New Trust Metric for RPL Routing Protocol

Nabil Djedjig*, Djamel Tandjaoui†
Faiza Medjek‡, Imed Romdhani§
*†‡Research Center on Scientific and Technical Information-CERIST, 03, Rue des Frres Aissou, Ben Aknoun, Algiers, Algeria.
*†Faculty of Exact Sciences, Abderrahamane MIRA University, Bejaia, Algeria
§Edinburgh Napier University, School of Computing, 10 Colinton Road, EH10 5DT, Edinburgh, UK

***Abstract*—Establishing trust relationships between nodes participating in constructing the routing paths represents a primary security milestone to have reliable routing processes that exclude infected or selfish nodes. In this paper, we propose a new scheme for RPL (Routing Protocol for Low-power and Lossy Networks) named: Metric-based RPL Trustworthiness Scheme (MRTS) to enhance RPL security and deal with the trust inference problem. MRTS addresses trust issue during the construction and maintenance of routing paths from each node to the BR (Border Router). To handle this issue, we extend DIO (DODAG Information Object) message by introducing a new trust-based metric ERNT (Extended RPL Node Trustworthiness) and a new objective function TOF (Trust Objective Function). In fact, ERNT represents the trust values for each node within the network, and TOF demonstrates how ERNT is mapped to path cost. In MRTS all nodes collaborate to calculate ERNT by taking into account nodes' behavior including selfishness, energy, and honesty components. We implemented our scheme by extending the distributed Bellman-Ford algorithm. Evaluation results demonstrated that the new scheme improves the security of RPL.**

***Index Terms*—RPL, Secure Routing, Trust management, IoT security, Internet of Things, 6LoWPAN.**

I. Introduction

Smart, low-power and low-processing objects (things) are able to interconnect, interact, cooperate with each other, and transfer sensing data to the Internet using compatible and heterogeneous wireless technologies, where computing and communication systems are seamlessly embedded [1]. This concept is known as the Internet of Things (IoT) networks. To overcome the connectivity issue of such networks, the 6LoWPAN IETF group [2] introduced the 6LoWPAN adaptation layer (IPv6 over Low power Wireless Personal Area Networks) where different IPv6 header compression techniques are used. The RPL routing protocol for Low-power and Lossy Networks (LLNs) [3] has been designed to handle routing issues in IoT. RPL sits on top of 6LoWPAN layer and other underpinning link-layer technologies (IEEE 802.15.4, Wireless HART, ISA100, etc.). Given that different nodes of the network shall exchange messages with each other to build the routing topology, trust relationships between them must be guaranteed.

In this paper, we introduce a new scheme of RPL that supports the definition of collaborative security [4]. This scheme named: Metric-based RPL Trustworthiness Scheme (MRTS) uses collaborative trustworthiness evaluation between the different nodes within the network. MRTS enhances RPL routing security by calculating and choosing the most trusted path from the source node to the Border Router (BR). To this end, MRTS introduces a trust metric named Extended RPL Node Trustworthiness (ERNT), and a trust-based objective function named Trust Objective Function (TOF). To calculate ERNT, MRTS uses nodes behaviors components: selfishness, energy, and honesty. It is worth mentioning that selfishness, energy, and honesty components have been proposed in the literature in the context of clustering WSNs [5] [6]. They have been adapted by MRTS in the context of IoT (RPL). Hence, in MRTS, each node within the network calculates trust values of its direct neighbors (1-hop neighbors). In addition, it combines the calculated trust values to the trust values received from its collaborative 1-hop neighbors. The final values are used to evaluate paths costs from that node to the BR. Finally, the node selects the 1-hop neighbor on the most trusted path as preferred parent.

The rest of this paper is organized as follow. Section II and III present RPL and related works on trust management for IoT, respectively. Section IV describes and evaluates MRTS scheme. Finally, section V concludes the paper and gives future works.

II. The Routing Protocol for Low-Power and Lossy Networks

The Routing Protocol for Low-Power and Lossy Networks (RPL) [3] has been designed and standardized for constrained and IP-based environments, such as 6LoWPAN networks, and is recognized as the routing protocol of the Internet of Things (IoT). RPL organizes a logical representation of the network topology as a Directed Acyclic Graph (DAG). The DAG is composed of one or more Destination-Oriented DAGs (DODAGs) with one root per DODAG. Each root represents a border router (BR), which is connected to the Internet and to other potential roots via a backbone. Each node in a DODAG has some parameters such as an IPv6 address, a list of parent(s), a list of discovered neighbors, and a Rank. The Rank represents the individual position of a node with respect to the BR and to other nodes. In fact, the Rank values should increase monotonically from the BR towards the leaf nodes, and decrease monotonically from the leaf nodes toward the BR. Furthermore, packets should be transmitted either upward towards the BR, or downward towards leaf nodes.

To optimize DAG paths construction, and to calculate node's rank, RPL uses a set of node/link routing metrics and constraints (i.e. node energy, hop count, throughput, latency, link color, and ETX -Expected Transmission Count) [7]. In addition, RPL uses an Objective Function (OF) that defines how routing metrics and constraints are used to compute node's rank [8] Indeed, routing metrics/constraints, OF, Rank, and other information are conveyed within the DODAG Information Object (DIO) messages. During the construction phase, the BR broadcasts an initial DIO message. This message contains the Rank of the BR, the DODAG ID, the DODAG Version, the OF, Trickle timer, and the metrics/constraints. When a node *n* receives DIO messages from its neighbors, it uses the information conveyed in these DIO messages to join a DODAG. The node selects a set of parents allowing it to reach the BR. Then, it chooses a preferred parent, which ensures traffic routing to the BR. In addition, the node computes its own rank. In the case where the node is a router, it generates and broadcasts a new DIO message to its neighbors. The process will be repeated by all neighboring nodes until each one joins the DODAG. Once the construction is completed, the maintenance begins respecting a Trickle timer mechanism [3]. This timer regulates the transmission rate of DIO messages. Thus, in the steady-case, the interval of the trickle timer increases, and the transmission rate will be slowed. Otherwise, if there are inconsistencies (e.g. altered DIO messages, etc.), which involve changes in the topology, the Trickle timer will be reset to a lower value, and transmission rate will be fastened. To handle inconsistencies, RPL uses global repair and local repair mechanisms. The first one is triggered by the BR, and the second one by any node detecting an inconsistency [3].

III. Related work

In the literature, there are only few works on trust management for IoT. Authors in [9], [10] and [11] proposed a trust management protocol for IoT that calculates the trust level of a node using social relationships metrics: honesty, cooperativeness, and community-interest. Nevertheless, the metrics used in this protocol are calculated using the energy as parameter. As a consequence, if a normal node is surrounded by selfish nodes, it will consume more energy, and it can be considered as non-trusted while it is trusted. In addition, since the protocol uses indirect recommendations, it can be vulnerable to Bad mouthing and Ballot-stuffing attacks. Further the above-mentioned problem, this protocol is mainly oriented toward social IoT environment, and then cannot be used in wide range of IoT applications. Saied et al. [12] proposed a centralized trust approach aiming to create a community of trusted elements to be involved in a collaborative IoT service. In this trust management system, a node intending to set up a collaborative service sends a trustworthiness request to a central unity (trust manager). The trust manager collects trustworthiness information on the context-aware nodes. It then outputs recommendations on nodes to the requesting node. The requesting node relies on the collaborative service provided by the recommended nodes, and assesses the quality of each individual service provision from each assisting node. Finally, the trust manager performs self-updates by learning from past operations to improve future operations. In [13], authors proposed a layered trust mechanism using fuzzy set theory and a formal semantics-based language. In this approach there exist a service requester and a service provider. The IoT is considered as a service provider, and is composed of three layers: sensor layer, core layer, and application layer. The trust management scheme includes three steps: trust information extraction specific for each layer, trust transmission to the next layer, and finally trust decision-making, which is transmitted to the service requester.

Different surveys have been done on trust management for IoT. In [14], authors classified the existing trust management for IoT into five categories depending on the context, the objective and subjective properties of the trustees and the trustors. According to the authors, trust should be ensured vertically in all the layers through different security aspects. These latter include trust evaluation between the entities in all layers, system and entities reliability and availability, privacy and key management, and trust routing and QIoTS (Quality of IoT Services). Also, the trust management, as seen by the authors, should be widely applied to various IoT systems. In [15], authors presented existing trust solutions for the IoT. According to the authors, the trust is a complex notion used in various context with different meanings. They classified trust for IoT into four categories based on: Social networking, Fuzzy methods, Cooperative approach, and Identity-based method. In [16], authors presented a classification of trust computation models for service management in service-oriented IoT systems. This classification contains eight classes based on five trust design dimensions: trust composition (QoS trust, Social trust), trust propagation (Distributed, Centralized), trust aggregation (Belief Theory, Bayesian systems, Fuzzy logic, Weighted sum, Regression Analysis), trust update (Event-Driven, Time-Driven), and trust formation (Single-trust, Multi-trust). Furthermore, the authors presented trust-related attacks which can perturb the trust computation models : Self-promotion attacks (SPA), Bad-mouthing attacks (BMA), Ballot-stuffing attacks (BSA), Opportunistic service attacks (OSA), and On-off attacks (OOA).

Recently, some works have been proposed on trust management for the routing protocol of IoT (RPL). For instance, the Packet Forwarding Indication (PFI) metric was introduced by Karkazis et al. to build trust knowledge as a trust-related metric for RPL [17] [18]. To calculate PFI metric, each node transmits a packet to one of its neighbors, and listens whether this neighbor forwards the packet or not. Then, it calculates the probability for this packet to travel along the path successfully. However, in this approach each node takes a decision based only on its own knowledge. Thus, if this node misbehaves, it will choose a failing path rather than a trusted one. To secure communications in an RPL-based network, authors in [19] proposed to use the classical security mechanisms of a Trusted Platform Module (TPM). In fact, this approach uses TPM to

establish trustworthiness of nodes before exchanging keying material. Furthermore, it provides a secure method to exchange group keys used to secure control messages. Nevertheless, the trustworthiness establishment is done only for exchanging keys securely and not for routing. Furthermore, if a node becomes infected or misbehaves after the establishment of its trustworthiness and the exchange of group keys, it still remains trustworthy against other nodes.

As the classical security approach provided by TPM [19] is not sufficient alone to manage trust in RPL, authors in [20] proposed a so-called trusted-RPL. The aim of trusted-RPL is to enhance RPL security by adding a new trust metric based on nodes behaviors. The trust metric is calculated by the collaboration of different neighboring nodes in the network. However, in this solution, a node within the network selects a path according to its direct neighbors (selects the parent having the greatest trust value). The authors did not consider the trust value along the path (trust inference problem). We do believe that the path taken could not be the most secure. For this reason, we propose a new scheme of trusted RPL, which takes into account trust along the path, i.e. from the node to the BR.

IV. Metric-based RPL Trustworthiness Scheme

In this section, we will depict MRTS functioning. In fact, we propose to complement MRTS approach by a built-in security in the nodes themselves. Therefore, we use a hardware security chip "Trust Platform Module (TPM)", as co-processor embedded to each node within the network. The aim of using the TPM is, firstly, to secure the control messages exchanged during RPL construction (i.e. authentication and cryptography), and secondly, to offload all security computations and processing (i.e. cryptography, and ERNT computations and storage). Table I summarizes different notations used within this paper.

A. Extended RPL Node Trustworthiness metric

The Extended RPL Node Trustworthiness (ERNT) metric represents a quantitative and dynamic routing metric. ERNT is firstly used to evaluate the trustworthiness of each node within the network, and secondly to quantify paths costs. This metric is exchanged between nodes through DIO messages.

1) ERNT evaluation: In the literature, several application-specific methods exist to quantify trust relationships between nodes. In this paper, we have taken as foundation the work of Bao et al. [6], which is flexible and can be adjusted by adding or removing behavioral components specific for a given application. Thus, ERNT is calculated using nodes behaviors components: selfishness, energy, and honesty, and collaboration of neighboring nodes, according to two steps:

*a) **Direct trust evaluation**:* Each node *i* evaluates the trust value, $\mathrm{ERNT_{ij}}(t)$, of its 1-hop neighbor node *j* at time *t* (*t* corresponds to a sending of DIO message, a local repair or a global repair), according to equation 1 [6]:

$$\begin{cases} ERNT_{ij}(t) = w_1 ERNT_{ij}^{honesty}(t) \\ \qquad\qquad +w_2 ERNT_{ij}^{energy}(t) \\ \qquad\qquad +w_3 ERNT_{ij}^{unselfishness}(t) \\ \mathrm{w_1 + w_2 + w_3 = 1} \end{cases} \tag{1}$$

Indeed, $\mathrm{ERNT_{ij}}(t)$ takes values between 0 and 1 (1 refers to complete trust and 0 indicates no trust). $\mathrm{w_1}$, $\mathrm{w_2}$ and $\mathrm{w_3}$ are weights associated respectively to the three trust components: honesty, energy and unselfishness. Each component $\mathrm{ERNT_{ij}^{X}}(t)$, $X \in \{honesty; energy; unselfishness\}$, is evaluated according to equation 2 [6], where Δt is the trust update interval corresponding to the DIO trickle timer; and $\alpha \in [0, 1]$ means that trust evaluation will rely more on direct or more on old observations.

$$\begin{aligned} \mathrm{ERNT_{ij}^{X}(t) = (1-\alpha) ERNT_{ij}^{X}(t - \Delta t)} \\ \mathrm{+\alpha ERNT_{ij}^{X,direct}(t)} \end{aligned} \tag{2}$$

*b) **Indirect trust evaluation**:* The direct trust values from step a) must be composed into a recommendation of how much to trust a given node. Thus, on receiving DIO messages containing propagated ERNTs from its neighbors:

- The node *i* computes $\mathrm{ERNT_{jFinal}}$ for each neighbor node *j*. According to equation 3, $\mathrm{ERNT_{jFinal}}$ is calculated as the average of the direct $\mathrm{ERNT_{ij}}(t)$ trust value, calculated using equation 1, and all ERNTs (i.e. $\mathrm{ERNT_{kj}}$) received for that neighbor *j*. $\mathrm{ERNT_{jFinal}}$ is used to calculate paths costs, and then to select a set of parents and the preferred parent.

$$\mathrm{ERNT_{jFinal} = \frac{ERNT_{ij}(t) + \sum_k ERNT_{kj}}{m}} \tag{3}$$

  In equation 3, $k = N[i] \cap N[j]$, and m represents the number of nodes from which node *i* received ERNTs for that neighbor *j* plus itself (see table I).

- The node *i* calculates its trust value $\mathrm{ERNT_i}$. In fact, each node absolutely trusts itself; that means $\mathrm{ERNT_{ii}} = 1$. Consequently and according to equation 4, $\mathrm{ERNT_i}$ is calculated as the average of $\mathrm{ERNT_{ii}}$ plus all received ERNTs for the node *i* itself (i.e. $\mathrm{ERNT_{k'i}}$) and $k' = N[i]$ and m' represents the number of neighbors of the node *i* plus the node *i* itself (see Table I).

$$\mathrm{ERNT_i = \frac{1 + \sum_{k'} ERNT_{k'i}}{m'}} \tag{4}$$

2) ERNT representation in RPL DIO message: The DODAG Information Object (DIO) carries, information on the metrics and the objective function to use while constructing RPL. To implement MRTS scheme, we introduce an ERNT object in the DAG Metric Container [3] of the DIO message.

As depicted in Figure 1, the ERNT object contains a number of ERNT sub-objects. MRTS uses the ERNT object both as a constraint and as a recorded metric depending on the C flag

TABLE I
TERMINOLOGY

Notation	Description
RPL	Routing Protocol for Low Power and Lossy Networks
TPM	Trust Platform Module
MRTS	Metric-based RPL Trustworthiness Scheme
ERNT	Extended RPL Node Trustworthiness
i, j	Nodes within the network
N[i]	Set of neighbors of node i
N[j]	Set of neighbors of node j
k	Set of neighbors of node i and j (i.e. $k = N[i] \cap N[j]$)
k'	Set of neighbors of node i (i.e. $k' = N[i]$)
m	Number of : neighbors of node i and j plus the node i itself (i.e. $m = \|k\| + 1$))
m'	Number of: neighbors of node i plus the node i itself (i.e. $m' = \|k'\| + 1$
w_1, w_2, w_3	Weights associated respectively to honesty, energy and unselfishness
$ERNT_{ij}(t)$	Trust value of the neighbor node j at time t evaluated by node i
$ERNT_{ij}^{X}(t)$	Trust value of the neighbor node j at time t evaluated by node i for the component $X \in \{honesty; energy; unselfishness\}$
$ERNT_{jFinal}$	Final trust value of the neighbor node j calculated by node i using the collaboration of neighboring nodes k
$ERNT_{kj}$	Trust value of neighbor j received by node i from neighboring nodes k
$ERNT_{k'i}$	Trust evaluation of neighboring nodes k' for the node i received by the node i
$ERNT_{i}$	Average trust value of the node i calculated by node i using the evaluations of neighboring nodes k' for the node i itself
$ERNT_{ii}$	Trust value of the node i with itself. $ERNT_{ii}$ is equal to 1
PC_i	The minimum of on-path nodes' trust values from the source node i to the destination BR
SOP	Set Of Parent

on the DAG Metric Container. In fact, the BR uses an ERNT sub-object as a constraint to indicate a Threshold. Nodes must use this Threshold to include or eliminate nodes that are not trusty. Likewise, the BR and other nodes use ERNT sub-object as a recorded metric. This later represents a scalar which determines the trustworthiness as well as the path cost. Hence, each node participating in the construction of RPL inserts ERNT sub-objects (records). One of the ERNT sub-objects conveys the trust value of the node itself ($ERNT_i$ from equation 4). The second one conveys the path cost regarding the preferred parent of the node itself. The others convey trust values of the neighbors of the node ($ERNT_{jFinal}$ from equation 3).

a) ERNT Sub-Object:

NID. The variable-length *NID* field represents the identifier of the evaluated neighbor *j* or of the node *i* itself. It can be an IPv6 address or a TPM_ID.

NT. When used as a constraint, the *NT* (8 bits) field defines a Threshold, which allows a node *i* to decide if a neighbor *j* can be trusted or not. When used as a metric, it is set to $ERNT_{jFinal}$, $ERNT_i$ or preferred-parent's path cost.

P. The *P* (1 bit) flag indicates if the node *NID* is a preferred parent. If *P* is set to 1 (*P*=1) then the node *NID* is the preferred parent of the neighbor *j*, else if *P* is set to 0 (*P*=0) then the node *NID* is not preferred parent of the neighbor *j*.

I. The *I* (1 bit) flag is set only by the BR. When *I* is set to 1 (*I*=1), the BR enables non-trusted nodes to participate in routing (i.e. non-trusted nodes can be included in the list of parents). When *I* is set to 0 (*I*=0), the BR indicates that non-trusted nodes must be excluded from the list of parents.

T. The *T* (1 bit) flag indicates the node type. It is used if the ERNT object is a metric. When *T* is set to 1 (*T*=1), the node *NID* represents a trusted node (i.e. $NT \geq Threshold$). When *T* is set to 0 (*T*=0), the node *NID* represents a non-trusted node (i.e. $NT < Threshold$).

Flags. The 5 *Flags*' bits remain unused. They must be initialized to zero by the sender and must be ignored by the receiver.

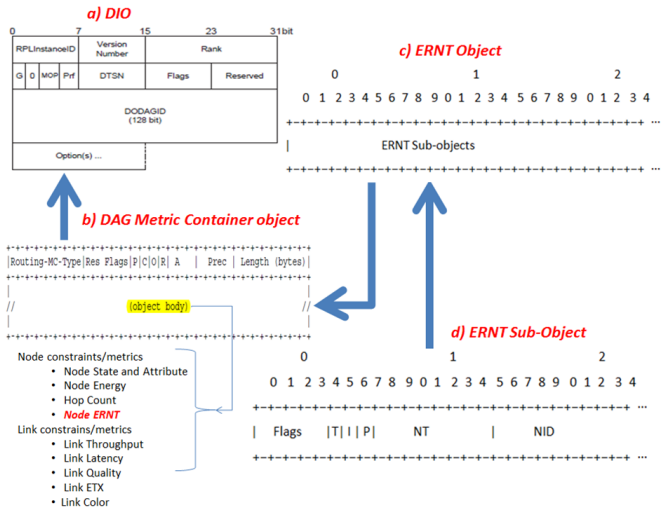

Fig. 1. ERNT Object and ERNT sub-objects within the DIO DAG-Metric-Container

B. Trust Objective Function

The Trust Objective Function (TOF) defines how nodes in MRTS use ERNT metric and constraint to select preferred parent and to calculate Rank. Also, TOF states how ERNT is transformed into path cost, and how this path cost is translated into node Rank. In fact, several MRTS's DODAGs may be used with TOF to find most trusted paths -paths with best trust values- (ERNT as metric) and avoid non-trusted paths -paths with non-trusted nodes- (ERNT as constraint).

1) MRTS Path Cost: To reach the destination (BR), each node *i* computes the Path Cost, $PC_{i,BR}$ (PC_i for short), through each reachable potential parent *j*. PC_i is a scalar value representing node characteristics along end-to-end path. There exist several ways to compute path cost using a trust metric [21] [22] [23]. It is known as trust inference problem. One way is to select the strongest path, determined by the path with the highest minimum value, and take the lowest value on that path [22] [23]. A second way is to choose the strongest path, determined by the path with the highest product of all values on the path, and take the product of all values on that path as path cost [21] [23]. A third way is to select the strongest path, determined by the path with the weighted average of the minima of the trust values along the disjoint paths [23]. Another solution could be to choose the strongest path, determined by the path with the highest average value, and take the average value on that path as path cost. To meet the MRTS routing requirements of consistency, optimality, and loop-freeness [24] [23], we decide to define the path cost PC_i according to the first solution, as the minimum of on-path nodes' trust values from the source node *i* to the destination BR (equation 5). According to equation 6, and recursively, PC_i is calculated as the minimum value between the potential parent path cost PC_j and $ERNT_{jFinal}$ for that parent *j*. When the BR sets the flag *I* to 0, the topology formed by RPL must avoid all non-trusted nodes, and thus avoid paths with non-trusted nodes.

$$PC_i = \min_{j \in \{SOP\} \& ERNT_{jFinal} \geq Threshold} ERNT_{jFinal} \quad (5)$$

$$PC_i = \min_{j \in \{SOP\} \& ERNT_{jFinal} \geq Threshold} PC_j, ERNT_{jFinal} \quad (6)$$

2) MRTS Parent selection: After node *i* evaluated ERNTs for all candidate neighbors *j*, if the Threshold in the ERNT constraint object is not satisfied (i.e. the trust value is less than the Threshold), the advertising node will not be selected as a parent by the node processing the DIO message. If the constraint is satisfied, the processing node *i* adds the advertising node to its set of parents. Then, it evaluates the path cost through each potential parent according to equation 6. Finally, it chooses the parent who is in the path having the greatest path cost as preferred parent. The best path (i.e strongest path) is the one with the highest minimum value, according to equation 7. Hence, among the candidate paths, the selected path can be the longest but remains the most secure (See Figure 2). If some candidate paths have the same path costs then, the processing node will choose as preferred parent the one having the lowest rank.

$$PC_i = \max \min_{j \in \{SOP\} \& ERNT_{jFinal} \geq Threshold} PC_j, ERNT_{jFinal} \quad (7)$$

In the following, we present two examples illustrating path cost calculation and parent selection. In the examples, we note paths from N4 to BR as P^1=<N4,N3,N1,BR> and P^2=<N4,N2,BR>. The node N4 receives DIO messages from nodes N3 and N2. It evaluates $ERNT_{N3}$ and $ERNT_{N2}$, and calculates their respective path costs.

Example 1: Using equation 6, $PC^1_{N4} = 0.6$ and $PC^2_{N4} = 0.5$. According to equation 7, $PC_{N4} = PC^1_{N4}$ ($PC^1_{N4} > PC^2_{N4}$), hence N4 will choose P^1 for routing. (See Figure 2)

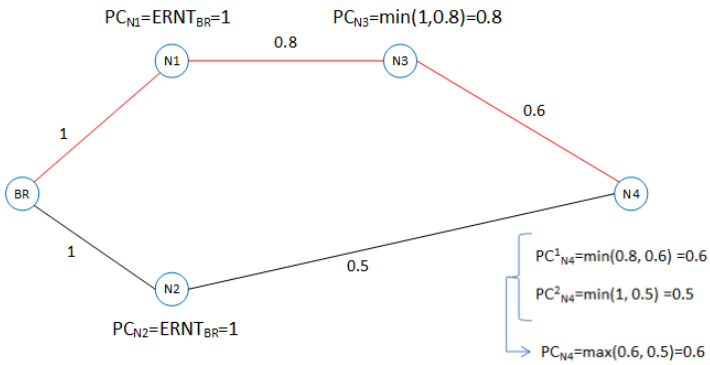

Fig. 2. Node N4 chooses the longest but most trusted path

Example 2: Using equation 6, $PC^1_{N4} = 0.6$ and $PC^2_{N4} = 0.7$. According to equation 7, $PC_{N4} = PC^2_{N4}$ ($PC^2_{N4} > PC^1_{N4}$), hence N4 will choose P^2 for routing. (See Figure 3)

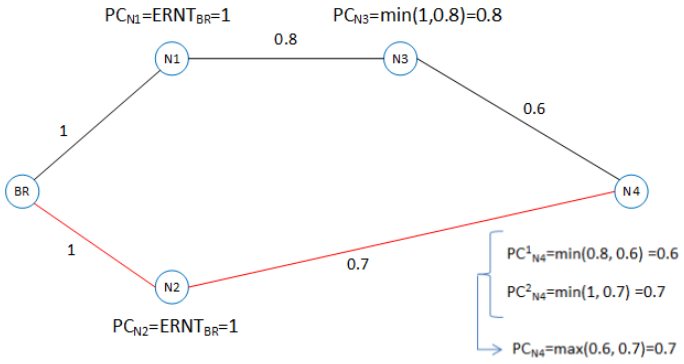

Fig. 3. The node N4 chooses the shortest and most trusted path

3) MRTS Rank calculation: The Rank should monotonically decrease when moving upward to the BR, and monotonically increase when moving downward to the leaf nodes. As well, it should be bounded by *MinHopRankIncrease* and *MaxRankIncrease* [3]. Therefore, to comply with the monotonic property of the Rank, the BR sets its Rank to *MinHopRankIncrease*. Then each node *i* calculates its Rank *R(i)* as the sum of the Rank of its preferred parent *R(PP)* and rank_increase. The node Rank is computed using equation 8. In this equation PC_i is the path cost through the preferred parent *PP*.

$$\begin{cases} R(i) = R(PP) + rank\_increase \\ rank\_increase = (1/PC_i) * 100 \end{cases} \quad (8)$$

4) MRTS calculations: When constructing RPL, the BR declares itself as a Floating root (i.e BR has no preferred parent) [3]. It broadcasts a first DIO message conveying the

ERNT Threshold and the ERNT metric values of its one-hop neighbors. On receiving DIO messages, each neighbor *i* of the BR selects the BR as preferred parent, calculates the rank according to equation 8, sets $ERNT_{BR}$ to 1 (i.e the BR is fully trusted), evaluates $ERNT_{jFinal}$ of the respective neighbors, and $ERNT_i$ of itself, and broadcasts DIO message conveying the ERNT objects. In this case, each neighbor *i* of the BR inserts an ERNT sub-object (Figure 1) with *P*=1, *NT*=1 and *NID*=BR, which means that the BR is the preferred parent and its trust value is equal to 1. This trust value represents the Path Cost PC_i (i.e. at the first stage, $PC_i = ERNT_{BR} = 1$). The process is repeated; each node *i* receiving DIO messages evaluates $ERNT_{jFinal}$ of its neighbors *j* and $ERNT_i$ of itself, calculates PC_i through each potential parent *j* to the BR according to equation 6, selects the preferred parent having the best path cost according to equation 7, calculates the Rank (equation 8), and broadcasts DIO message conveying the ERNT objects. In this case, each node *i* inserts an ERNT sub-object (Figure 1) with *P*=1 and *NT*= PC_i, which means that the node *NID* is the preferred parent of *i* and the path cost through it is equal to PC_i (equation 7).

To compare the various approaches proposed in related work, we realized a comparative table (table II) summarizing the different solutions and classifies them according to the trust properties and trust-related attacks which can perturb the trust computation models. This classification relies on the survey of Guo et all [16]. IT contains trust QoS (T-QoS) and trust social (T-social) metrics, trust propagation (Distributed, Centralised), Weighted sum. we added also Collaboration and Trust inference. Trust-related attacks are [16]: Self-promotion attacks (SPA), Bad-mouthing attacks (BMA), Ballot-stuffing attacks (BSA), Opportunistic service attacks (OSA), and On-off attacks (OOA).

In fact, our solution focuses on RPL routing paths trustworthiness. Compared with existing trust-based solutions for RPL (Related work), our work is based on a distributed trust computation model using the collaboration of nodes in calculating trust to decide the forwarding path to select while handling trust inference problem. Furthermore, our model considers two trust metrics: QoS trust (Energy and Selfishness) and Social trust (honesty). In addition, our scheme can deal easily with different attacks such as Self-promotion, Bad-mouthing, and Ballot-stuffing attacks. This is due to the fact that each node uses several evaluation values received from different neighboring nodes to calculate the trust value of a specific node. Hence, even if a node *i* transmits a bad or a good fake evaluation for another node *j* or itself, the values received from other neibhoring nodes will counter this evaluation (node's *i* evaluation).

C. MRTS Evaluation

RPL is a distance-vector routing protocol, and it uses Bellman-Ford algorithm to calculate path cost [18]. In a weighted directed graph $G(V, E)$, this algorithm computes shortest paths from a source node to a destination node. It is able to handle graphs in which some of the edge weights are negative numbers. To evaluate MRTS, we implement a Linux-based paths simulator. In the first part of the simulator, we implement the standard RPL using ETX metric. In the second part, we implement MRTS based on a distributed extended version of Bellman-Ford algorithm using ERNT metric - MRTS's Bellman-Ford algorithm- (See Algorithm 1). In our evaluation, MRTS-based network is defined as a directed weighted graph $G(V, E)$, where V is the set of nodes, and E is the set of edges representing links between neighboring nodes. Each edge, $e = (i, j)$ is associated to a positive weight corresponding to $ERNT_{jFinal}$, where $e \in E$, $i, j \in V$, and $ERNT_{jFinal}$ is the final trust evaluation of node i for its neighbor node j. Hence, as inputs to Algorithm 1, we have the network graph G, the function to calculate paths costs f, the source node s, and the destination BR.

Algorithm 1 MRTS(G,f,s,BR)

Input: G, f, s, BR
Output: PC[], p[]
function BellmanFord(G(V, E), s, BR)
1) Step 1: Initialize graph
foreach vertex $v \in V$ **do**
PC[v]$\leftarrow$ inf
p[v]$\leftarrow$NIL
end foreach
PC[s]$\leftarrow$0
p[s]$\leftarrow$s
2) Step 2: Relax edges repeatedly
for i **from** 0 **to** $|V| - 1$ **do**
foreach $(u, v) \in E$ **with weight** $(1 - ERNT_{vFinal})$ **do**
if $((v \in Parent[u])$ **and** $(ERNT_{vFinal} \geq Threshold))$ **do**
if $(PC[v] > max(PC[u], (1 - ERNT_{vFinal})))$ **do**
$PC[v] \leftarrow max(PC[u], (1 - ERNT_{vFinal}))$
p[v] $\leftarrow$ p[u]$\oplus$(u,v)
end if
end if
end foreach
end for
Return PC[], p[]

We specify that in our evaluation we consider an initial representation of a network of 13 nodes represented in Figures 4 and 5, where the BR is the only destination and the trust Threshold is set to TH=0.5. The Figure 4 displays a network where values on edges represent ETX-link values. While, the Figure 5 displays a network where values on edges represent mutual $ERNT_{ij}$ evaluations of neighboring nodes (from equation 1). It is obvious from evaluations in Figure 5

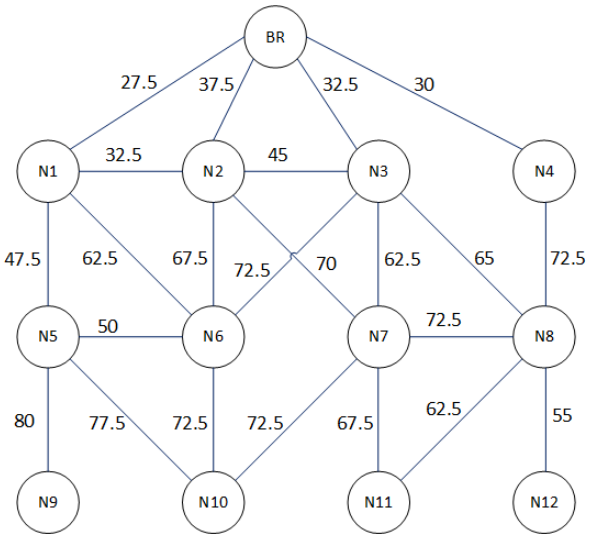

Fig. 4. Network using *ETX* values

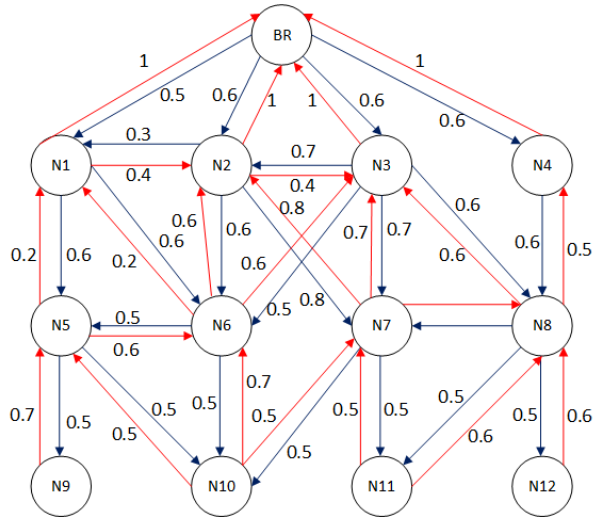

Fig. 5. Network using $ERNT_{ij}$ values

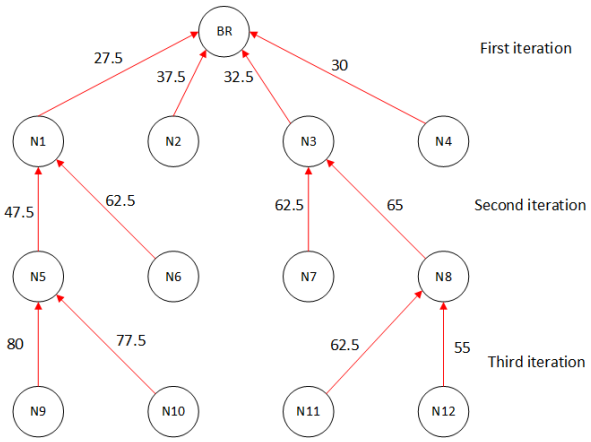

Fig. 6. RPL topology using *ETX* values

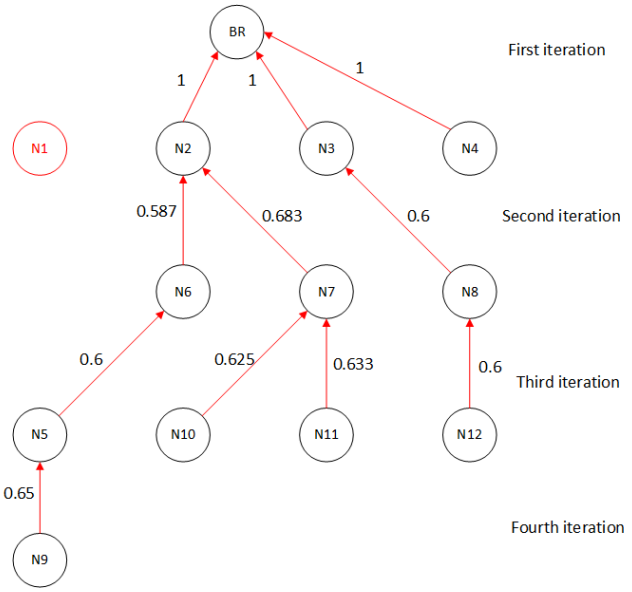

Fig. 7. RPL topology using ERNT values

TABLE II
SYNTHESIS OF EXISTING TRUST-BASED SOLUTIONS

	RPL	T-QoS	T-Social	Distributed	Centralised	Wight sum	SPA	BMA and BSA	collaboration	T-inference
[9] [10] [11]		X	X	X	X	X	X			
[12]		X			X	X	X	X		
[13]					X					
[17] [18]	X			X						
[19]	X			X						
[20]	X	X	X	X		X	X	X	X	
MRTS	X	X	X	X		X	X	X	X	X

that the node N1 is untrusted. Figure 6 represents the network topology of RPL using Bellman-Ford algorithm with ETX metric. Whereas, Figure 7 represents the network construction using ERNT and MRTS (Algorithm 1), where values on edges represent $\text{ERNT}_{\text{jFinal}}$ evaluations of neighboring nodes (from equation 3). Comparing Figures 7 and 6, we notice clearly that the routing topology has completely changed (we have 4 levels instead of 3, respectively). This is due to the fact

N1 is untrusted. So, in the case of using ETX (standard RPL in Figure 6), each node selects the path with minimum total ETX, and thus can forward packets through the non-trusted node N1. Consequently, in a network of 13 nodes there are 4 nodes (N5, N6, N9, N10) that use the wrong paths, which represents the third of the network. However, when we use MRTS to construct the routing topology (see Figure 7), N1 is avoided and the selected paths are more secure. We extend our simulations to 51 nodes. We notice that for standard RPL every time the number of malicious node increases, the probability that the routing paths uses malicious nodes also increases. Nevertheless, in MRTS construction, every time the number of malicious node increases, the topology changes while avoiding malicious nodes. Hence, malicious nodes could not participate in the network operations and trigger attacks because they had already been avoided.

V. Conclusion

In this paper, we proposed a trust management scheme to secure routing in RPL. This new trust-based-RPL, namely, MRTS is based on a distributed and collaborative trust model, where nodes' behaviors are used to evaluate nodes' trust values. The trust value is named ERNT trust metric. After ERNT collaborative evaluation, our scheme MRST considers only the trusted nodes. The nodes path cost in MRTS aids the routing discovery process to set up secure routing paths. Compared with standard RPL, the MRTS's routing algorithm shows better performance in term of trustworthiness, according to results of the evaluation experiments. In fact, MRTS allows a secure network self-organization based on nodes trust status. In our future work, we project to implement and evaluate MRTS with respect to energy consumption, routing and security overheads by using Cooja-contiki simulator.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Rfc 4944," Transmission of IPv6 packets over IEEE, vol. 802, no. 4, 2007.

[3] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "Rpl: Ipv6 routing protocol for low-power and lossy networks," RFC 6550, Internet Engineering Task Force, 2012.

[4] J.-M. Seigneur, Collaborative Computer Security and Trust Management. IGI Global, 2009.

[5] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its application to trust-based routing," in Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, 2011, pp. 1732–1738.

[6] F. Bao, R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," Network and Service Management, IEEE Transactions on, vol. 9, no. 2, pp. 169–183, 2012.

[7] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low power and lossy networks," RFC 6551, Internet Engineering Task Force, 2012.

[8] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," RFC 6552, Internet Engineering Task Force, 2012.

[9] F. Bao and I.-R. Chen, "Trust management for the internet of things and its application to service composition," in IEEE WoWMoM 2012 Workshop on the Internet of Things: Smart Objects and Services. IEEE, 2012, pp. 1–6.

[10] I.-R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," 2015.

[11] R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," IEEE Transactions on Dependable and Secure Computing, vol. 13, pp. 684–696, 2016.

[12] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the internet of things: a context-aware and multi-service approach," Computers & Security, vol. 39, pp. 351–365, 2013.

[13] J. P. Wang, S. Bin, Y. Yu, and X. X. Niu, "Distributed trust management mechanism for the internet of things," Applied Mechanics and Materials, vol. 347, pp. 2463–2467, 2013.

[14] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," Journal of Network and Computer Applications, vol. 42, pp. 120–134, 2014.

[15] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," Computer Networks, vol. 76, pp. 146–164, 2015.

[16] J. Guo, R. Chen, and J. J. Tsai, "A survey of trust computation models for service management in the internet of things systems," Computer Communications, vol. 97, pp. 1–14, 2017.

[17] P. Karkazis, H. C. Leligou, L. Sarakis, T. Zahariadis, P. Trakadas, T. H. Velivassaki, and C. Capsalis, "Design of primary and composite routing metrics for rpl-compliant wireless sensor networks," in Telecommunications and Multimedia (TEMU), 2012 International Conference on. IEEE, 2012, pp. 13–18.

[18] P. Karkazis, I. Papaefstathiou, L. Sarakis, T. Zahariadis, T.-H. Velivassaki, and D. Bargiotas, "Evaluation of rpl with a transmission count-efficient and trust-aware routing metric," in Communications (ICC), 2014 IEEE International Conference on. IEEE, 2014, pp. 550–556.

[19] S. Seeber, A. Sehgal, B. Stelte, G. D. Rodosek, and J. Schönwälder, "Towards a trust computing architecture for rpl in cyber physical systems," in CNSM, 2013, pp. 134–137.

[20] N. Djedjig, D. Tandjaoui, and F. Medjek, "Trust-based rpl for the internet of things," in 2015 IEEE Symposium on Computers and Communication (ISCC). IEEE, 2015, pp. 962–967.

[21] J. Golbeck, "Trust on the world wide web: a survey," Foundations and Trends in Web Science, vol. 1, no. 2, pp. 131–197, 2006.

[22] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," Selected Areas in Communications, IEEE Journal on, vol. 24, no. 2, pp. 318–328, 2006.

[23] C. Zhang, X. Zhu, Y. Song, and Y. Fang, "A formal study of trust-based routing in wireless ad hoc networks," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–9.

[24] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in INFOCOM 2008. The 27th conference on computer communications. IEEE. IEEE, 2008.