \beta \times \frac{S_{good_n} - \alpha \times S_{bad_n}}{S_{good_n} + \alpha \times S_{bad_n}} \qquad (4)$$

We define two variables for the calculation of δ_n, the first of which is $\beta = 8$ which corresponds to the sensitivity factor influencing the sigmoid function, as presented in [25]. The second, α, is the weight of malicious actions upon the reputation. By changing this value, we can increase or decrease the impact of *bad* actions in relation to *good* actions. As a result, it is possible to increase or decrease the consequences of misbehaving nodes, making the network more or less tolerant. Fig.4 presents the evolution of a node's reputation based upon the value of α. As we can see, the higher the value, the higher the impact on the overall reputation and the more unforgiving the network becomes. This illustrates the impact of a node becoming malicious, where the more malicious actions are performed, the more the reputation will decrease. Furthermore, thanks to α, we can specify the impact of these actions, allowing the reputation to respond quickly to variations and changes in the node's behaviour.

4.1.2 Reputation decay

As presented in Section 3.3, certain threats can pertain to malicious access or corruption of legitimate nodes. Once their activities have been detected, a bad reputation is inevitable, resulting in the node no longer being used during routing. However, once a node has been isolated from the network, the attacker no longer has any use for it.

In many cases, the malicious party will move to a better position to continue their attack, leaving the compromised node alone. Since a node's reputation only evolves when they participate in routing activities, there is no way to reintegrate this node back into the network. To counter this issue and permit reintegration, we propose a new metric called *Reputation Decay*. Over time when the node does not participate in routing activities, their reputation will slowly decay towards the neutral value of 0.5. This will increase the chances of a node being used once more for routing, allowing it to clear its name. However, this decay does not change the number of *good* and *bad* actions performed by the node, but serving simply as a means for granting it a second chance. It also allows nodes which possess a very good reputation and have not been used for a while, to decrease back towards the neutral 0.5 as well.

We define Rd_{n_t} as the reputation decay of node n at time t, λ as the decay factor, $t_{\frac{1}{2}R}$ as the half-life of the reputation and R_{n_t} as the resulting decayed reputation of node n at time t.

$$Rd_{n_t} = (t - t_{R_n}) \times (\frac{\lambda}{t_{\frac{1}{2}R}}) \tag{5}$$

$$R_{n_t} = R_n - Rd_{n_t} \tag{6}$$

By varying the value or the function of λ, we can influence the rate of decay, allowing the convergence towards 0.5 to occur sooner or later. Fig.5 shows the evolution of the decay rate from a base value of 1 towards the neutral 0.5, with a half-life of $t_{\frac{1}{2}R} = 15\,min$ with various decay methods. For the rest of our analysis, we kept a half-life of 15 minutes and decided on a linear decay function with a decay value of $\lambda = 0.25$. As a result, a node's reputation will return to neutral from either an extreme of 1 or 0, after $2 \times t_{\frac{1}{2}R}$, corresponding here to 30 minutes.

4.2 Protocol integration

With the ability to calculate the reputation of a node based upon its actions, it is necessary for it to be integrated into the AODV routing protocol. Being a reactive routing protocol, discovery is performed only when needed, meaning it can take advantage of the existing reputations. However, for the reputation to influence the choice of route, modifications to the existing AODV packet structure is necessary. Furthermore, with new additions to the discovery process, we can provide the necessary information for the miners to accurately and reliably perform their activities.

4.2.1 Link cost

As explained previously, AODV determines the best route based on the number of hops thanks to the RREQ *hop-count* field, thus discarding longer routes and keeping only the most direct possible. However, in our context it is necessary to exchange the length of the route and instead use its reliability factor. As performed in [25], we replace the *hop-count* field with a metric called *link-cost*. This allows the nodes to calculate the "cost" of using a certain neighbour, based upon that neighbour's current reputation. With this metric, we can differentiate and separate *good* nodes from *bad* ones by simply increasing the *link-cost* the lower the node's reputation. Upon receiving an RREQ or RREP packet, the node calculates the sender's reputation, along with its potential decay. It then determines the *link-cost* corresponding to the final reputation, increasing the value of the *link-cost* field accordingly. By updating this field, no modifications are brought to the overall functionality of AODV, where the route with the lowest *hop-count* is selected, only here the value corresponds to the most reliable route. This allows the route to contain as fewer malicious nodes as possible, all the while facing a trade-off of longer routes for increased route integrity.

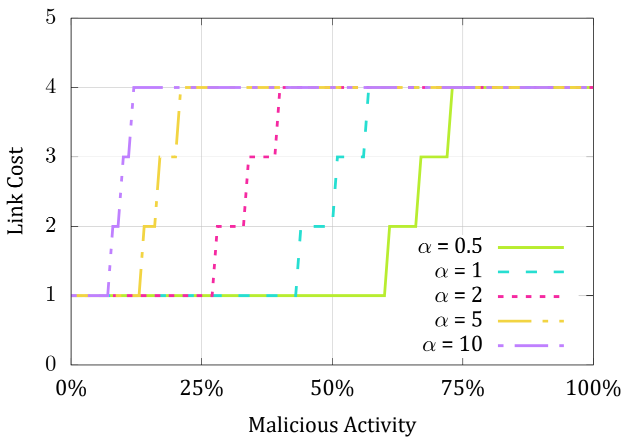


Fig. 6 – Link cost evolution

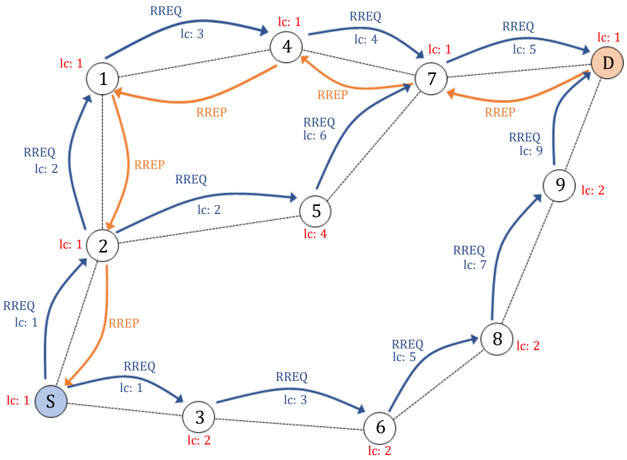


Fig. 7 – *AODV-Miner* discovery process

$$C_n = \lfloor (1 - R_{n_t}) \times (C_{max} - (C_{min} - 1)) + C_{min} \rfloor \tag{7}$$

We define C_n as the *link-cost* between the current node and the node n, with R_{n_t} corresponding to the reputation of said node at time t. As R_{n_t} is normalised between

0 and 1, it is necessary to expand and adapt the resulting *link-cost*. We, therefore, define C_{min} and C_{max} as the minimum and maximum values possible for this cost. By setting $C_{min} = 1$, we assure that even with an excellent reputation, the *link-cost* field will always be incremented by one, thus removing the risk of infinite cost calculation loops. Finally, the resulting value is then decreased to the nearest natural number, less than or equal to the calculated value. Since AODV's *hop-count* field is only one byte in width, the value of the *link-cost* must be adjusted accordingly. With an overall maximum potential network cost of 255, we can calculate the maximum possible *link-cost* C_{max} based upon the number of potential nodes in the network.

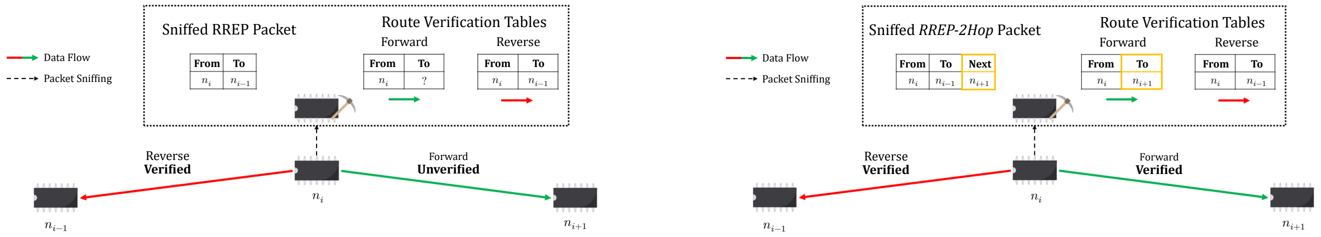
$$C_{max} = \frac{255}{L_{max}} - 1 + C_{min} \tag{8}$$

With L_{max} corresponding to the maximum possible route length (i.e., number of nodes traversed), we can adjust the precision of the *link-cost* metric. For example, with $L_{max} = 32$, we could accommodate a maximum value of 8, whereas $L_{max} = 64$ would only allow for four individual values. By proposing an adaptable scaling function, we can increase or decrease the precision of the *link- cost* metric in relation to the number of nodes. Also, by tying this value into AODV itself with the `NET_DIAMETER` parameter, we can provide a seamless integration between the two. However, although AODV allows each node to customise the value of `NET_DIAMETER` accordingly, our method needs the value of L_{max} to remain constant throughout the network, or risk a route being dropped for cost overflow. For the rest of our analysis, we decided on $L_{max} = 64$, which corresponds to the maximum TTL value widely used in networking, resulting in our routes containing at most 64 nodes. Fig.6 shows the calculated *link-cost* values for the different reputational values previously presented in Fig.4. Fig.7 illustrates the discovery process of *AODV-Miner*. By comparing this with Fig.3, we can see the differences where node 5 exhibits malicious tendencies. Since AODV selects the shortest route possible in terms of hops, the RREPs will always transit via node 5 for a maximum of 4 hops compared to 5 hops via the other routes, putting the data at the mercy of our bad guy. By adding the *link-cost* into the equation, we can influence the route selection process, thus avoiding the malicious entity. This is visible in Fig.7 where each node possesses a *link-cost* (lc) . Since node 5 is malicious, we assume it has received a low reputation, resulting in a high *link-cost* of 4. This high value causes an increase of the total route cost, bringing it up to 6 from the source node to node 7. In this case, the top route is the winner, with a total cost of 5 from source to destination, making it the most efficient and trustworthy route. Thanks to the quick reactions of the reputation metric, the *link-cost* can also adapt in a timely manner, immediately influencing the selection of the next route. Indeed, since the validation process takes place after a route has expired, the updated reputations only enter into play the next time the node is needed. This means that as long as the route remains active, the malicious node can impact the routing activities, however, the more actions it performed the more severe the consequences. It is also important to note that by artificially lengthening the route used depending on each node's reputation, we do not explicitly isolate nodes from routing. Our method simply encourages the protocol to seek another route towards the destination avoiding the malicious entities as much as possible. However, in some cases, no alternative routes exist, and the malicious node is utilised, thus impacting the network security. Further study into these two points can help reinforce the network security, and is also one of our current directions.

4.2.2 RREP 2-Hop

So that the miners can achieve their goals of route validation, they must know to whom the packets must be sent. By overhearing passing RREPs, miners can construct their view of the expected route towards the destination, but also back towards the source, adding the hops to the corresponding *RVTs*. Unfortunately, although overhearing RREP packets allows the miners to construct parts of the route, they are missing some elements of the big picture. Indeed, since RREPs only serve to inform node $n - 1$ to transmit towards n, the miners are only aware of the expected exchange between these nodes. This information is insufficient, as in many cases node n is not the destination and will, therefore, need to transmit its data onwards. However, it its current state, the miners are incapable of prediction to whom this packet will go, meaning they are incapable of validating the behaviour. This problem is illustrated in Fig.8a, where we can see that our miner can only overhear the communications coming from node n_i. As a result, the RREP packet only informs on the reverse route back to the source through n_{i-1}, and not the forward route towards n_{i+1}.

To remedy this, we propose an amelioration to the RREP packet format, allowing us to include the information for the next hop. This new packet format, called *RREP-2Hop* is presented in Fig.9. We can see the addition of the new *2Hop* section, containing the IP and MAC addresses of the node's next hop. By providing the MAC addresses of the next hop, the miners can complete their *RVTs* and achieve their goals. By also taking advantage and incorporating the corresponding IP address, each node can also construct *2Hop Routes* in their routing tables, if they so desire. As we can see in Fig.8b, this new addition allows the miner to determine the forward route from n_i towards $n_i + 1$, allowing full validation to take place. So as to allow our solution to be adapted to existing AODV routing, we also incorporated a *Miner Flag* into the packet header. This allows the system to differentiate and identify the RREP packets, allowing the choice to function with or without our addition.



(a) Validation with RREP

(b) Validation with RREP-2Hop

Fig. 8 – Illustration of the need for RREP-2Hop

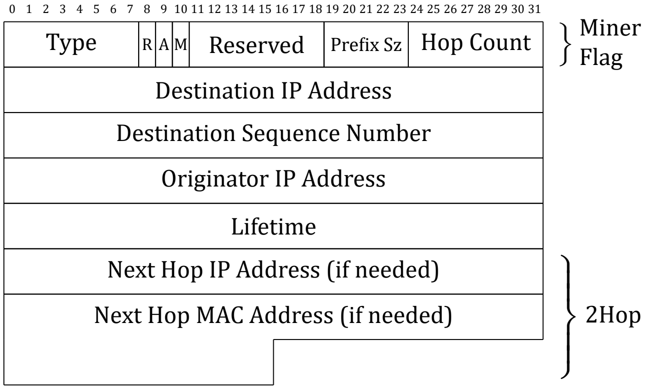


Fig. 9 – RREP-2Hop packet structure

4.3 Behavioural validation

To be able to determine the reputation and influence the route selection, there are a few steps which need to be performed. In this section, we present the miners themselves, taking a look at how they perform their different roles. However, before they can perform their activities, the miners themselves must be selected and differentiated from the routes' routing nodes.

4.3.1 Miner selection

As stated previously, we provide the ability for all network nodes to determine their own role per route. However, nodes cannot take on both roles of miner and router at the same time for the same route as this could result in a conflict of interest. This is because a routing node cannot objectively analyse their own behaviour, or that of the node which has transmitted the information to them. Furthermore, by separating the roles between multiple nodes, we reduce the probability that the potential malicious node could also impact the validation phase, subsequently corrupting the reputation table. The selection process is performed during the AODV route phase, allowing all miners to be identified and possess all routing information needed to perform route validation once the route becomes active. As presented previously with the definition of *RREP-2Hop*, miners use RREP packets to gather the necessary hop information. Upon receiving an RREP packet, the node first analyses the destination address. If the RREP is destined for them, then they identify themselves as part of the route, processing the packet information as normal and constructing the different routes in its routing table, using the *2Hop* address if desired. On the other hand, if the RREP is not destined for them, then the node enters an internal validation phase. They first check if they are not already a router for the route, in which case the RREP is immediately dropped without further analysis. If not, then the destination link-layer address is extracted from the packet header and the *2Hop MAC* address from the *RREP-2Hop* payload. Both addresses are then used to construct the reverse and forward *Routing Validation Table* entries for the node which transmitted the RREP.

4.3.2 Routing analysis and validation

Once the route discovery has completed, the route can begin transmitting data. The selected miners then begin to "*mine their route*" by observing and analysing all data traffic originating from neighbour nodes. To accurately analyse the data exchange, the miners utilise their forward and reverse *RVTs*. Each table contains the ordered list of expected hops in transmission range of the miner. These tables, visible in Fig.8b, allow the miner to verify that all packets follow the same hop ordering. This allows us to detect any redirecting attacks where the destination does not conform to the table entry, or packet destruction where the hop list is not traversed completely. However, it is important to note that as presented previously, we are only able to validate data originating from the route's source towards the route's destination and not intermediate exchanges taking advantage of the routing table entries.

For each packet received, the miners process the data to determine its authenticity, as presented in Algorithm 1. During the analysis, the miners verify the packet's destination as well as its integrity, allowing it to identify if the transmitting node has malicious tendencies. The verification phase stays active as long as the route itself is in use. Upon expiration, the miners first check their passing packet buffer, identifying packets currently in transit. If the buffer contains data, then the last associated node is considered to have not transmitted the data onwards and, therefore, increasing the number of *bad* actions. Once all actions have been totted up, the miners all drop their *RVTs* for the route and enter their final phase of block confirmation.

Algorithm 1 Miner route validation run at miner m upon reception of pkt(llsrc,lldst,src,dst)

```
if New packet detected then
    Create new $buf_{pkt}$ entry with $hash_{pkt}$
    set $buf_{pkt}$ as valid
else Previous malicious activity detected ; Exit ;
end if
$RTE =$ Get route entry for $[src \rightarrow dst]$
$RVT =$ get validation tables from $RTE$ for $llsrc$
if $RTE$ & $RVT$ both empty then
    ▷ No route validation table, Malicious behaviour
    Increment $bad_{llsrc}$; Set $buf_{pkt}$ as invalid
else
    $nextHop_{pkt} =$ get the next hop from $RVT$
    if $nextHop_{pkt} \neq lldst$ then    ▷ $lldst$ is not the next expected hop - Malicious behaviour
        Increment $bad_{llsrc}$; Set $buf_{pkt}$ as invalid
    else    ▷ Valid behaviour
        Increment $good_{llsrc}$
    end if
end if
```

4.3.3 Block confirmation and dissemination

To allow consensus-based confirmation, the miners must first create their own block containing the number of*good* and *bad* actions for each and all routing nodes which it has mined. The block is then broadcast up to a maximum distance of 2 hops, allowing it to reach only nodes in proximity which are potentially miners for the route. Upon receipt of such a block, the miner proceeds with two calculations. Firstly, they analyse the number of *good* and *bad* actions contained in the block, calculating the number differences with their own block. If this value is too high, the block is considered to be invalid and the miner transmits their own block as a response. However, if no differences are detected, the miner then performs an efficiency evaluation to determine if the block is more efficient than its own. This is achieved by calculating the percentage of nodes in common in the received block, P_B versus the miner's own block P_M, with B corresponding to the list of nodes in the received block and M those in the mined block.

$$P_B = \frac{|M \cup B|}{|B|} \qquad (9) \qquad\qquad P_M = \frac{|M \cup B|}{|M|} \qquad (10)$$

The miner only transmits its own block in this case if it is deemed more efficient, in other words if $P_B < P_M$ where P_M is considered to possess more nodes overall and a higher percentage of shared nodes. Since miners can corrupt the results of this exchange, the process relies on a consensus where responses from miners overrule previously transmitted blocks. To stop validation loops, miners can only transmit their own block once, allowing the last block to correspond to the majority. If the received block is considered more efficient, the miner then identifies all common nodes as "overridden", meaning they have been confirmed by another more efficient block. This allows miners to detect if they possess a node which has not been validated by other miners, allowing them to retransmit their own block containing only the missing nodes for validation. As a result, the last blocks to be received and not overruled are considered both valid and more efficient since they possess the largest quantities of nodes possible, without overlapping with other blocks. The only task left is purely blockchain related, where the miners hash the contents of their blocks, inserting the hash of the last received blockchain block, then inserting it into the blockchain by broadcasting it throughout the network. This allows all network nodes to extract the list of *good* and *bad* actions for each node, knowing that the block is valid.

5. IMPLEMENTATION AND SIMULATION

As stated in the previous section, each node contains two *RVTs*, storing the ordered list of forward hops, towards the destination, and reverse hops, back towards the source. The nodes also possess a *Packet Buffer*, containing a list of packet hashes as calculated by miners along with their next expected hop. This allows the miners to detect modifications to the packets, as well as serving as a reminder as to which hop is next expected for this packet. The nodes also own a *Node Reputation Table*, which contains the list of *good* and *bad* actions for each node as extracted from the blockchain. These actions are input into Eq. *(1)* - *(4)* to calculate the node's current reputation. The number of actions stored in this table is influenced by the size of the reputation window W_n as shown in Eq. *(1)* and *(2)*.

Since our implementation revolves around a lightweight version of the blockchain, its functionalities are emulated. This means that the chain itself is not stored on the nodes, but only disseminated and analysed by the network. By not storing the received blocks, we save on node memory, which we can put to other uses such as reputation values or the behavioural validation itself. Upon receipt of a new block from the blockchain, each node calculates the block's hash, allowing them to verify the integrity of each subsequent block. When a route discovery is triggered, each node accesses the *Node Reputation Table* entry for the RREQ or *RREP-2Hop* sender and calculates the corresponding reputation. The node then determines the time since the last use of the corresponding node and applies the reputation decay function *(5)* as needed. The resulting reputation is then fed to the *link-cost* function *(7)*, providing the corresponding cost for using said node. By comparing the *link-cost* field of received RREQs, we can make sure to propagate only the lowest values onwards, thus eliminating potentially malicious routes as the discovery process advances. However, with the addition of this metric, it is possible that on occasion the calculated *link-cost* is lower than the previous. This is due to a field overflow after a significant number of hops and as a result the corresponding RREQ can be discarded as it can be considered too malicious. By only propagating RREQs with low *link-cost* values, we can assure that the destina-

Table 2 – Simulation parameters

Parameter	**Setting**
Area	*Varying*
Number of nodes (N)	*Varying*
Malicious Activity	*Varying*
Malicious Weight (α)	*Varying*
Distribution	Random uniform
Transmission Range	50m
Max Length (L_{max})	64
Window Size (W_n)	5
Reputation Decay	Linear
Initial Reputation	0.5
Number of Simulations	100
Simulation Duration	15 min.
Messages per Transmission	5
Transmission Interval	1 min.

tion only receives the most reliable routes possible. Furthermore, contrary to the approach in [25], here the destination node does not wait for the most reliable route before responding towards the source, thus providing all possible routes for the source itself to choose the best possible option. In our implementation, upon receipt of an RREQ, the destination waits for a small period of time before transmitting the RREP back towards the source. If any subsequent better RREQs are received, the destination waits once more before transmitting the corresponding RREP. Once the RREPs return to the source node, the node also waits for a slightly longer time period for potential other RREPs to arrive, before transmitting along the most efficient route. Any subsequent RREPs update the route as transmissions are occurring, without impacting network operations.

5.1 Simulation settings

For our analysis, *AODV-Miner* was implemented using the Contiki-NG [27] operating system and subsequently simulated using their Cooja simulator. Table 2 presents the general parameters used throughout our simulations. The simulated Cooja nodes possess a wireless interface using the IPv6 net-stack running a 6LoWPAN network layer and a non-beacon-enabled always-on CSMA radio. Although CSMA allows us to reduce the probability of collisions, it does not remove it entirely, especially concerning nodes which are list listening and overhearing transmissions. Since this problem can impact AODV and data transmissions as much as our Miners, we rely on the underlying network protocols, as well as our multi-miner validation approach to reduce the possible consequences. Similarly, the always-on radio permits the nodes to remain in the necessary active state, needed for both AODV and the validation miners. Their on-board systems are initialised using individually-generated seeds, allowing each node to possess a different random generator, all the while providing precise calibration of parameters. The different malicious nodes are distributed throughout the network using a random distribution function, only impacting data traffic whilst leaving AODV-related communications unscathed for the analysis of the routing protocol. For ease of analysis, we simulate the network against two types of threats: black-holes and grey-holes. As previously explored in Section 3.3, although we are capable of detecting many threats, our detection system revolves around the same methods: deviation from expected activities. As a result, black-holes allow us to simulate complete data destruction, whereas grey-holes allow us to vary the probability of destruction, allowing more or less packets to transition through the network. This means that even with only two attacks, we can hypothesise that the results would be similar with the other attacks, since their consequences and subsequent detection would be the same.

During our analysis, we used two network topologies, pitching *AODV-Miner* against its older brother AODV. The first contains 100 nodes in an area of 300m$\times$300m whereas the second contains only 30 nodes, in a smaller area of 150m$\times$150m. This allows us to test our system in two different situations, where the possible route length significantly increases, as well as the number of potential malicious nodes. In both situations, we transmit five random data packages every minute, allowing the network time to perform route discovery, packet routing and blockchain dissemination.

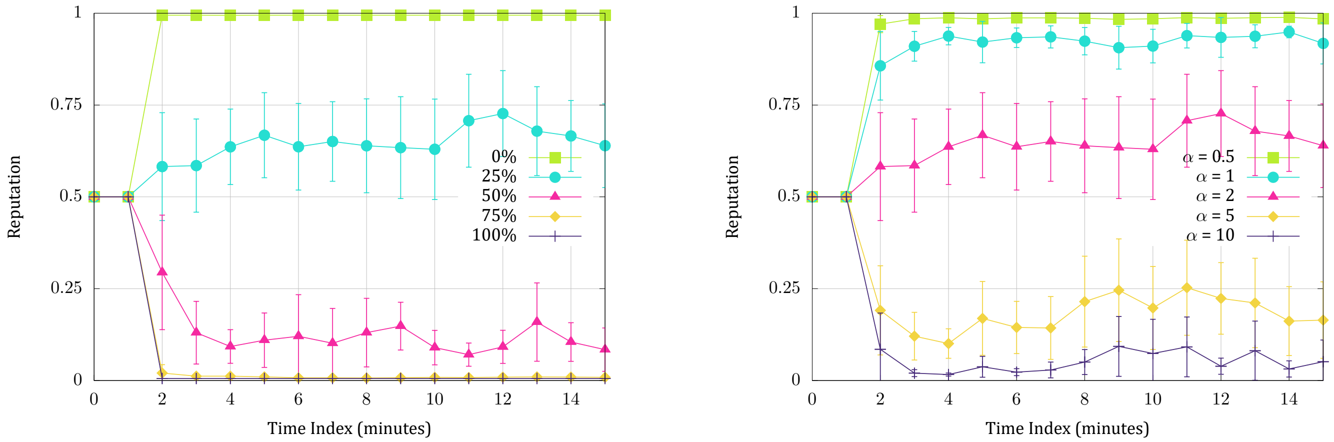
6. RESULTS

Our simulations allowed us to evaluate and analyse the overall functionalities and efficiency of our approach. By varying the topological layout, we could verify that our methodology would be able to handle different-sized networks. We start our analysis by evaluating the functionality of the reputation metric, before taking a gander at the routes themselves. Finally, we analyse how our method holds up against varying degrees of malicious activities, simulating both black-hole and grey-hole attacks.

6.1 Reputation analysis

Fig.10a shows the evolution of a node's reputation over time with varying degrees of malicious intentions. By using $\alpha = 2$, we double the weight of malicious activities in relation to $good$ actions. This can be observed with 25% malicious activities, where the resulting reputation resides around the neutral 0.5 mark. As a result, the greater the malicious activities, the lower the reputation, with 75% and 100% practically indistinguishable. Furthermore, we can also notice that the reputation is established immediately after the first route expires, round about the 1 minute mark. We can also see that, although the values fluctuate, they remain in the same overall area throughout the simulation.

By varying the value of α, presented in Fig.10b, we can observe its impact on the reputation. In this figure, we analyse the evolution of the reputation for 25% malicious activities. We can verify this by comparing the results of



(a) Reputation over time with varying degrees of malicious activities

(b) Impact of α with 25% malicious activity

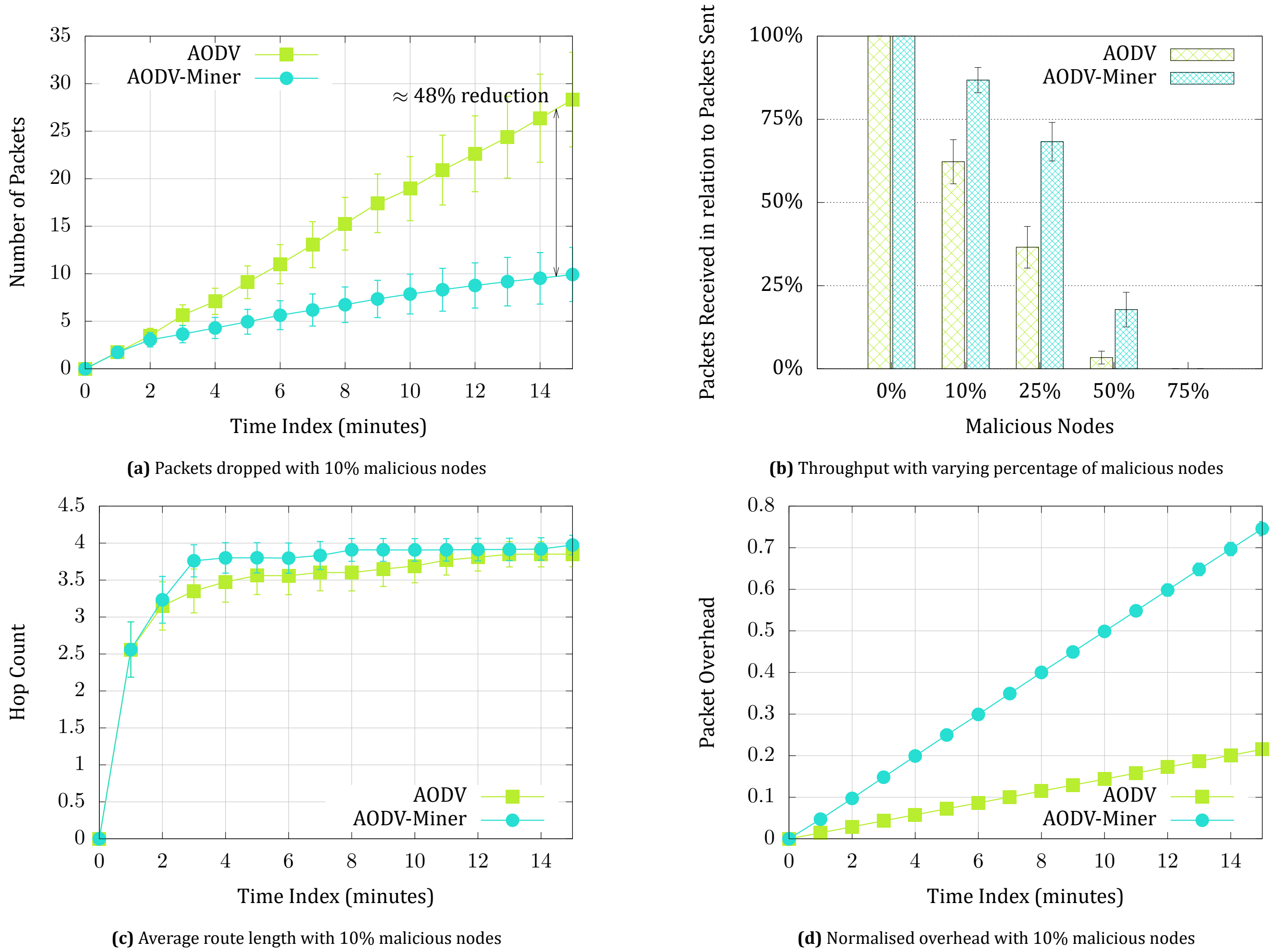
Fig. 10 – Evolution of node reputation

$\alpha = 2$ with the 25% malicious activities from Fig.10. Immediately, we can confirm our hypothesis of the impact of α as we can clearly observe that the greater the value, the lower the reputation. This is also true in the opposite direction, with the corresponding results for lower values of α finding themselves closer to the perfect reputation of 1. In essence, by acting on this variable we can actively influence the weight of all bad behaviour, instantly punishing a node for misbehaving, granting them forgiveness more swiftly.

6.2 Route analysis

By analysing the routing efficiency, we can determine if *AODV-Miner* can reach its goal of isolating as many malicious nodes as possible from the determined routes. Figures 11 and 12 compare these results against the standard AODV protocol in a network of 30 and 100 nodes respectively. Firstly, we analyse the number of packets dropped $(|P acketsSent| - |P acketsReceived|)$, visible in Fig.11a and Fig.12a. We can immediately see that there is a reduction in lost packets, with an overall increase in efficiency of 48% with 30 nodes, and 38% with 100 for a network with 10% malicious activities. Furthermore, these results are corroborated in Figures 11b and 12b, where we can see that *AODV-Miner* possesses a higher overall throughput than AODV for both typologies, whatever the percentage of malicious nodes. It is to be noted that not all drops can be prevented, since the reputation is computed on the fly, leaving time for malicious entities to cause mayhem. It is also possible that in some cases, traversing a node with a *link-cost* of 4, is still considered more efficient than five nodes with a cost of 1. However, there is a consequence to this increase in efficiency. Indeed, Fig.11c and Fig.12c show a trade-off, where we may indeed have better efficiency, but at the cost of longer routes. In our network with only 30 nodes this difference is minimal, however, by increasing the number of nodes we can see an increase in the number of hops. This is not the only cost of our implementation. Another is linked to the activities of the miners, since block validation and distribution increases the number of packets exchanged throughout the network. Our final analysis in Fig.11d and Fig.12d demonstrates this increase, with both 30 and 100 node typologies possessing a significantly higher overhead, ending up around the 80% mark. Although this may seem high, it is a necessary evil to ensure that a higher percentage of data reaches its destination unscathed.

Thanks to these results, we can confirm that our method allows us to isolate and avoid malicious nodes, increasing the probability of data reaching its destination. Fig.13 illustrates this process in networks of 30 and 100 nodes, both with 25% exhibiting black-hole characteristics, represented with thick outlines. By superimposing the computed reputation for all nodes, as well as the most used route by both AODV and *AODV-Miner*, we can visualise this increase in performance. In both networks, we can see that AODV attempts to take the shortest most direct route possible per its programming, which unfortunately results in encountering a malicious node. In contrast, *AODV-Miner* is capable of discovering a free trustworthy route between the source and destination, avoiding malicious entities. As we can see by the colour gradient, nodes have been attributed both high and low reputations, depending on their activities during routing. By analysing Fig.13a, we can see that a total of eight nodes have been attributed reputations higher than the neutral 0.5, whereas three others have received low reputations. As stated previously, it is a necessary evil to allow messages to be lost to allow for the malicious activities to be detected and the reputation computed. This means that in this scenario, three determined routes ended with all their data being lost before *AODV-Miner* was able to adapt. Of course, this effect is amplified the larger the network, and consequently the more malicious nodes are present. In contrast, Fig.13b presents a significant sixteen nodes possessing a high reputation and seven with low values, four more than the smaller network. We can also see a



(a) Packets dropped with 10% malicious nodes

(b) Throughput with varying percentage of malicious nodes

(c) Average route length with 10% malicious nodes

(d) Normalised overhead with 10% malicious nodes

Fig. 11 – Routing efficiency between *AODV-Miner* and AODV with a network of 30 nodes

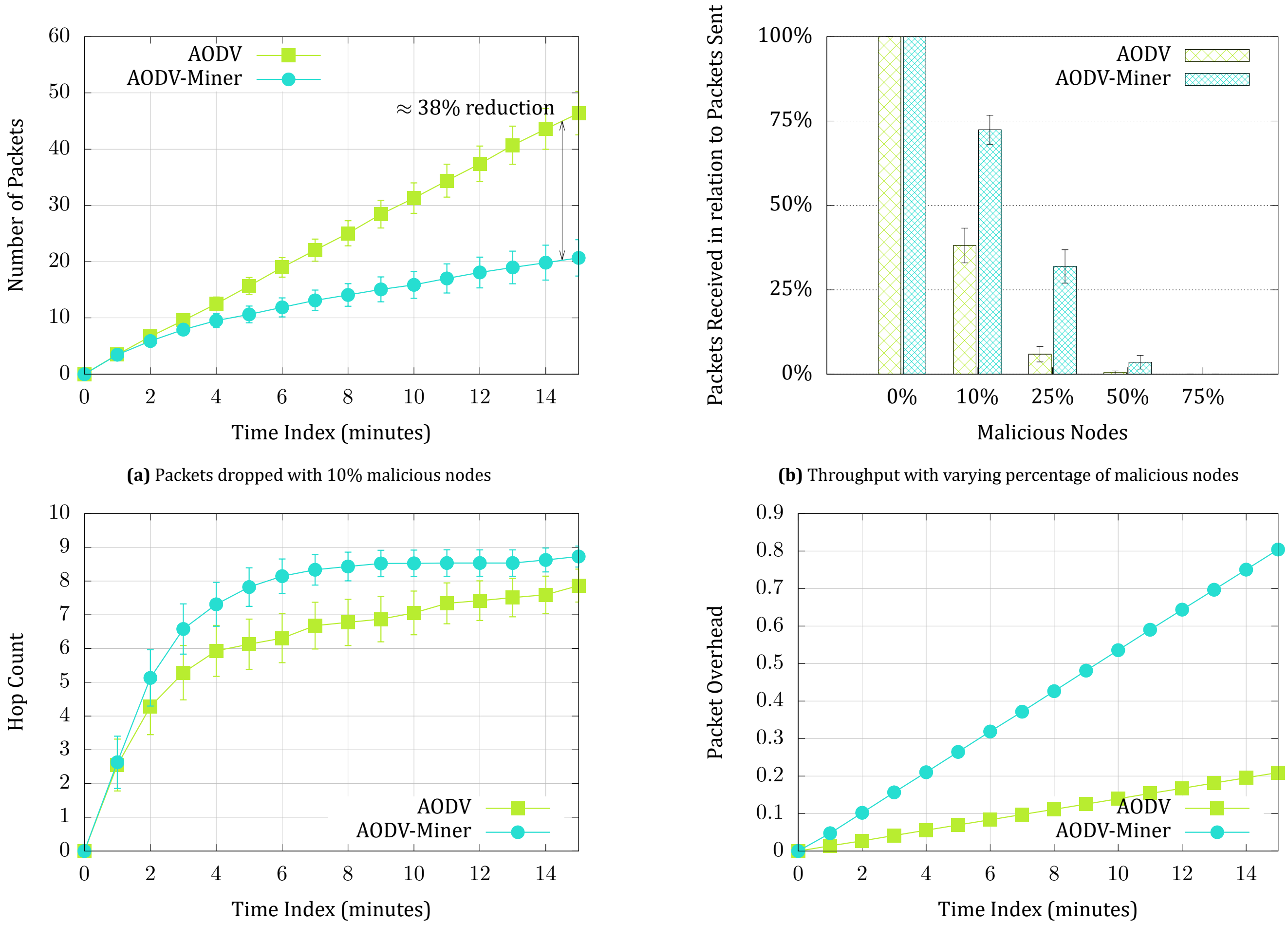
cluster of four malicious nodes in the centre of the network separating the source from the destination, all of which have been detected and subsequently avoided. One final note is that, as is the case with AODV, the route selected may on occasion change due to various reasons. We can see this with the fact that in both figures, there are nodes which have good reputations, and yet are not part of the most used route. This is possible where some RREQ messages are lost due to collisions, forcing the network to select an alternate route, or simply arriving too late to change the selected route.

6.3 Threat adaptation

The final aspect of our analysis concerns the ability of our system to adapt to different threat types. In this context, we pitch the *AODV-Miner* against varying degrees of packet drops in a grey-hole attack. Some grey-hole attacks use packet selection to decide which data to destroy and which to let pass, also called Selective Forwarding (*RTE06* in Table 1). In our case, we use internal probability functions to decide which packet to drop on each malicious node, each initialised with a different seed allowing different values of probability between them.

Fig.14 shows an analysis of these activities for a network of 30 nodes and Fig.15 for a network of 100 nodes.

If we turn our attention to the analysis of the 30 node topologies, Fig.14 shows the different throughput levels of AODV against *AODV-Miner* with varying numbers of malicious nodes, based on the grey-hole probability in use. We also extend this analysis by comparing the results with different values of α, thus showing its impact on the determination of the reputation and consequently the routing efficiency. We can see that in general, *AODV-Miner* performs well, keeping an overall throughput higher then the corresponding values of AODV. Naturally, the more nodes turn to the dark side, the harder it is for *AODV-Miner* to determine a free route, which we can see with the very slight increase in network efficiency. Fig.14a shows the results where $\alpha = 0.5$ corresponding to a very forgiving network where malicious activities have half the impact of *good* activities. This means that a node needs to perform twice the amount of *bad* activities than *good* to warrant a decrease in its reputation. This can be confirmed in the results with 10% and 25% malicious nodes possessing a malicious probability of 50%, where the throughput drops slightly since on average the nodes drop every other packet they receive. However, the moment the



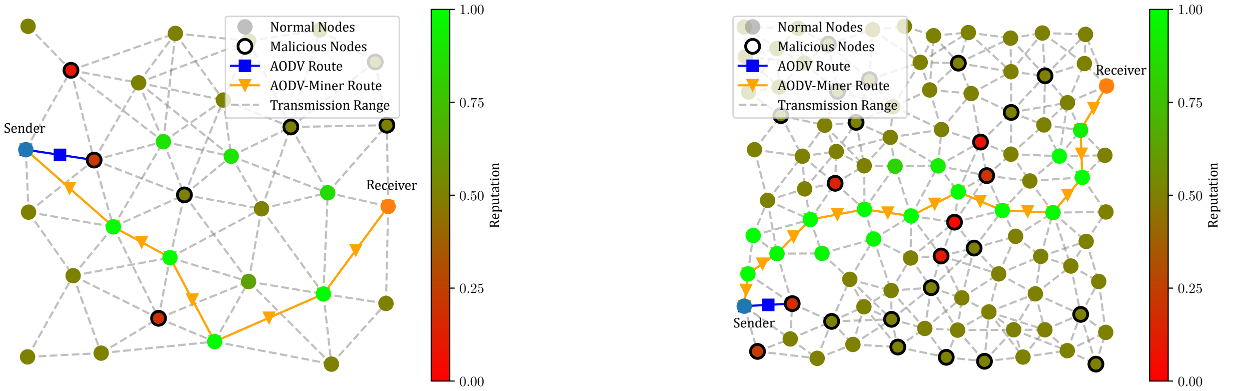
(a) Packets dropped with 10% malicious nodes

(b) Throughput with varying percentage of malicious nodes

(c) Average route length with 10% malicious nodes

(d) Normalised overhead with 10% malicious nodes

Fig. 12 – Routing efficiency between *AODV-Miner* and AODV with a network of 100 nodes



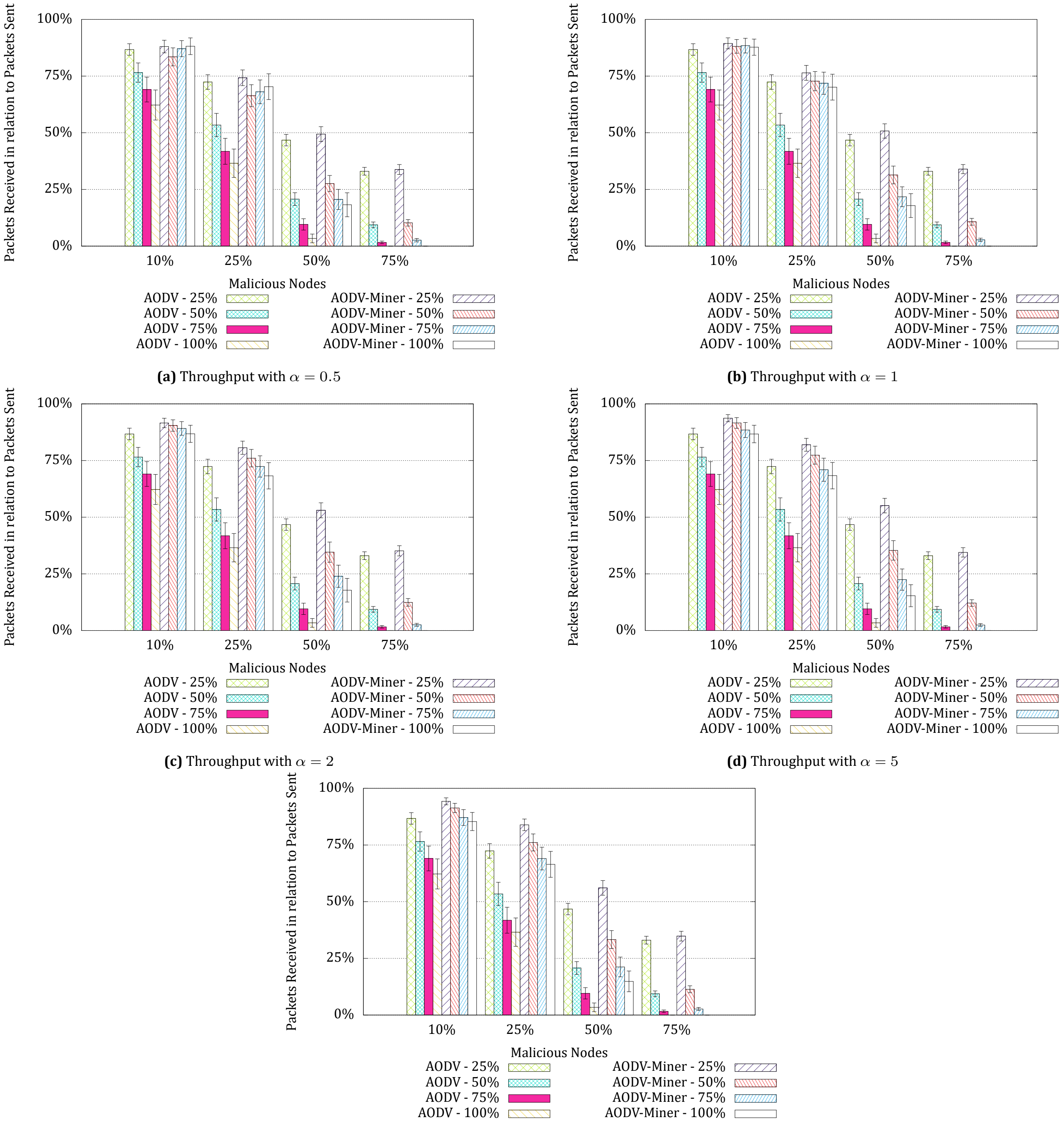
(a) 30 Nodes

(b) 100 Nodes

Fig. 13 – Visualisation of route reputation after 15 mins. with 25% malicious nodes

percentage of packets dropped is higher than a ratio of 1 : 1, the throughput rises once more, increasing even higher when all packets are being destroyed, reaching the same value as 25% malicious probability. In contrast, Fig.14b represents the case where *good* and *bad* activities possess the same weight, $\alpha = 1$.

Here we can see that, for 10% malicious nodes, the throughput decreases only slightly the higher the malicious probability, simply due to the need for packets to be dropped before the reputation can be computed. The rest of the results decrease in throughput



(a) Throughput with $\alpha = 0.5$

(b) Throughput with $\alpha = 1$

(c) Throughput with $\alpha = 2$

(d) Throughput with $\alpha = 5$

(e) Throughput with $\alpha = 10$

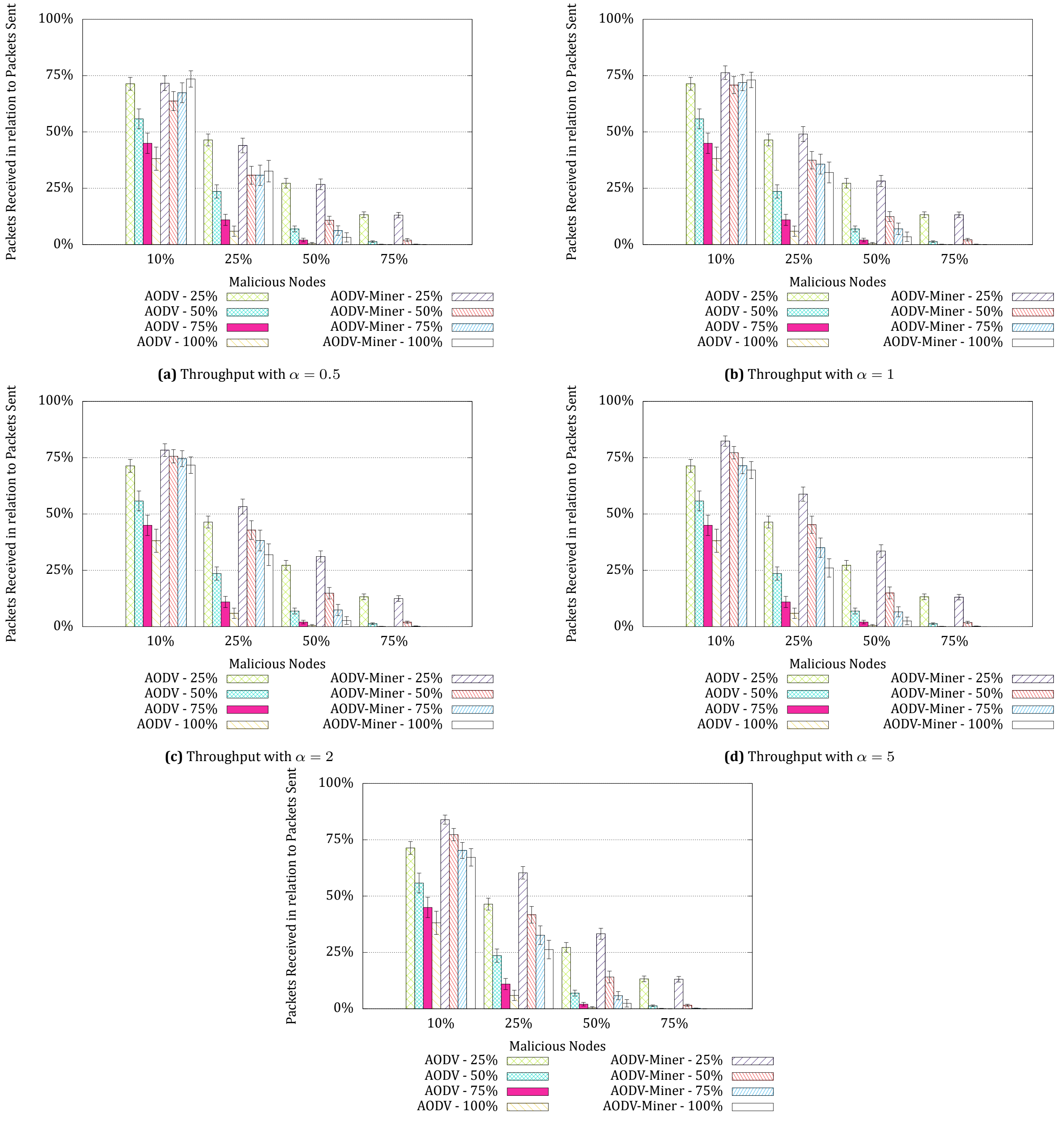
Fig. 14 – Throughput comparison between *AODV-Miner* and AODV with a network of 30 nodes subjected to grey-hole attacks

the higher the probability, all the while remaining slightly higher, or on par, with the results from Fig.14a. However, we can already identify a slight decrease in throughput when all packets are being dropped when compared to the previous figure. Fig.14c shows the first analysis where malicious activities possess a higher weight to *good*, with $\alpha = 2$. Comparing with $\alpha = 0.5$, here nodes need to perform twice the amount of *good* actions than *bad*, to stabilise their reputation once more. We can observe that, contrary to the previous analyses, there is a distinct decrease in reputation the higher the malicious probability, all the while remaining higher or equal to AODV. However, once more we can see that once more, the throughput for 100% of packets being dropped is lower than the previous values of α. On the other hand, due to the increase in malicious weight, the initial throughput with only 25% of nodes exhibiting malicious tendencies is higher than before. As a result, the higher the value of α, the more weight is accorded to *bad* actions and the faster *AODV-Miner* can react. That being said, there is a point where we reach peak efficiency, and the throughput cannot increase any higher and even starts to decrease slightly. This is the case of Fig.14d and Fig.14e with $\alpha = 5$ and $\alpha = 10$ respectively. We can see that the values remain extremely similar, with in some cases $\alpha = 10$ presenting slightly lower results than $\alpha = 5$, amplifying the previous observations for 100% malicious probability. However, as stated previously, when the vast majority of the network has become one with the enemy, there is only so much that can be done to try and combat the issue. This is the case with 75% of nodes exhibiting malicious habits, where the results for all five values of α are extremely close with very low throughput levels.

By analysing the results from networks of 100 nodes, presented in Fig.15, we can analyse and strengthen our hypotheses. First off, we can see that in general the larger network size has resulted in general decrease in throughput level, due to the presence of more malicious nodes, as illustrated in Fig.13b. By beginning our analysis once more with $\alpha = 0.5$ in Fig.15a, we can see the same pattern as previously, where the throughput drops between 25% and 50% malicious probability with 10% malicious nodes, only to rise once more, this time surpassing the throughput with 25% probability. This is also the case with 25% malicious nodes, although the increase is more subtle than the 30 node network in Fig.14a. However, here we can see that for 25% malicious probability, the corresponding throughput is lower than that of AODV for all percentages of malicious nodes. This reinforces our hypothesis that a low value of α makes the network more forgiving, meaning it takes longer to detect and isolate malicious nodes, resulting in them being used more often, dropping more packets. Furthermore, whereas AODV on occasion will change routes depending on which RREP returns first and the potential RREQ losses, *AODV-Miner* would continue to use the node, since it would receive a good reputation, as previously demonstrated in Fig.10b. Increasing the value of α consequently increases the overall throughput, although some parallels with the low value of α can still be made. This is the case for $\alpha = 1$ in Fig.15b, where a similar phenomena can be observed with 10% malicious nodes, all the while possessing a generally higher throughput. By looking at the values for 25% malicious probability, we can see that *AODV-Miner* is once again higher than AODV, reinforcing our previous hypothesis. Increasing the influence of bad actions, visible in figures 15c, 15d and 15e demonstrates the advantages but also disadvantages of higher values. If we turn our attention to the results for 25% malicious probability, we can see the corresponding throughput increases the higher the value of α, also visible in the other two figures. However, the higher the malicious probability, the more the associated throughput seems to struggle, decreasing slightly the more α rises, similarly to the network of 30 nodes. This can be explained by the fact that malicious nodes are detected quicker, the higher the vale of α, explaining the increase in throughput for 25% malicious probability. This advantage allows *AODV-Miner* to determine new routes constantly once a malicious node has been detected. Furthermore, with a malicious probability of 25%, on average 1 packet in 4 is dropped, meaning it is possible that for every four packets transmitted along the same route, up to *four* malicious nodes can be detected, increasing the efficiency of *AODV-Miner*. As a consequence, the higher the malicious probability, the longer it takes to detect and circumnavigate malicious nodes. In the previous example, a malicious probability of 50% would produce a drop rate of 1 in 2, meaning that for four packets we could potentially detect only *three*, further decreasing to *two* for 75%, ending up with only a *single* node when black-holes are used. This means that it would take *AODV-Miner* potentially four times longer to identify malicious nodes when they drop all packets when compared to grey-holes dropping only 25%. This delay would consequently manifest in a lower throughput, as more malicious nodes need to be encountered directly to identify a route. Finally, as already examined previously, a network where 75% of all nodes are beyond hope, even by changing the route constantly in an effort to reach the destination, it is highly unlikely to find a clear route. This is illustrated by the fact that *AODV-Miner* results in a lower throughput for 25% malicious probability than AODV, where the significant presence of malicious nodes simply hinders the overall performance.

7. DISCUSSION AND FUTURE WORK

As we have presented previously, *AODV-Miner* has provided some overall good results. By providing an analysis against various degrees of grey-holes, we have demonstrated the adaptability of our protocol and its ability to cope with different attack scenarios. However, we are aware that this analysis possesses some limitations. Firstly, our system revolves around an emulated lightweight blockchain, basically assimilated to a dissemination tool only. This was motivated to allow us to concentrate further on the validation miners themselves and their activities related to behavioural analysis. Blockchain storage is a well-known challenge when it comes to the IoT, where many applications are turning towards cloud computing strategies to store their data [28]. This means that the blocks themselves in our case are not stored on the nodes due to the inherent hardware limitations of IoT devices. Instead, the information is simply extracted and used to update the *Node Reputation Tables*, before forwarding the blocks onwards. Our consensus-based validation metric also responds to the specificities of IoT devices, reducing computation and energy consumption inherent to the *PoW* concept. Secondly, we only concern ourselves with malicious nodes infiltrating the routing process. This choice was motivated by our interest to demonstrate the efficiency of our module



(a) Throughput with $\alpha = 0.5$

(b) Throughput with $\alpha = 1$

(c) Throughput with $\alpha = 2$

(d) Throughput with $\alpha = 5$

(e) Throughput with $\alpha = 10$

Fig. 15 – Throughput comparison between *AODV-Miner* and AODV with a network of 100 nodes subjected to grey-hole attacks

against such attacks, without the risk of further compromise by a malicious party. However, the protection of the validation process itself is one of our current interests and we are proposing an extension to this module to secure the PoW against malicious miners.

Our consensus-based reputation system has been proposed and evaluated using AODV, since it provides both a simple and efficient platform for analysis. However, our approach has been realised in such a way that it can be applied to every platform respecting certain requirements. Indeed, many new protocols have emerged since its elaboration, each with their own advantages and security integrations. Our next step would be to fully analyse the advantages and functionality of our system with these new protocols, by integrating our consensus-based reputation system into the route decision-making process itself. By comparing these results with our AODV baseline, we can evaluate in a more in-depth context the efficiency and functionality of our system. Furthermore, by deploying our system on real devices, we can extrapolate real-life results from the idealistic simulation environment, as well as evaluate the impact of the implementation itself. Through this experimentation, we can extend our study to encompass further criteria, such as the impact of the overhead on the energy consumption and lifespan of the devices themselves.

8. CONCLUSION

In this paper, we introduced a secure consensus-based routing method using node reputation metrics to identify the most trustworthy route available. The consensus-based validation technique employed allows us to accurately separate malicious nodes from the masses, avoiding them in subsequent communications. Furthermore, by using blockchain as a method for distributing the computed reputation throughout the network, we assure that all nodes receive the correct and valid reputation values for the entire network. Finally, with the application of a reputation decay functionality, we provide the ability for the network to heal itself by reintroducing repaired and salvaged nodes without user intervention. By implementing our module in an AODV-like routing protocol, *AODV-Miner*, and analysing the overall efficiency in multiple scenarios with different network topologies and complexities, we can demonstrate the adaptive capabilities of our network. Through extensive simulations, we have not only proved the increase in security and efficiency of *AODV-Miner* in relation to AODV, but also the importance of reputation-based routing in multi-hop networks. However, a significant increase in overhead forms a necessary trade-off in the strive for increased integrity and security in routing activities.

ACKNOWLEDGEMENTS

This work was partially supported by a grant from CPER DATA and by the European Union’s Horizon 2020 Project “CyberSANE” under Grant Agreement No. 833683 addressing the topic SU-ICT-01-2018.

REFERENCES

[1] L. Pycroft and T. Z. Aziz. “Security of implantable medical devices with wireless connections: The dangers of cyber-attacks”. In: *Expert Review of Medical Devices* 15.6 (2018). PMID: 29860880, pp. 403–406. DOI: 10.1080/17434440.2018.1483235. eprint: https://doi.org/10.1080/17434440.2018.1483235. URL: https://doi.org/10.1080/17434440.2018.1483235.

[2] J. Sengupta, S. Ruj, and S. Das Bit. “A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT”. In: *Journal of Network and Computer Applications* 149 (2020), p. 102481. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2019.102481. URL: https://www.sciencedirect.com/science/article/pii/S1084804519303418.

[3] NARA. *Blockchain White Paper*. White Paper. National Archives and Records Administration, Feb. 2019.

[4] A. M Antonopoulos. *Mastering Bitcoin: Programming the open blockchain.* ” O’Reilly Media, Inc.”, 2017.

[5] S. R. Das, C. E. Perkins, and E. M. Belding-Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. July 2003. DOI: 10.17487/RFC3561. URL: https://rfc-editor.org/rfc/rfc3561.txt.

[6] F. Bao, I.-R. Chen, M.J. Chang, and J.-H. Cho. “Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection”. In: *IEEE Transactions on Network and Service Management* 9.2 (2012), pp. 169–183. DOI: 10.1109/TCOMM.2012.031912.110179.

[7] D. K. Bangotra, Y. Singh, A. Selwal, N. Kumar, and P. K. Singh. “A Trust Based Secure Intelligent Opportunistic Routing Protocol for Wireless Sensor Networks”. In: *Wireless Personal Communications* (2021), pp. 1–22.

[8] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani. “Trust-aware and cooperative routing protocol for IoT security”. In: *Journal of Information Security and Applications* 52 (2020), p. 102467. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2020.102467. URL: https://www.sciencedirect.com/science/article/pii/S2214212619306751.

[9] J. Tang, A. Liu, M. Zhao, and T. Wang. “An aggregate signature based trust routing for data gathering in sensor networks”. In: *Security and Communication Networks* 2018 (2018).

[10] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen. “A survey on the security of blockchain systems”. In: *Future Generation Computer Systems* 107 (2020). DOI: https://doi.org/10.1016/j.future.2017.08.020. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17318332.

[11] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani. “Applications of Blockchains in the Internet of Things: A Comprehensive Survey”. In: *IEEE Com. Surveys Tutorials* 21.2 (2019). DOI: 10.1109/COMST.2018.2886932.

[12] A. Moinet, B. Darties, and J.-L. Baril. “Blockchain based trust & authentication for decentralized sensor networks”. In: *ArXiv* abs/1706.01730 (2017).

[13] Yu Zeng, Xing Zhang, Rizwan Akhtar, and Changda Wang. "A Blockchain-Based Scheme for Secure Data Provenance in Wireless Sensor Networks". In: *2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. 2018, pp. 13–18. DOI: 10.1109/MSN.2018.00009.

[14] C. Machado and C. M. Westphall. "Blockchain incentivized data forwarding in MANETs: Strategies and challenges". In: *Ad Hoc Networks* 110 (2021), p. 102321. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2020.102321. URL: https://www.sciencedirect.com/science/article/pii/S1570870520306752.

[15] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren. "A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks". In: *Sensors* 19.4 (2019). ISSN: 1424-8220. DOI: 10.3390/s19040970. URL: https://www.mdpi.com/1424-8220/19/4/970.

[16] H. Lazrag, A. Chehri, R. Saadane, and M. D. Rahmani. "A Blockchain-Based Approach for Optimal and Secure Routing in Wireless Sensor Networks and IoT". In: *Int. Conf. on Signal-Image Technology Internet-Based Systems (SITIS)*. 2019.

[17] J. Wang, Y. Liu, S. Niu, and H. Song. "Lightweight blockchain assisted secure routing of swarm UAS networking". In: *Computer Communications* 165 (2021), pp. 131–140. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.11.008. URL: https://www.sciencedirect.com/science/article/pii/S0140366420319885.

[18] G. Ramezan and C. Leung. "A blockchain-based contractual routing protocol for the internet of things using smart contracts". In: *Wireless Communications and Mobile Computing* 2018 (2018).

[19] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. "A Specification-Based Intrusion Detection System for AODV". In: *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*. SASN '03. Fairfax, Virginia: Association for Computing Machinery, 2003, pp. 125–134. ISBN: 1581137834. DOI: 10.1145/986858.986876. URL: https://doi.org/10.1145/986858.986876.

[20] S. Bhargava and D.P. Agrawal. "Security enhancements in AODV protocol for wireless ad hoc networks". In: *IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No.01CH37211)*. Vol. 4. 2001, 2143–2147 vol.4. DOI: 10.1109/VTC.2001.957123.

[21] S. Gurung and S. Chauhan. "A Dynamic Threshold Based Algorithm for Improving Security and Performance of AODV under Black-Hole Attack in MANET". In: *Wirel. Netw.* 25.4 (May 2019), pp. 1685–1695. ISSN: 1022-0038. DOI: 10.1007/s11276-017-1622-y. URL: https://doi.org/10.1007/s11276-017-1622-y.

[22] L. Guillaume, J. van de Sype, L. Schumacher, G. Di Stasi, and R. Canonico. "Adding reputation extensions to AODV-UU". In: *IEEE Symp. on Comm. and Vehicular Technology in the Benelux (SCVT)*. 2010.

[23] A. Jarjis and G. Kadir. "Blockchain Authentication for AODV Routing Protocol". In: *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. 2020, pp. 78–85. DOI: 10.1109/BCCA50787.2020.9274452.

[24] C. Ran, S. Yan, L. Huang, and L. Zhang. "An improved AODV routing security algorithm based on blockchain technology in ad hoc network". In: *EURASIP Journal on Wireless Communications and Networking* 2021.1 (2021), pp. 1–16.

[25] M. A. A. Careem and A. Dutta. "Reputation based Routing in MANET using Blockchain". In: *Int. Conference on COMmunication Systems NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450.

[26] E. Staddon, V. Loscri, and N. Mitton. "Attack Categorisation for IoT Applications in Critical Infrastructures, a Survey". In: *Applied Sciences* 11.16 (2021). ISSN: 2076-3417. DOI: 10.3390/app11167228. URL: https://www.mdpi.com/2076-3417/11/16/7228.

[27] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes. "The Contiki-NG open source operating system for next generation IoT devices". In: *SoftwareX* 18 (2022), p. 101089. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2022.101089.

[28] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz. "On blockchain and its integration with IoT. Challenges and opportunities". In: *Future Generation Computer Systems* 88 (2018), pp. 173–190. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.05.046. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17329205.

AUTHORS

Edward Staddon has been a PhD student in the FUN Team at Inria Lille-Nord Europe, France since Oct. 2019, working as part of the H2020 CyberSANE project. He received his B.Sc. and M.Sc. degrees in computer science from Université Bretagne Sud, France in 2016 and 2018 respectively. His research interests include wireless communications, cybersecurity and the Internet-of-Things.

Valeria Loscri has been a permanent researcher at Inria Lille since Oct. 2013. From Dec. 2006 to Sept. 2013, she was research fellow in the TITAN Lab of the University of Calabria, Italy. She received her MSc and PhD degrees in computer science in 2003 and 2007, respectively, from the University of Calabria and her Habilitation à Diriger des recherches in 2018 from Université de Lille (France). Her research interests focus on emerging technologies for wireless communication. She is on the editorial board of IEEE COMST, TNB, Elsevier ComNet, JNCA. Since 2019, she is Scientific International Delegate for Inria Lille-Nord Europe.

Nathalie Mitton received MSc and PhD. degrees in computer science from INSA Lyon in 2003 and 2006 respectively. She has been an Inria full researcher since 2006 and from 2012, the scientific head of the Inria FUN team. Her research interests focus on self-organization from PHY to routing for wireless constrained networks. She has published her research in more than 50 international revews and 120 international conferences. She is involved in the H2020 CyberSANE project and in several TPC such as Infocom, PerCom, DCOSS (since 2019), ICC (since 2015), Globecom (since 2017). She also supervises several PhD students and engineers.