Trust management in Industrial Internet of Things

Chaimaa Boudagdigue, Abderrahim Benslimane, *Senior Member, IEEE,*
Abdellatif Kobbane, *Senior Member, IEEE,* Jiajia Liu, *Senior Member, IEEE*

***Abstract*—Automobile manufacturers around the world are increasingly deploying Industrial Internet of Things (IIoT) devices in their factories to accompany the Industrial Revolution 4.0. Security and privacy are the main limitations to the integration of Internet of Things (IoT) into industrial processes. Therefore, it is necessary to protect industrial data contained in IIoT devices and keep them confidential. As a step towards this direction, in this paper, we propose a dynamic trust management model suitable for industrial environments. We propose also to change the traditional centralized architecture of IIoT networks in automotive plants into a hybrid architecture based on a set of new industrial relationship rules. The performance evaluation in this work is done in two parts. In the first part, we compare our proposed architecture with the traditional architecture of the plant's IIoT network. The results of this comparison show that our architecture is more suitable to simplify trust management of IIoT devices. In the second part, we demonstrated the ability, the adaptiveness and the resiliency of our proposed trust model against behavioral changes of IIoT nodes in malicious environments.**

***Index Terms*—Internet of Things, Industrial IoT networks, Automotive factories, Security, Trust management.**

I. Introduction

TO accompany the industrial revolution 4.0 ([1], [2], [3]) all automobile manufacturers rely heavily on the Internet of things in order to change the traditional architecture of their plants into a fully automated and highly connected architecture. IIoT devices (sensors, actuators, robots, etc.) will be deployed in all automotive industrial services and processes: manufacturing, maintenance, monitoring and other tasks, making them increasingly attractive targets for hackers. Malicious competitors can target alarm systems, robots and connected machines either to harm the production or to disclose data related to the industrial processes. It is necessary to monitor the plant's IIoT network continuously.

Integrate security protections [4], [5] into IIoT devices is very important but it is preferable to strengthen it by securing the whole plant's IIoT network. An IIoT system can behave in untrustworthy manner even after implementing all necessary security and confidentiality measures in its IIoT devices [6], [7], [8] as the Stuxnet worm attack [9] on Iran's nuclear installations showed in 2010. This virus destroyed a large number of centrifuges in the Natanz power station by slightly disrupting their operations. This type of attack can be produced in any industrial plants including automotive plants. Indeed, behavior based analysis of IIoT devices is required in order to predict the performance of the devices over time. Trust management provides behavior based analysis of IIoT devices by using their past behavior and their reputation in the network. It is crucial to prevent from unwanted activities conducted by compromised devices.

Usually, trust is managed either in a distributed manner, where each node evaluates the trust metrics of its neighbors, or in a centralized manner, where a single trust management entity manages the entire IIoT network. However, a single centralized trust management entity is not able to continuously manage the trust in the plant's IIoT network composed of a large number of heterogeneous and sensitive devices. Admittedly, distributed trust management systems are the most widely used in the composition of services in IoT but they are not appropriate to manage the trust in industrial environments.

The ultimate solution is to change the traditional architecture of plant's IIoT network by grouping the IIoT devices into clusters in order to simplify trust management. In Social Internet of Things (SIoT) [10], SIoT objects build their social communities according to their common social relationships [11]. However, social relationships cannot be established between IIoT devices, hence the need to define new and specific relationships more appropriate to industrial environments.

Inspired from SIoT, our contribution is three fold. In the first contribution, we define a new concept called industrial relationships where IIoT objects are able to establish industrial relations between them. Instead of traditional architecture of the plant's IIoT networks, the industrial relationships define a new hybrid architecture H-IIoT constituted of groups called industrial communities. Within the same community, the evaluation of trust becomes more accurate because it will be calculated and defined in the same context. Hence, each community is monitored by a trusted leader which calculates the trust of nodes and returns the results to the IIoT server to record them. The trust is evaluated according to three performance metrics namely, cooperation, direct and indirect honesty. In the second contribution of this paper, we use the three calculated metrics to propose a dynamic trust management model Tm-IIoT. The Tm-IIoT model is suitable for industrial environments. In the last contribution, we implement our proposed H-IIoT architecture by using contiki cooja simulator, we demonstrate also the ability and the adaptiveness of the proposed Tm-IIoT model to detect trust attacks established by malicious and compromised nodes.

The rest of the paper is structured as follows. In section II, we discuss some related trust models proposed for IoT and IIoT networks. We describe the future automotive factory architecture in section III. The Tm-IIoT model is proposed in section IV. We present the performance evaluation, the

C. Boudagdigue and A. Benslimane are with LIA/CERI, University of Avignon, 84911 Avignon, France.

A. Kobbane is with ENSIAS, Mohammed V University, Rabat, Morocco.

J. Liu is with the Scool of Cyber Engineering, Xidian University, Xi'an 710071, China.

*corresponding author: Abderrahim Benslimane (E-mail:benslimane@ieee.org)

comparisons, the simulation results in section V. Finally, in section VI, we conclude the paper.

II. Related work

The use of trust management to monitor IIoT devices in automotive factories was negligible and was not considered as a main concern by researchers. Indeed, in these factories, IIoT devices are considered as traditional IT devices, their security is managed by traditional security measures while they have limited resources and contain sensitive production data. Therefore, IIoT devices require a continuous behavior monitoring by using the trust management.

To make decisions, the IIoT server should have the global trust metric of each IIoT node in the network. However, in distributed trust management models [[14], [15], [16], [17], [18], [19], [20]] each IoT node calculates the trust metrics of its neighbors and stores them locally in order to use them for its own interests: service composition [18], decision making [21] or access control [22]. In order to have a global trust metric, it is necessary to gather local trust metrics calculated for each node. This process is very demanding in terms of energy consumption and also in terms of time wasted to have the final convergence trust metrics. In the case of centralized trust management models [[14], [19], [23]], just one trust management entity must monitor, alone, the trust of all IIoT devices in the network. This model is very difficult to apply when the IIoT network becomes larger. In conclusion, whether centralized or distributed models cannot be applied to manage the trust in IIoT plants.

The ultimate solution is to combine the two types of models in order to have a hybrid trust management model based on a clustering. The authors in [24] propose a scalable hierarchical trust management solution for IoT environments based on clustering. The trust metric of each cluster node is managed by a master node and it is collected from peer cluster nodes. An algorithm is proposed to eliminate outliers, and the total trust value is calculated as an average. This work is vulnerable to coalition attacks because the proposed algorithm makes its decisions based on the evaluation given by the majority of cluster nodes. The proposed algorithm has as a majority the evaluations given by the malicious cluster nodes, therefore, it eliminates the evaluations given by the good and fair nodes. This model also lacks accuracy in the calculation of trust values as the evaluations are not carried out in a well-defined context. Consequently, it is necessary to have a context between the monitored and the monitoring nodes in order to have a high accuracy when managing the trust.

Authors in [25] proposed a clustering architecture based on the similarity of interest and the relationships between objects to create a trust management context. By analyzing this work, in order to construct the communities of interest, authors considered the social relationship between owners instead of considering the social relationships between objects. The choice of belonging to a particular community is made by human, therefore the objects are not considered autonomous, which contradicts the objectives of SIoT concept. To elect the leader of a community, the authors in [25] considered the following metrics: trust level, capability and scalability. Scalability promotes nodes with a large number of friends to be elected as a leader, which makes the model highly vulnerable to coalition attacks e.g. a set of malicious nodes can meet between them and elect a malicious leader. This malicious leader will then perform vulnerable attacks in the network like bad-mouthing and ballot stuffing attacks [26].

Authors in [18] proposed a trust management protocol to support service composition in SOA-based IoT systems. They developed a technique based on distributed collaborative filtering to select feedback from owners of IoT nodes sharing similar social interests. To measuring social similarity, the authors rely on three social relationships, i.e., friendship, social contact, and community of interest. To compute similarity rates, devices exchange in clear their profiles (friend list and location list of their owners) which does not preserve the privacy of users and allows their traceability.

Although the SIoT is currently an integral part of our daily lives, it is rarely used to implement an efficient and robust trust management model [27].
Authors in [11] were the first to consider social relationships in trust management for IoT networks. They proposed a new protocol based on three trust factors: honesty, cooperativeness and community interest. The authors in [11] have calculated cooperativeness value as the ration of the number of common friends over the total number of friends between two nodes. The calculation of cooperation in this work is very subjective and the fact that two nodes are friends does not really reflect their willingness to cooperate together [26].

The biggest limitations of this work are the energy efficiency and the adaptability of the protocol. To improve this work, authors in [28] reused the same trust metrics as work [11] and they take into consideration other aspects such as the scalability, the adaptability and the survivability of the protocol. As in [11], the update of trust metrics is always events-driven and the trust metrics are computed only for a limited set of nodes to minimize computation and to ensure scalability. However, a new storage management strategy is proposed, it permits to use limited storage space and enhance scalability.

Several works have used blockchain technology [[12], [13]] to propose secure trust management systems. This technology concerns distributed trust management systems that need to securely store and exchange trust scores between nodes within the network. Blockchain technology cannot be used to manage trust in the plant's IIoT network because distributed models are not appropriate to manage the trust in these environments for all the reasons explained above.

The proposed work in [29] allows to build a reliable trust management model based on the behavior of IoT devices. Each node in the network computes the trust metrics of its friends based on its own experience and on the opinion of its common friends. The basic trust parameters used for calculating trust metric in this work are: Feedback system, transaction factor, total number of transactions, relationship factor, computation capability, credibility and centrality. This work is scalable and resilient against self-promoting, bad-mouthing and ballot-stuffing attacks but it does not ensure the power efficiency and survivability [26].

The trust management protocols proposed in [11], [28] and [29] are very greedy in terms of energy consumption. They are based on the change of state of IoT nodes (monitored monitoring and vice versa) which can cause a fast drainage of the node battery. In these works, there is no differentiation between nodes and any device can be a monitor node even small sensors with very limited resources. The evaluation of trust metric in these works is very subjective, each node calculates the trust metric of its neighbors according to its own context, stores it locally and uses it in case it needs to interact with other nodes in the network.

In our previous work [30], we proposed a distributed trust management model for IoT environments based on three performance metrics namely, cooperation, direct honesty and indirect honesty. However, distributed systems cannot be used to properly manage the trust in IIoT environments. To solve this problem, in this paper, we propose a suitable trust management model for IIoT environments, which is based on hybrid architecture. We will use the same performance metrics as in [30] but they will be evaluated differently according to the IIoT context as shown in section IV phase III. For all these reasons, we would like to mention that the core idea of the two works is totally different.

In the light of the existing works, there is a huge gap between the real concept of IIoT and the existing trust management models. As mentioned above, the existing trust management models cannot be applied in the industrial automotive factories since the main characteristics of IIoT are not taken into consideration. In this paper, we will define a new type of relationships called industrial relationships to establish an appropriate trust management model in automotive plants.

III. Future architecture of automotive plants

According to the official websites of several car manufacturers ([31], [32], [33], [34]), in their future factories, everything will be smart, autonomous and traceable. IIoT devices will communicate between them to be easily located and recovered. IIoT will be used to better manage the plant's inventory; smart devices will monitor the stock availability and will send the report to the plant's inventory management system, which will automatically communicate with suppliers to order necessary parts.

In case of complicated maintenance to be done, technicians can easily contact the experts by using their smart devices (connected tablets, watches...) to intervene remotely. They can also use their connected glasses to follow the expert's constructions in order to update the machines in the factory. The technician's connected glasses will allow the expert to monitor all the technician's manipulations and intervene if necessary.

In the factory, there are also connected robots that support production and send report about the number of vehicles produced per time period. Even after its delivery, the vehicle remains connected with the after-sales service, it sends periodic reports about the technical condition of engine and parts. In case of problem, the customer will be notified to do the necessary maintenances. This will allow the automobile manufacturer at the same time to improve its after-sales service as well as improve the choice of their suppliers. Factory temperature sensors will be connected to the fire department, once the temperature exceeds the limits, the water-filled pipes will be made available throughout the factory and a report will be sent back to the fire department to intervene.

In order to monitor installations and accesses, connected IP cameras will be installed everywhere in the factory to guarantee remote and real-time monitoring. In the automotive plant, smart meters will be installed everywhere to measure the energy consumption of the plant's machines, production lines and factory installations in real time.

As explained above, automotive manufacturers are using IIoT in sensitive applications that are becoming increasingly attractive targets for hackers. Malicious concurrent may target IIoT devices to harm production or to disclose trade secret data. Only one compromised device can compromise the security of the entire plant's IIoT system and cause financial damage. It is necessary to monitor the plant's IIoT network continuously.

To simplify this task, we chose to organize the plant's IIoT network by defining a new concept called industrial relationships where objects can establish industrial relations between them. The industrial relationships used in our contribution are defined as follows:

- Parental relationship (PR): exists between objects which have the same technological characteristics and capabilities in the network.
- Co-worker relationship (CWR): binds devices that perform the same tasks or that always collaborate together to provide intelligent services.
- Ownership relationship (OR): exists between objects belonging to the same employee or the same production line.
- Social relationship (SR): is defined by the social relations (hierarchy relations, collaboration relations...) that exist between the owners of the IIoT devices.

The monitor node easily detects the behavior changes of a monitored node when they are linked by CWR relationship since they work together to achieve the same objective. In order to have more accurate calculation of trust metric Tm, and more homogeneous clusters, this relationship can be strengthened by OR and SR relationships. The calculation of trust becomes more accurate when the monitor and the monitored nodes are linked by several industrial relationships. The use of IIoT in the automotive industry involved only four industrial relationships. Maybe some users will be interested with these proposed relationships for given industrial applications like (oil and gas industry, transportation, healthcare...), users can use these relations or update them according to the requirements of their industrial applications.

IV. The proposed trust model analysis

As discussed above, we aim to change the traditional architecture of IIoT networks in automotive plants in order to simplify trust management. Hence, the IIoT network will be divided into clusters called industrial communities as depicted

Notations

Tm	Trust metric
CL_j	Community leader
MN_i	Member node
$PR(MN_i, CL_j)$	Parental relationship between the MN_i and the CL_j nodes
$CWR(MN_i, CL_j)$	Co-worker relationship between the MN_i and the CL_j nodes
$OR(MN_i, CL_j)$	Ownership relationship between the MN_i and the CL_j nodes
$SR(MN_i, CL_j)$	Social relationship between the MN_i and the CL_j nodes
fi	Monitoring frequency for node i
fi_r	Recommended frequency for node i according to the sensitivity of the data it manages
fi_N	Recommended frequency by the network and the community parameters
Ds	The number of member nodes in a community j.
Tr	Monitoring period for which Tms of all IIoT devices in a community must be updated and registered in the IIoT server
$T0$	Initial trust metric
DI	Designation indicator
Er	Energy reserved for monitoring
CPm	Computing power specific for each node
Cm	Centrality of a node in the network
D	Distance threshold set by the IIoT server
SI	Selection indicator
$d(MN_i, CL_j)$	Physical distance between MN_i and CL_j nodes.
$A(N_i)$	Activity list of node N_i
$S(N_i)$	Social friends list of node N_i
E	State that corresponds to the highest level of trust
Y_t	Random variable, it is the current trust metric of an IIoT device at time t.
P	Transition matrix corresponding to the state transition diagram
$P_{i,j}(t)$	Probability to transit from state i to state j
C_{MN_i}	Cooperation rate of MN_i
NF	The total number of messages transmitted to a member node by its leader
Pm	Percentage of malicious nodes
C_{mi}	Number of the successful forwarded messages
pc	Probability that the MN_i cooperates in the network
$pnet$	Probability that reflects the constraints related to the nature of wireless networks
D_{MN_i}	Direct honesty rate of MN_i node
$A(i)$	List of activities that node i must perform based on its profile.
$AF(MN_i)$	List of feedback activities performed by the MN_i node
pdh	Probability that the monitored MN_i is honest
I_{MN_i}	Indirect honesty rate of MN_i node
$RMN_{i'}, MN_i$	Reputation given by the member node $MN_{i'}$ to the node MN_i
N	Total number of MN_i nodes after eliminating spam nodes
pih	Probability that the monitored MN_i node has a good reputation
BER	Bit error rate
AT	Designation indicator threshold set by the IIoT server
$C(CL_j)$	Community of a CL_j node
RI	Relationship indicator

in Fig.1. It is composed of K community leaders and L member nodes. Let $CL_j|j\ in1, 2, , K$ and $MN_i|i\ in1, 2, , L$ represent the set of community leaders and member nodes, respectively.

Community leader CL_j manages the trust metric Tm of each member node MN_i in its community by calculating three performance metrics: cooperation, direct honesty and indirect honesty. CL is a specific node that must have high level of trust, sufficient computing power, storage and energy resources to perform monitoring tasks.

Each node that wants to join the network must send a request to the IIoT server. IIoT server assigns to nodes a unique identifier and creates a profile for them. The profile is stored in the IIoT node and in the databases of the IIoT server. It sets the activities that can be performed by the IIoT node (such as measurements, transactions, operations...) and the data that can be shared by each node in the network. It determines also the types of the industrial relationships that can be established between two nodes. Based on these defined rules, each node can allow to start, update or terminate any relationships with another node.

Each node that wants to be a CL must communicate with the server to take authorization. According to a set of criteria explained below in phase I, the server gives or not the permission to a specific node to be leader. Once a node becomes a CL, it builds its community. CL contacts the server either to send to it the results of its monitoring or to retrieve data needed to evaluate the Tm of its community members.

The following section is decomposed into three phases. The first phase concerns the designation of CLs based on several parameters. The second phase is the formation of the industrial communities around the designated CLs based on physical distances and industrial relationships. The last phase concerns the monitoring process where CL_j node manages the trust metric Tm of its community members in order to detect suspicious behaviors.

A. *Phase I: Designation of CL*

The trust of a monitored node MN_i is evaluated according to the cooperativeness, the honesty and the monitoring

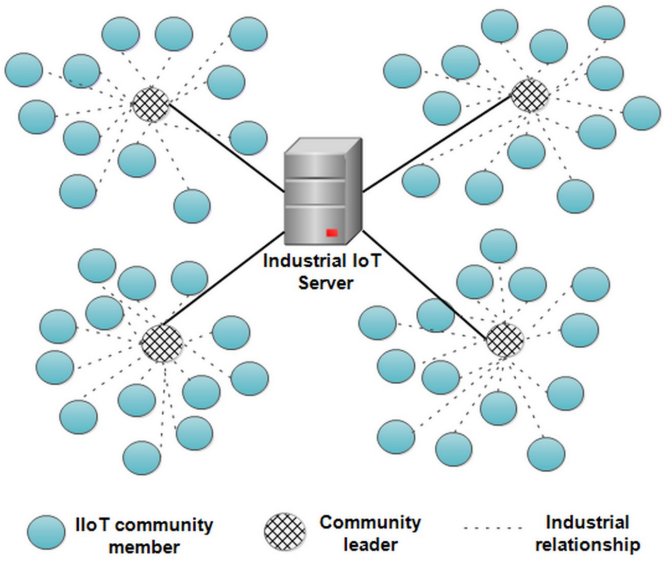


Fig. 1: Proposed H-IIoT architecture

frequency of this node. The monitoring frequency depends on the sensitivity of data managed by the monitored nodes; IIoT nodes containing production-sensitive data must be continuously monitored with a higher frequency compared to nodes containing less sensitive data. It depends also on the community parameters: the number Ds of nodes which must be monitored by the same CL node and the period Tr for which Tms of all IIoT devices in the community must be updated and registered in the IIoT server. The monitoring frequency for a node i will be calculated as follows:

$$fi = \kappa fi_r + (1-\kappa) fi_N \tag{1}$$

Where fi_r is a recommended frequency for node i according to the sensitivity of the data it manages, fi_N is a recommended frequency by the network and the community parameters. fi_N is the frequency that must be respected to ensure that all nodes in the IIoT network will be managed within a fixed period Tr. It is calculated as follows:

$$fi_N = \left(\frac{Ds}{Tr}\right) \tag{2}$$

The selection of $(0 \leq \kappa \leq 1)$ is important to setting the order of monitoring, it is used to weigh the sensitivity of IIoT nodes relative to the network parameters in the evaluation of the monitoring frequency. Monitoring process can set the parameter κ to a high value in order to ensure continuous monitoring of critical nodes and therefore have a better accuracy of trust calculation. Parameter κ is set to a smaller value when the monitoring process aims just to ensure that all nodes in the network will be monitored in a specific time period Tr regardless of their sensitivity.

Each CL_j node will need to continuously monitor the behavior of its MN_i nodes; it compares their current behavior with their profiles stored in the database of the IIoT server. CL must have a high level of trust to manage its members, it must have a sufficient reserved energy to perform monitoring operations and to communicate with the server and it must have also a sufficient computing power to calculate cooperation and honesty rates necessary for the evaluation of the Tms.

Before the implementation of the H-IIoT architecture, designation of CLs and construction of communities, the plant's IIoT network will be deployed in order to nodes acquire knowledge about their neighbors, each node has an initial trust metric $T0$. In this deployment phase, it is assumed that the trust management architecture of the plants IIoT network is traditional i.e. centralized architecture, where the IIoT server is the only entity in the IIoT network that manages the Tm of nodes. When the Tm of a node reaches a high level, the server asks this node to calculate its designation indicator DI according to equation (3). This node will probably be designated as a leader if its DI checks some other conditions explained later in this phase. DI evaluates the ability of a node to properly perform the leader's tasks. Only nodes with high Tm will be contacted by the server to calculate and send their DI, hence, none of these nodes can announce false data, e. g. a high DI value since they are trusted nodes. If a node does not send its DI, it will be penalized and judged by the server as a selfish node because it does not participate in the monitoring process. In order to ensure secure communications between IIoT devices, we have chosen to adopt the same secure and lightweight certificateless signature scheme for IIoT environments proposed in [35]. The designation indicator DI has no unit, its value ranges from 0 to 1. It is calculated as a follows:

$$\begin{aligned} DI =& \alpha Tm + \beta\left(\frac{Er}{max_{CL_j}(Er)}\right) + \lambda\left(\frac{CPm}{max_{CL_j}(CPm)}\right) \\ &+ \omega\left(\frac{Cm}{max_{CL_j}(Cm)}\right) \end{aligned} \tag{3}$$

Where Tm is the node's trust metric, it is unitless and its value is between 0 and 1. Er is the energy reserved for monitoring; we assume that each node has a reserved energy just for monitoring, CPm is the computing power specific for each node, it reflects the number of instructions that the node's CPU can perform per unit of time. Cm designates the centrality of a node in the network, it is relative to the number of neighbors of this node. Cm is unitless, it ensures the uniform distribution of the CL nodes in the network and it avoids them to be placed in the extremities.

$max_{CL_j}(CPm)$, $max_{CL_j}(Er)$ and $max_{CL_j}(Cm)$ are respectively the maximum power, the maximum energy, and the maximum neighbors that a CL_j node can have. They are used to normalize the designation indicator DI.

The weight $\alpha + \beta + \lambda + \omega = 1$ and $0 \leq \alpha, \beta, \lambda, \omega \leq 1$. In practice, the monitoring process give weight (α, β, λ, ω) on each component (Tm, Er, CPm, Cm) according to its importance and according to the network conditions. Indeed, a CL_j node must always have a high level of trust since it must monitor the trust metrics of other nodes in the network. Hence, the weight α must always be higher than the other parameters ($\alpha \geq \beta + \lambda + \omega$). If the IIoT network is very dense, in addition to the high Tm ($\alpha \geq \beta + \lambda + \omega$), the CL_j must also prove a high computing capacity CPm and sufficient monitoring energy Er to perform monitoring tasks

($\beta+\lambda \geq \omega$). Otherwise, if the IIoT network is less dense with isolated nodes, the CL_j must be chosen based on its Tm ($\alpha \geq \beta+\lambda+\omega$) and also based on its centrality ($\omega \geq \beta+\lambda$) to avoid being placed in the extremities of the network and building empty communities.

The IIoT server checks all received DI, only nodes with a DI greater than the acceptance threshold (AT) take the authorization to announce themselves as a CL_j nodes. The AT is set by the monitoring process, it reflects the type of devices deployed in the network; indeed, if the network contains sensitive and heterogeneous devices, the value of AT must be set at a high value in order to impose more selectivity in the designation and the choice of CL_j nodes. Once approved by the server, the designated CL_j broadcasts beacons to all other nodes within its transmission range to announce itself as a leader.

As explaining in Algorithm 1, when a CL_j receives a beacon from another $CL_{j'}$, it first checks that the DI($CL_{j'}$) sent by the $CL_{j'}$ in the beacons is above the AT. If the opposite is true, CL_j rejects the beacon and reports to the IIoT server that $CL_{j'}$ performs an impersonation attack. Otherwise, if DI($CL_{j'}$) is greater than AT and the physical distance between the two nodes is less than the threshold D, hence, the leader with the lowest DI must be eliminated to optimize the number of CLs and the number of communities formed especially if the network is not dense. The value of D is chosen according to the density of the network; IIoT server chooses a small value of D if the IIoT network is dense in order to increase the number of CLs. If the network is dense and the value D chosen is large, it will cause congestion at the CL nodes and a quick drainage of their batteries. In case of the network is not dense, the threshold D must be large to optimize the number of CLs in the network. Choosing a small D while the network is not dense will constitute communities with very few member nodes or even empty. If the IIoT server notices that no CL_j from a set of nodes located in the same zone has been chosen (since none of this set has a DI exceeding the threshold AT), the IIoT server will designate as CL_j the node that has the largest DI among this set.

Algorithm 1 The designation algorithm

CL_j is a designated leader;
$CL_{j'}$ is a new designated or an already designated leader; where $j \neq j'$
When CL_j receives a beacon from node $CL_{j'}$;
begin
1. **if** DI($CL_{j'}$) $<$ AT **then**
2. **RejectBeacon**;
3. **S**end the feedback to the server; **GOTO** (End)
4. **else**
5. **if** ($d(CL_j, CL_{j'}) < D$ **then**
6. **if**(DI(CL_j)>DI($CL_{j'}$))
7. Status(CL_j)=CL;
8. Status($CL_{j'}$)=MN;
9. Send the feedback to the server;
10. $CL_{j'}$ stops brodcast beacons to other nodes and CL_j continues to send them;
11. **else if**(DI(CL_j)<DI($CL_{j'}$)) **then**
12. Status($CL_{j'}$)=CL;
13. Status(CL_j)=MN;
14. Send the feedback to the server;
15. CL_j stops brodcast beacons to other nodes and $CL_{j'}$ continues to send them;
16. **else if**(DI(CL_j)=DI($CL_{j'}$)) **then**
17. The leader with the highest Tm will continue its leadership functions and the other will become MN;
18. **End if**
19. **else if** ($d(CL_j, CL_{j'}) > D$) **then**
20. Status($CL_{j'}$)=CL;
21. Status(CL_j)=CL;
22. Send the feedback to the server;
23. The two nodes continue to send beacons to their communities respectively;
24. **End if**
25. **End if**
End

B. Phase II: The formation of communities

Each MN_i will choose to join community of one and only one CL_j and each CL_j will monitor one and only one community. The number of non-empty and feasible communities CK is 2^K-1. In order to form communities around the designated CL_j, we propose a community formation algorithm.

MN_i computes the selection indicator SI_{CL_j} for each node CL_j from which it has received beacons and it chooses the CL_j having the maximum SI_{CL_j}. SI_{CL_j} is calculated based on the physical distance, the quality of link and the industrial relationships between the two nodes as follows:

$$\begin{aligned} SI_{CL_j} =& \delta(1-\left(\frac{d(MN_i,CL_j)}{max_{CL_j}(d(MN_i,CL_j))}\right)) \\ &+\kappa(1-BER_{CL_j}) \\ &+\eta(RI_{CL_j}) \end{aligned} \tag{4}$$

Where $\delta+\kappa+\eta=1$ and $0 \leq \delta,\kappa,\eta \leq 1$, $d(MN_i, CL_j)$ is the physical distance between MN_i and CL_j nodes. When a node receives a signal from its adjacent node, it uses the power of this received signal to calculate the distance between them. It is assumed that IIoT nodes already have an appropriate identity management system. MN_i has an interest to choose a leader close to it in order to save its energy and to avoid retransmissions and data losses. SI_{CL_j} is unitless, its value ranges from 0 to 1. $max_{CL_j}(d(MN_i, CL_j))$ is the maximum physical distance that can be between MN_i and CL_j nodes, it is used to normalize the SI_{CL_j}.

The bit error rate (BER) is the number of bit errors divided by the total number of transferred bits during a studied time interval. BER is a unitless, it is defined as following:

$$BER_{CL_j} = \left(\frac{The\ number\ of\ bit\ errors}{The\ total\ number\ of\ transferred\ bits}\right) \tag{5}$$

The relationship indicator RI_{CL_j} evaluates the industrial proximity of MN_i with CL_j based on the industrial relationships between the two nodes. It is defined as follows:

$$\begin{aligned} RI_{CL_j} =& \Gamma(PR(MN_i,CL_j)) + \Theta(CWR(MN_i,CL_j)) \\ &+ \rho(OR(MN_i,CL_j)) + \mu(SR(MN_i,CL_j)) \end{aligned} \tag{6}$$

Where $\Gamma+\Theta+\rho+\mu=1$ and $0 \leq \Gamma,\Theta,\rho,\mu \leq 1$, each node sets its own weights conforming to its profile. Once a node is designated as a leader, it starts sending beacons to nodes in its transmission range to announce its presence. MN_i can receive periodic beacons from one or more CL_j nodes. Beacons contain the constructor code, the owner code, the

activity list and the list of the social friends of CL_j node that sends them. These data are already predefined in the profile of each node. They are sent hashed to MN_i nodes, because they contain technological and industrial data of the CL_j that sends them. MN_i uses these data to evaluate its industrial relationships with CL_j as a follows:

- Parental relationship $PR(MN_i, CL_j)$: In the heterogeneous IIoT network, there are devices that require specific abilities from their monitoring nodes to accurately monitor their trust. A small sensor can be monitored by any other device proving at least the same technological characteristics as it, however, a connected production robot cannot be monitored by a simple sensor. The sensor does not have sufficient technical resources to accurately evaluate the trust metric of a connected production robot. MN_i will privilege parental relationship in order to have approximately the same technological capabilities as its leader. To evaluate this relationship, MN_i compares its constructor code with the constructor code sent hashed in the beacon of CL_j. If the two codes are similar, the $PR(MN_i, CL_j)$ value takes 1 otherwise it takes 0.

- Ownership relation $OR(MN_i, CL_j)$**:** During trust management, CL_j asks the MN_j nodes for a set of information about their operations. In automotive plants, some IIoT nodes contain critical and production-sensitive data. Critical MN_i node chooses its leader based on this relationship to restrict the circulation of its monitoring data just between the nodes belonging to the same production line or user as it. To evaluate this relationship, MN_i compares its owner code with the owner code sent hashed in the beacon of CL_j. If the two codes are similar, the $OR(MN_i, CL_j)$ value takes 1 otherwise it takes 0.

- Co-worker relationship $CWR(MN_i, CL_j)$**:** This relationship makes the calculation of trust more accurate because the MN_i node will be evaluated by a leader that performs the same activities to it. To evaluate this relationship, MN_i calculates the similarity between its activity list $A(MN_i)$ and the list $A(CL_j)$ that contains the set of hashed activities of node CL_j as follows:

$$CWR(MN_i, CL_j) = \left(\frac{A(MN_i) \cap A(CL_j)}{A(MN_i) \cup A(CL_i)} \right) \quad (7)$$

- Social relationship $SR(MN_i, CL_j)$**:** is chosen when the MN_i nodes are based on the social relations of their users (hierarchy relations, collaboration relations...) to choose their leaders. To evaluate this relationship, MN_i calculates the similarity between the list of its social friends $S(MN_i)$ and the list $S(CL_j)$ that contains the set of hashed social friends of the CL_j node as follows:

$$SR(MN_i, CL_j) = \left(\frac{S(MN_i) \cap S(CL_j)}{S(MN_i) \cup S(CL_j)} \right) \quad (8)$$

To form communities between CL_j and MN_i nodes, the community formation algorithm is proposed as follows:

Each MN_i has a preference list where it stores the identities of its favorite leaders. When MN_i receives new beacons from $CL_{j'}$ node other than its leader, it calculates the selection indicator $SI_{CL_{j'}}$ of $CL_{j'}$ and it stores its identity in the preference list. CL_j nodes are ranked in the list in descending order of their selection indicators. The leader with the largest selection indicator will be placed at the top of the preference list. If the list is full, the MN_i node compares the calculated $SI_{CL_{j'}}$ with the selection indicator of the last node in the preference list, if it is greater, MN_i removes the last leader from the list and adds the $CL_{j'}$ which will be ranked in the list according to its selection indicator value. MN_i can join the community of leaders stored in its preference list in case it will no longer receive beacons from its own leader.

Algorithm 2 The community formation algorithm

Input: CL_j and MN_i nodes;
Output: Industrial Communities;
Begin
1. **FOR** each $MN_i | i\ in 1, 2, , L$
2. Fixe the coeficient δ, κ, Γ, Θ, ρ, μ;
3. **FOR** each $CL_j | j\ in 1, 2, , K$
4. Compute $d(MN_i, CL_j)$, BER_{CL_j}, $PR(MN_i, CL_j)$, $CWR(MN_i, CL_j)$, $OR(MN_i, CL_j)$, $SR(MN_i, CL_j)$;
5. **END FOR**
6. **END FOR**
7. **FOR** each $CL_j | j\ in 1, 2, , K$
8. Compute the initial community $C(CL_j)$={ $MN_i | i\ in 1, 2, , L$; $d(MN_i, CL_j) \leq transmission\ range$ };
9. **END FOR**
10. **FOR** each $MN_i | i\ in 1, 2, , L$
11. **FOR** each $CL_j | j\ in 1, 2, , K$
12. **if** $(CL_j | j\ in 1, 2, , K)$;
13. Compute $SI(MN_i, CL_j)$
14. **END If**
15. **END FOR**
16. **FOR** each $CL_j | j\ in 1, 2, , K$;
17. Compute the optimal community $C(CL_j)$
18. **END FOR**
18. **END FOR**
19. Return the optimal community $C(CL_j)$;
End

C. Phase III: The proposed trust model analysis

In this phase, we propose the Tm-IIoT model suitable for the H-IIoT architecture defined in the previous phases I and II. The IIoT server sends to each CL_j all necessary data to evaluate the Tm of its members. CL_j is based on these data to calculate three performance metrics called cooperation, direct honesty and indirect honesty, needed to update the trust metrics. Each node integrates the network by having an initial trust metric $T0$. The value of Tm and $T0$ varies within the interval [0, 1]. The Tm of a monitored node will increase, decrease, remain unchanged or go down to zero [36] as shown in the state transition diagram in Fig.2. The transitions depend on the three trust performance metrics and the current state of the monitored MN_i node.

To formalize the transitions of Tm, we use a state transition diagram at $E + 1$ states as shown in Fig.2. Each state corresponds to a specific value of Tm, state 0 corresponds to a non-trust level and state E corresponds to a high level of trust. We divide the interval[0, 1] of the Tm into $E+1$ states, each one represents a step ϕ where (1 mod ϕ= 0)) [36].
The transition matrix corresponding to the state transition diagram of our proposed approach is expressed as follows:

$$P = (P_{i,j}(t))_{0 \leq i,j \leq E} \quad (9)$$

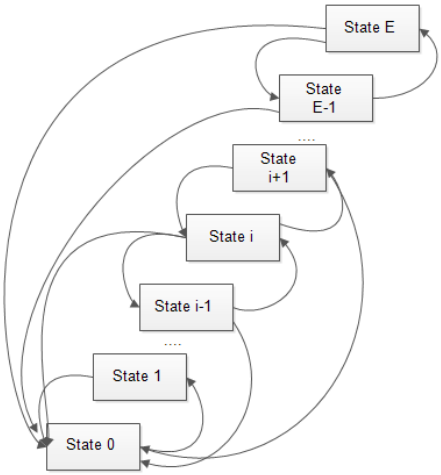


Fig. 2: State transition diagram

Where $P_{i,j}$ is the probability to transit from state i to state j, it is expressed as follows:

$$P_{i,j}(t) = Pr(Y_t = j \mid Y_{t-1} = i) \tag{10}$$

The random variable Y_t is the current Tm of an IIoT device at time t.
From the transition matrix expressed in equation (9), given that the initial state of a node is Y_0=1 it is possible to calculate the probability to be in a state i at time t as follows:

$$p_i(t) = \sum_{zin[1...E]} P_{1,z}(t_z) * P_{z,i}(t) \tag{11}$$

In order to evaluate the different transition probabilities allowing to update the Tms of member nodes, CL uses the information received from the IIoT server to calculate three performance metrics describing the behavior of the monitored MN_i nodes over time.
- The cooperation rate: evaluates the behavior of the monitored MN_i related to its cooperations in the network. The cooperation rate C_{MN_i} of MN_i is the number of the successful forwarded messages divided by the total number NF of the messages transmitted to it by its CL_j. The cooperation rate of the monitored node MN_i is calculated as follows:

$$C_{MN_i} = \left(\frac{\sum_{i=1}^{NF}(C_{mi})}{NF} \right) \tag{12}$$

Where C_{mi} has a value of 1 if the message mi is forwarded correctly or 0 if the message mi is not forwarded. The probability pc that the MN_i cooperates in the network is expressed as follows:

$$pc = CMN_i * pnet \tag{13}$$

Where $pnet$ is the probability that reflects the constraints related to the nature of wireless networks: congestions, retransmission, obstacles and quality of links between the sender and the receiver. It is assumed that each environment has a well-known probability $pnet$ calculated from the number of retransmissions and the number of erroneous bits. The purpose of introducing this probability is to take into account the constraints of the monitoring environment in the final calculation of trust.
- The direct honesty rate: measures the compatibility between the activities performed by the MN_i node in the network and the activities that MN_i must perform and predefined in its profile. The node that monitors a production machine in order to send reports every hour to the production system is considered as a dishonest node when it does this task every four hours, hence, it must be isolated from the network in order to define the causes of this behavior. The direct honesty rate allows to detect this type of malicious activities and behavior changes produced by MN_i nodes.

The IIoT server sends to CL_j node, in a list $A(MN_i)$, the set of hashed activities, that MN_i node must perform in the nework. The hashes contained in the list $A(MN_i)$ are stored in the IIoT server and in the CL_j node. The activities are sent hashed to CL_j nodes in order to prevent data leakage due to probable interception of monitoring messages by malicious nodes and in order to ensure that in the event of CL_j intrusion, the activities will not be disclosed. In order to evaluate the direct honesty rate D_{MN_i} of a node MN_i, CL_j compares the set of hashes contained in the two lists $A(MN_i)$ and $AF(MN_i)$ by calculating their similarity ratio as follows:

$$D_{MN_i} = \left(\frac{A(MN_i) \cap AF(MN_i)}{A(MN_i) \cup AF(MN_i)} \right) \tag{14}$$

pdh is the probability that the monitored node MN_i is honest, it is calculated as follows:

$$pdh = D_{MN_i} * pnet \tag{15}$$

The list $AF(MN_i)$ contains the set of hashed feedback activities of node MN_i. The hashes are generated by the MN_i node and sent to the CL_j node in order to perform comparisons. After the comparisons, the set of hashes contained in the list $AF(MN_i)$ will be deleted automatically by the CL_j node to optimize the memory space.

Malicious activities are defined in our work as behavior changes and non-cooperation of IIoT nodes in the network. In the SIoT concept [10], nodes have an infinite number of activities to perform because they are related to the unlikely activities of the human being, but in a factory the number of activities defined for each IIoT node is finite, each node has a finite number of tasks to perform previously defined by the IIoT server. Hence, when an IIoT node changes behavior, its leader easily detects the changes by calculating its direct honesty rate. Behavior changes of IIoT nodes are dangerous, simple disruptions can disclose the company's trade secret or cause plant failure and major material damage as the Stuxnet worm attack [9] on Iran's nuclear installations showed in 2010.
- The indirect honesty rate: reflects the reputation of the monitored MN_i inside its community. The nodes in the community that already have experiences with MN_i must give it a score according to its behavior in the network. The nodes that will give extreme scores (too big or too small) than the other member nodes will be sounded as malicious nodes that aim to perform ballot stuffing attacks or bad-mouthing attacks. CL_j contacts its member nodes to give their opinions on the MN_i node. Afterward, it removes spams by using the

estimation algorithm [30] and it sends to the IIoT server the list of nodes that generate spams to sanction them. CL_j calculates the indirect honesty rate of node MN_i as an average of the scores given by the community members while deleting spams as follows:

$$I_{MN_i} = \left(\frac{\sum_{MN_{i'}=1}^{N} R_{MN_{i'},MN_i}}{N} \right) \quad (16)$$

Where $RMN_{i'}, MN_i$ is the reputation given by the member node $MN_{i'}$ to the node MN_i, $MN_i \neq MN_{i'}$. MN_i does not give a score for itself to avoid self-promotion attacks. N is the total number of the member nodes after eliminating the spam nodes. The probability pih that the monitored MN_i node has a good reputation is calculated as follows:

$$pih = I_{MN_i} * pnet \quad (17)$$

To obtain the transition matrix P in equation (9), CL_j calculates as in our previous work [30] the increasing and decreasing state probabilities ($P_{i,i+1}(t)$ and $P_{i,i-1}(t)$), the state stay probability $P_{i,i}(t)$, the probability to sojourn in the trusted state E $P_{E,E}(t)$ and the probability to transit to state 0 $P_{i,0}(t)$. According to the state transition diagram in Fig.2, knowing the trust state of a MN_i node in time $t-1$, at time t this state will have only five choices: state+1, state-1, remain in the same state, reach the trusted state E or the non-trusted state 0. The probabilities that the Tm is in these states at time t are each calculated using equation (11). At time t, Tm of a monitored MN_i takes the trust value corresponding to the state where the probability calculated by equation (11) is the max. Each state corresponds to a specific value of Tm, state 0 corresponds to $Tm = 0$ and state E corresponds to $Tm = 1$.

V. Performance evaluation

In order to prove the robustness, the adaptiveness and the reliability of our proposed Tm-IIoT model, we used the InstantContiki 2.7 platform [37]. The various simulation parameters are listed in table I. We run the simulation experiments within an automotive plant of 200 heterogeneous devices (sensors, robots, connected machines...) that are able to instate industrial relationships between them based on their profile previously defined by the IIoT server. We used the TMote Sky motes (Cooja simulator).

Simulation tool	contiki/cooja 2.7
Mote type	Tmote Sky
Nodes Distribution	Random
Simulation run time Tr	24h
Total number of node	200
Deployement environment	automotive sector
Network protocol	IP based
Rooting protocol	RPL
Transmission ranges Trg	[50,100]
Interference ranges Irg	[50,100]
Number of states E	10
ϕ	0.1

The performance evaluation of this work is done in two main parts. In the first part, we compare the energy efficiency of the H-IIoT architecture with the energy efficiency of the traditional architecture of the plant's IIoT network. In the second part, we evaluate the ability, the adaptiveness and the resiliency of our proposed trust model against TMCoi-SIOT model [25], Adaptive IoT Trust model [18] and CITM-IoT model [24].

1) Part 1: Comparative energy study : In this part, the topology of the plant's IIoT network is represented according to two architectures: traditional architecture in Fig.3 and H-IIoT architecture in Fig.4. The H-IIoT architecture is the result of the application of equation (3), equation (4), algorithm 1 and algorithm 2 to the traditional architecture represented in Fig.3.

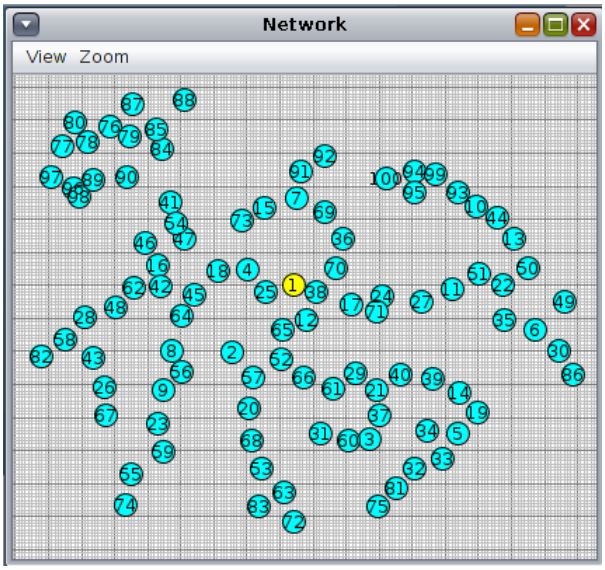


Fig. 3: Topology of traditional centralized IIoT architecture

IIoT nodes contribute to the trust management by sending monitoring packets to their trust entities. The monitoring packets contain cooperation and transaction histories, they also contain recommendations for other nodes in the network. We apply the Tm-IIoT model on both architectures, we evaluate for each one, in a time period Tr, the average energy consumption of IIoT nodes during their contributions to trust management. The results of this evaluation are shown in Fig.5 and Fig.6. For each architecture, we have chosen to represent just the average power of nodes with the highest and the lowest energy consumption.

According to histograms in Fig.5 and Fig.6, the central processing unit (CPU) of nodes belonging to the same architecture consumes on average the same amount of energy because all of these nodes generate the same type and quantity of

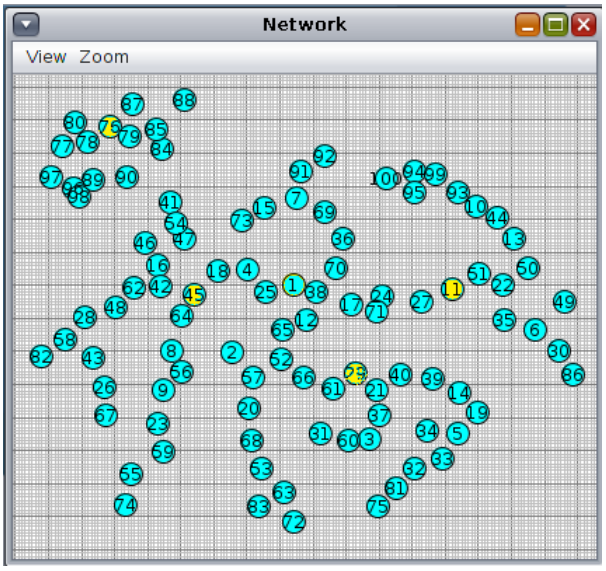


Fig. 4: Topology of H-IIoT architecture

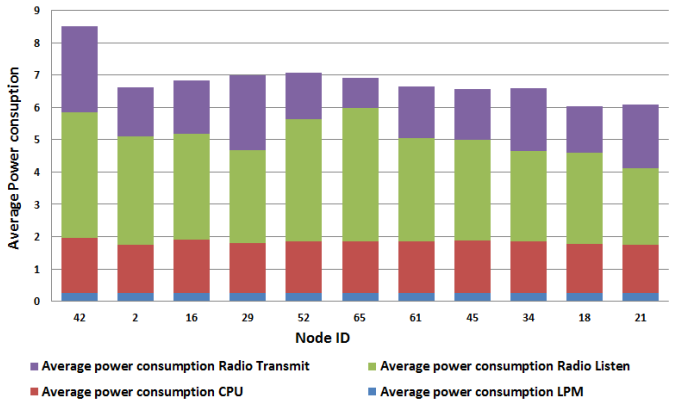


Fig. 5: Average power consumption in traditional IIoT architecture of 200 nodes, in a time period Tr

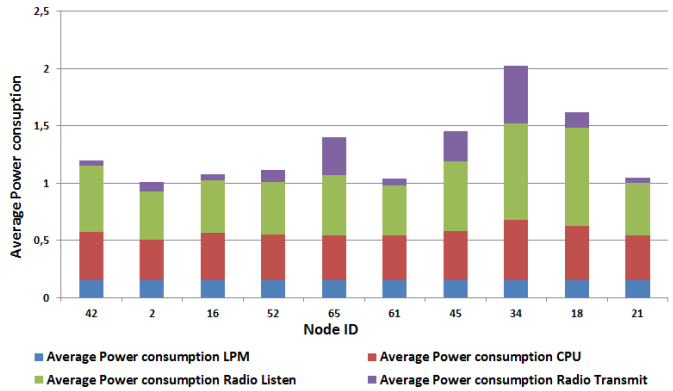


Fig. 6: Average power consumption in H-IIoT architecture of 200 nodes, in a time period Tr

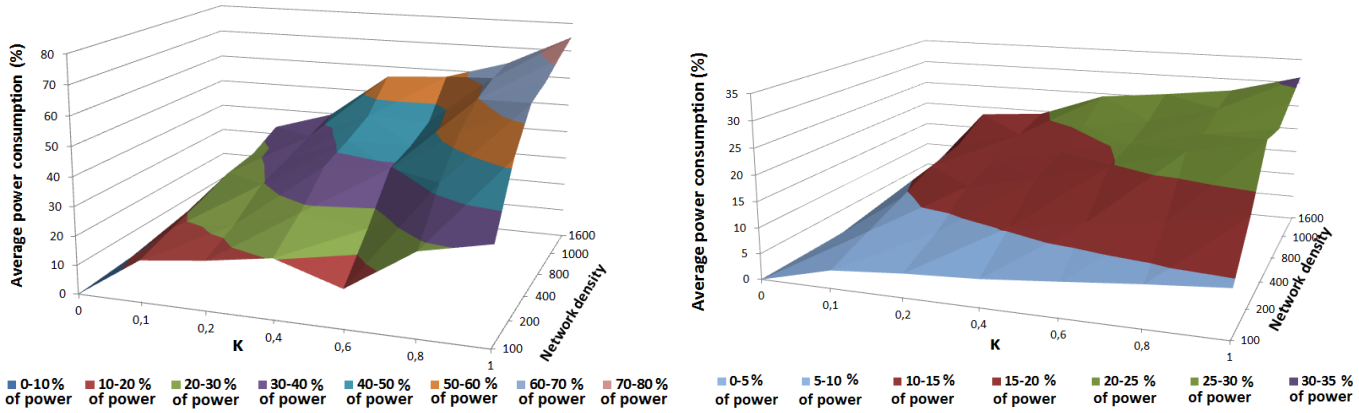
monitoring data. The average of energy consumed by CPU of nodes in Fig.5 is a little more significant than the average consumed by CPU of nodes in Fig.6, this is explained by the fact that nodes in traditional architecture generate more recommendations. They must give recommendations for all other nodes in the network which is impossible to achieve because nodes do not have a total knowledge of all network, they can give ratings just for nodes with which they have already had experiences. While in H-IIoT architecture, each MN_i provides recommendations just for the members of its community.

According to Fig.5 and Fig.6, we notice what the overall energy consumption of IIoT nodes during their contributions to monitoring is impacted by their radio operations. Indeed, IIoT nodes consume on average a lower energy in H-IIoT architecture compared to the traditional centralized architecture. In the traditional architecture, IIoT nodes consume more radio resources in order to relay monitoring packets from the entire IIoT network. Whereas in our proposed architecture, nodes relay just the monitoring packets of the other nodes in their clusters. For instance, node 42 in Fig.5, consumes on average 8.149 mW (listen radio Power: 3.894 mW, radio transmission Power: 2.658mW) compared to Fig.6 where it consumes 1.2 mW (listen radio Power: 0.578 mW, radio transmission Power: 0.05 mW). These obtained simulation results are estimated for 200 IIoT devices and for an accuracy $\kappa = 0.5$. Over time, with the evolution of the network and the need to have more accuracy κ, these results will increase exponentially as shown in Fig.7.

The obtained results in Fig.7 estimate the average power consumption of IIoT nodes according to the evolution of the network (from 100 nodes to 1600 nodes) and according to the level of accuracy that we want to achieve in the evaluation of trust. As shown in Fig.7, the average power consumption increases with the increasing accuracy. The consumption increases even more when the number of nodes in the network increases, especially in traditional centralized architecture Fig.7.a where energy consumption is more significant. As shown in Fig.7.a, normal nodes can consume on average until 75% of their power just in order to relay monitoring packets, while at the base, nodes must use this power for their own tasks. We justify these results by the fact that the more the network evolves and the accuracy increases, the more the generation of monitoring packets increases and the more radio operations become more important and consume more energy. In Fig.7.a and Fig.7.b, with a network of 1600 nodes and a maximum accuracy $\kappa = 1$, the average power consumption of MN_i nodes in H-IIoT architecture reaches 30% compared to 75% in traditional architecture, this is due to the fact that in H-IIoT architecture, the network is managed in communities so each Mn_i relays the monitoring packets of its community and not of the whole network. In the H-IIoT architecture, the number of CLs is proportional to the density of the nodes in the network.

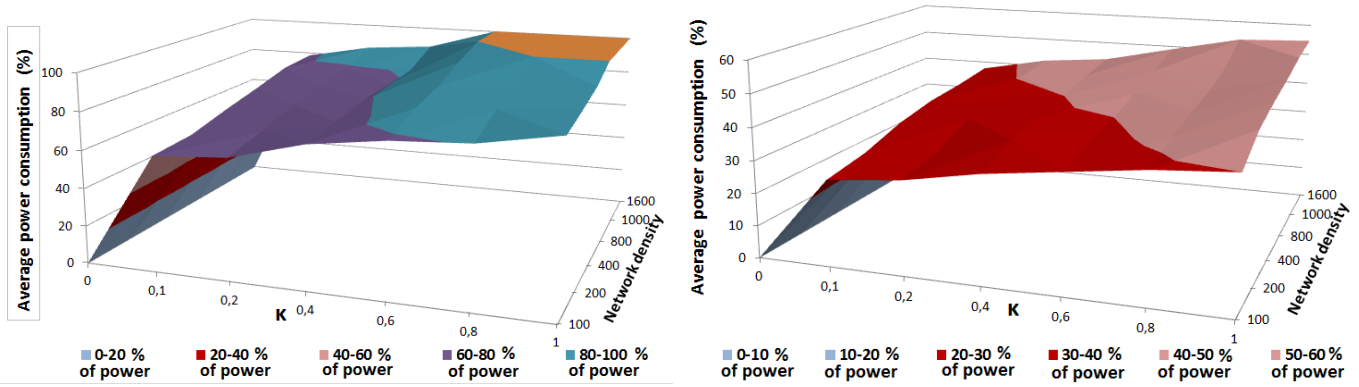
To manage the Tms of 1600 IIoT devices with an accuracy $\kappa = 1$, CL nodes consume on average until 54.60% of their energy reserved for monitoring as shown in Fig.8.b. While for the traditional architecture as seen in Fig.8.a and for the same conditions, the trust management entity consumes on average until 100% of its energy reserved for monitoring.

According to Fig.9, the average number of lost monitoring packets is more significant when the accuracy and the density



(a) Centralized architecture (b) H-IIoT architecture

Fig. 7: Average power consumption of nodes with the evolution of the network population and the increase of κ



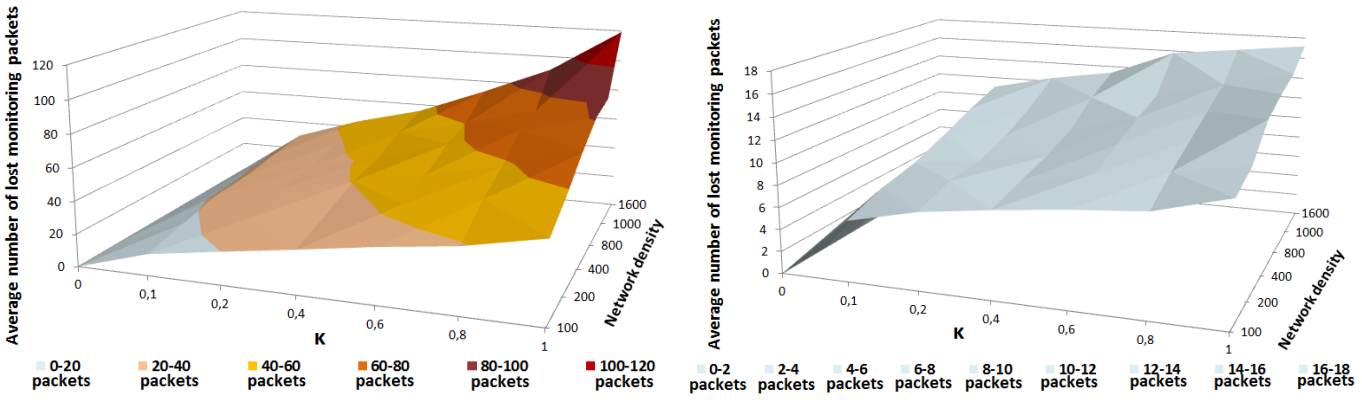
(a) Centralized architecture (b) H-IIoT architecture (consumption of CL node)

Fig. 8: Average power consumption of trust management entities, with the evolution of the network population and the increase of κ

of the IIoT network increase. The losses are higher in the centralized architecture Fig.9.a compared to the H-IIoT architecture in Fig.9.b. The lost packets contain monitoring data that will allow the trust entity (CLs in our architecture) to manage the Tms, each lost packet introduces errors in the accuracy of the Tms evaluation and causes retransmissions as well as energy losses. All obtained results in this part justify the need to change the traditional centralized architecture in industrial automobile plants to the H-IIoT architecture. In order to justify the need for a new architecture in the presence of DLTs, we compare our proposed H-IIoT architecture with the trust architecture based on a blockchain proposed in [12]. As shown in Fig.10, we notice that nodes in this architecture consume more energy than nodes in our architecture Fig.7.b. In [12], each node calculates the Tm of its neighbors and stores them in a database serving as a ledger of transactions, whereas in our architecture only CL_j nodes manage the trust in the network and the other nodes perform their normal tasks. If we use blockchain technology to manage trust in the plant's IIoT network, IIoT nodes will consume energy unnecessarily because they don't need to know the Tm of their neighbors. Only the IIoT server must have the global Tm of each IIoT node in the network to isolate malicious nodes.

2) Part 2: The ability, the adaptiveness and the resiliency of our proposed trust model against existing trust management schemes : In this part, we analyze the trust convergence properties of an IIoT node MN_i belonging to a community of 40 nodes in an environments containing varying percentage of malicious nodes Pm ranging from 20% to 80%. We use the simulation parameters listed in table I. We observe in Fig.11 that the convergence time and the trust bias increase with the increase of malicious nodes. However, even with a malicious node population of 50%, the trust bias remains $\leq$15%, this proves the resilience of the proposed Tm-IIoT model in the most unfavorable environments.

We demonstrate the effectiveness of our proposed Tm-IIoT model with a comparative performance analysis against TMCoi-SIOT model [25], Adaptive IoT Trust model [18] and CITM-IoT model [24]. The TMCoi-SIOT model is applied directly to the industrial context without modification, but the Adaptive IoT Trust model and the CITM-IoT model require some modifications and adaptations; we have replaced social



(a) Centralized architecture

(b) H-IIoT architecture

Fig. 9: Average number of lost monitoring packets, with the evolution of the network population and the increase of κ

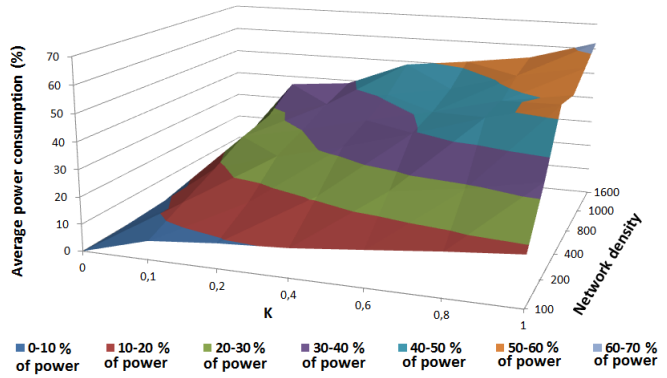


Fig. 10: Average power consumption of nodes in the architecture proposed in [12], with the evolution of the network population and the increase of κ

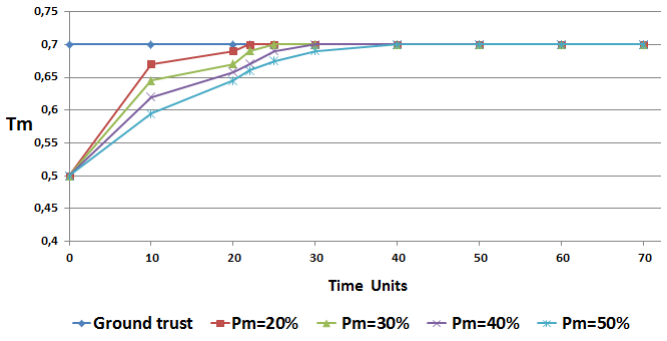


Fig. 11: Tm of a Good node with Pm ranging from 20% to 50%

relations with industrial relations.

As shown in Fig.12, with a malicious node population of 20%, the convergence time in CITM-IoT model is faster compared to that of our model. This is due to the non-complexity of the calculations performed in the CITM-IoT model; Tm of a cluster node is the average of the evaluations collected from peer cluster nodes after removing outliers. We notice also that in CITM-IoT model, the Tm never reaches the exact value 0.7, this is justified by the lack of trust management context, the authors do not specify on which basis the nodes manage the peers of their cluster. The trust bias in CITM-IoT model increases brutally ($\geq 35\%$) even more when the number of malicious nodes reaches 50% as shown in Fig.13, this is due to the lack of context and also to the vulnerability of the model to coalition attacks. The spam algorithm used in this model is unable to detect coalition attacks when the number of malicious nodes increases. It makes its decision based on the evaluation given by the majority of cluster nodes. It has as a majority the evaluations given by malicious cluster nodes and it eliminates the evaluations of good and fair nodes. When the number of malicious nodes reaches 50%, the Adaptive IoT Trust model reaches convergence at 47 time units as shown in Fig.13, this is due to the distributed type of the model. In order to have the final Tm of a monitored node, we have to do an aggregation of the all local Tms given by its neighbors. When the number of malicious nodes reaches 50%, the trust bias increases brutally ($\geq 30\%$) because there is no spam value extraction before calculation of the final Tm value. The trust bias increases even more ($\geq 40\%$) for TMCoi-SIOT model specially when the population of malicious nodes increases as shown in Fig.13. TMCoi-SIOT model does not detect spam recommendations, it takes them into account in its trust calculation.

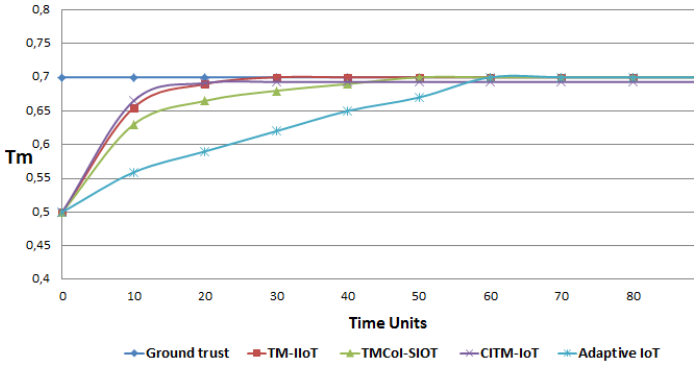


Fig. 12: Tm of a good node, Pm=20%

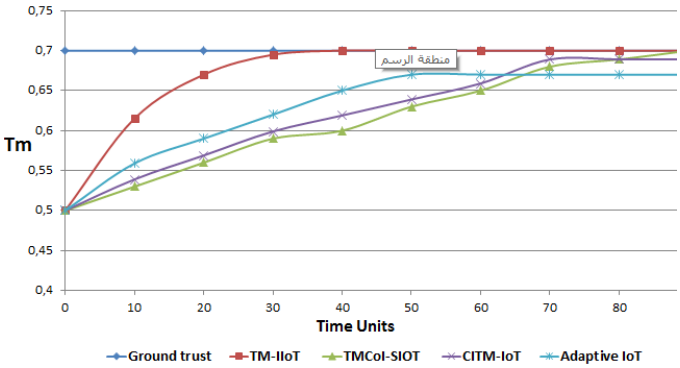


Fig. 13: Tm of a good node, Pm=50%

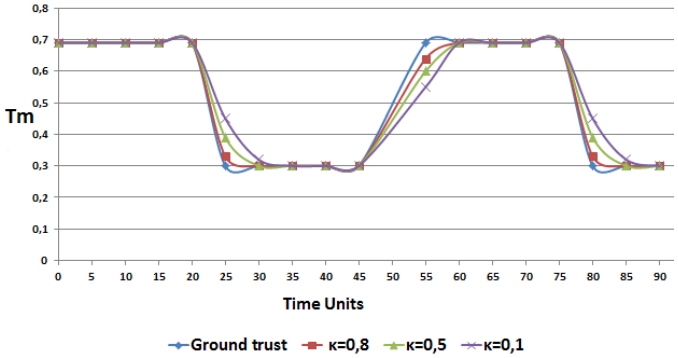


Fig. 16: Behavior changes of a critical node, $\kappa = 0.8$, $\kappa = 0.5$, $\kappa = 0.1$

In order to prove the resiliency of our proposed Tm-IIoT model against the dynamic behavior of IIoT nodes, we evaluate the behavior change detection of two nodes: a normal node containing non-sensitive data in Fig.14 and a critical node containing more sensitive data in Fig.15. In fact, during the first 20 time units, the two nodes behave positively in the network to increase their Tm and they change their positive behavior to a malicious behavior following an attack during 20 time units followed by a positive behavior during 20 time units and so forth as shown in Fig.14 and Fig.15.

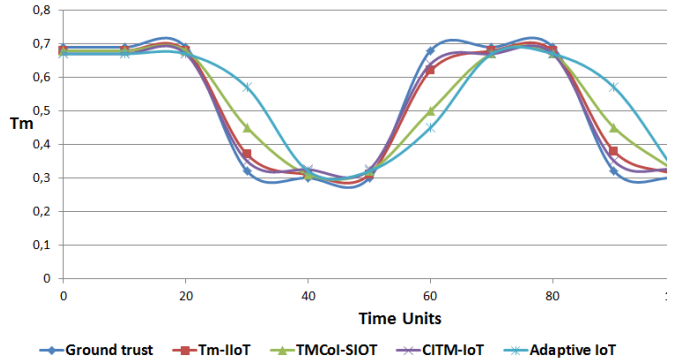


Fig. 14: Behavior changes of a non-critical node, Pm=30%

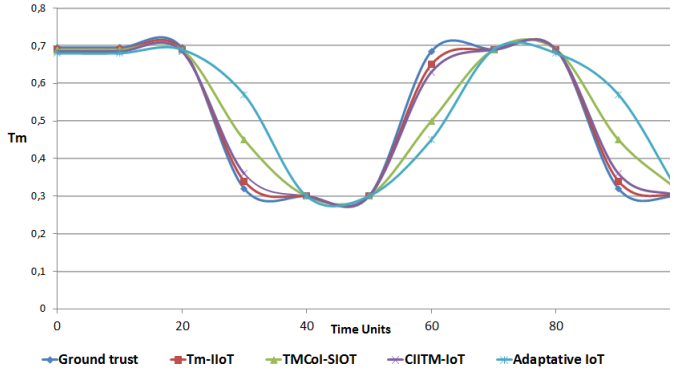


Fig. 15: Behavior changes of a critical node, Pm=30%

According to the obtained results, for the less critical node in Fig.14, Tm-IIoT model is more sensitive to behavior changes, it accurately detects node behavior perturbations at the 25th time units, compared to the TMCoi-SIOT model and the Adaptive IoT Trust model [18] which take more than 29 time units and 34 time units respectively to detect the disruptions.

Our model takes less time (22 time units) to detect behavior changes of the sensitive node in Fig.15, unlike other models that take the same average time as for the less critical node. They do not differentiate between sensitivity levels of IIoT nodes, unlike our model.

When a node returns to its initial positive behavior, the Tm-IIoT model quickly detects this behavior recovery and the Tm of this node returns to its previous value after 2 time units if it contains critical data else 5 time units if it contains less critical data.

TMCoi-SIOT model, Adaptive IoT Trust model and CITM-IoT model are less sensitive to behavior changes, they take a longer time to recover the previous Tm value of a node that resumes its initial good behavior. Regardless of the sensitivity of nodes, the recovery of the Tm takes at least 10.5 time units for TMCoi-SIOT model, 20 time units for Adaptive IoT Trust model and 10 time units for CITM-IoT model. This experiment proves the adaptiveness and the resiliency of the Tm-IIoT model. These results are obtained for $\kappa = 0.8$. As shown in Fig.16, with the variation of κ, the behavior change detection will not maintain the same accuracy. The Tm-IIoT model detects behavior changes of critical nodes after 6 time units when $\kappa = 0.1$ and after 4 time units when $\kappa = 0.5$ compared to 2 time units when $\kappa = 0.8$. Despite this variation, our model shows a high resiliency against behavioral changes compared to other models which regardless of the sensitivity of nodes they keep the same treatment.

Coalition attacks: occur when a group of malicious nodes:

- Mobilize against a good node in order to reduce its Tm by sending it a bad recommendations. In this case, the coalition attack is called bad-mouthing attack [38].
- Increase the Tm of a malicious node. In this case the coalition attack is called a ballot-stuffing attack [38].

To evaluate the resiliency of our model against coalition attacks, we propose the following two scenarios: In Fig.17, we assume a bad-mouthing attack against a good behavior node MN_1 with Tm=0.8 and in Fig.18, we assume a ballot stuffing attack performed to increase the Tm of a malicious node MN_2 with Tm=0.4. Each of MN_1 and MN_2 belongs to a community of 40 IIoT members. The evaluation is conducted in an environment that initially contains 20% of malicious nodes and then 50% to properly evaluate our model in the most unfavorable environments.

It is assumed for both scenarios that malicious nodes perform coalition attacks at the 21th time unit. We notice that for a 20% of malicious nodes, the Tm in the Tm-IIoT model varies briefly until reaches the maximum value of 0.72 in Fig.17 and 0.46 in Fig.18. The Tm disruptions in the two scenarios will not have serious consequences on the safety and the survivability of MN_1 and MN_2 because the Tms vary slightly and quickly return to their previous values at the 45th time unit.

The convergence time is longer for our model when the percentage of malicious nodes reaches 50%, this is due to the estimation algorithm described in [30], it needs more time to detect spams. The estimation algorithm must be reinforced by the direct honesty rate that compares the current behavior of IIoT nodes with their profiles described in IIoT server.

In comparison with TMCoi-SIOT model and Adaptive IoT Trust model, we clearly notice that the Tm strictly diverges and never recovers its previous value in both models. This is due to the fact that these models do not use a mechanism to detect spam recommendations from malicious nodes, they take them into account when calculating and updating the Tms. In our model, the trust metric quickly returns to its old value by using the estimation algorithm described in [30]. Fig.17.b and 18.b show that the CITM-IoT model is very vulnerable to coalition attacks when the percentage of malicious nodes reaches 50%. The proposed algorithm to eliminate outliers in this work is based on the evaluation given by the majority of cluster nodes. It has as a majority the evaluations given by malicious cluster nodes and it eliminates the evaluations given by good nodes.

VI. Conclusion

In the first part of this paper, the purpose was to change the traditional architecture of the automotive plants into a new hierarchical architecture called H-IIoT in order to manage the trust of the IIoT devices. The H-IIoT architecture will be based on a new concept called industrial relationship between IIoT devices. In the second part of this paper, we proposed the Tm-IIoT model to monitor trust metric of IIoT nodes in the proposed H-IIoT architecture. The simulation results have proven the energy efficiency of our H-IIoT architecture compared to the traditional IIoT network architecture of automotive plants. They have also proven the ability, the adaptiveness and the resiliency of the Tm-IIoT model to detect behavior changes. In this contribution we focused on trust management to protect IIoT devices against malicious attacks but we ignored some security aspect as cryptography and access control. This is foreseen for the future works.

References

[1] E. Hofmann and M. Rusch, Industry 4.0 and the current status as well as future prospects on logistics. In: Computers in Industry, vol. 89, pp. 23-34, 2017.

[2] G. J. Cheng, L. T. Liu, X. J. Qiang, Y. Liu, Industry 4.0 Development and Application of Intelligent Manufacturing. In: 2016 International Conference on Information System and Artificial Intelligence (ISAI), pp. 407-410, 2016.

[3] S. R. C. K. M., M. P., Industry 4.0 Challenges and Solutions for the digital transformation and use of exponential technologies. Deloitte AG, pp. 32, 2015

[4] Soumya Kanti Datta, Christian Bonnet, Securing IoT Platforms. In: Consumer Electronics (ICCE) 2019 IEEE International Conference on, pp. 1-2, 2019.

[5] F. D. Hudson, Enabling trust and Security - TIPPSS for IoT. In: IT Professional, vol. 20, pp. 15-18, Mar 2018.

[6] Arman Pouraghily, Md Nazmul Islam, Sandip Kundu, Tilman Wolf, Poster Abstract: Privacy in Blockchain-Enabled IoT Devices. In: 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), pp: 292-293.

[7] Chao Li, Balaji Palanisamy, Privacy in Internet of Things: From Principles to Technologies. In: IEEE Internet of Things Journal. Volume: 6, Issue: 1, Feb. 2019, pp: 488-505.

[8] Abbas M. Hassan, Ali Ismail Awad, Urban Transition in the Era of the Internet of Things: Social Implications and Privacy Challenges. In: IEEE Access Year: 2018, Volume: 6, pp: 36428-36440.

[9] D.P. Fidler, "Was stuxnet an act of war? decoding a cyberattack", IEEE Secur. Privacy, vol. 9, no. 4, pp. 56-59, 2011.

[10] L. Atzori, A. Iera, G. Morabito, and M. Nitti, The social Internet of Things (SIoT)-When social networks meet the Internet of Things: Concept, architecture and network characterization. In: Comput. Netw., vol. 56, no. 16, pp. 3594-3608, 2012.

[11] F. Bao, I.R. Chen, Dynamic trust management for internet of things applications. In: Proceedings of the 2012 international workshop on Self-aware internet of things, Self-IoT '12, USA, San jose, 2012, pp.1-6

[12] Asma Lahbib et al., Blockchain based trust management mechanism for IoT, 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019.

[13] A. Moinet, B. Darties, J. L. Baril, Blockchain based trust and authentication for decentralized sensor networks, 2017

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

[15] 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.

[16] 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.

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

[18] I.R. Chen, J. Guo, and F. Bao, Trust Management for SOA-based IoT and Its Application to Service Composition. In: IEEE Transactions on Service Computing, vol. 9, no. 3, 2016, pp. 482-495.

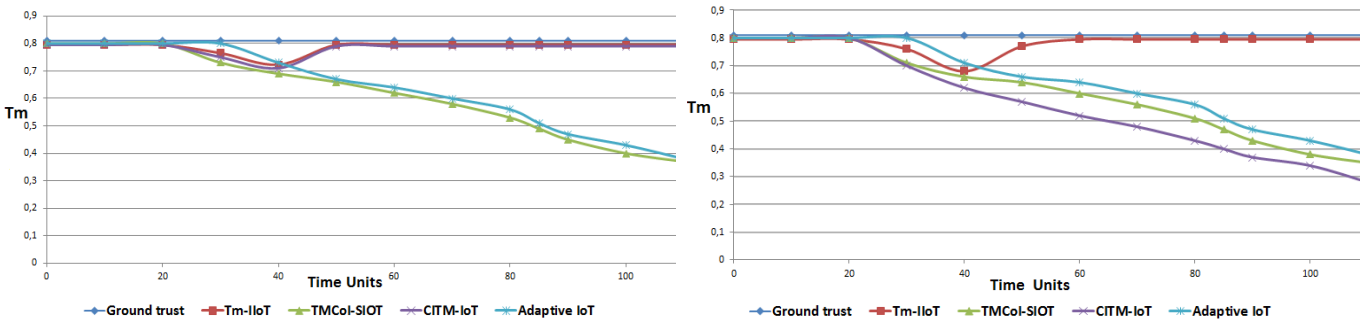
[19] J. Guo, R. Chen, and J. J. Tsai, A survey of trust computation models for service management in the internet of things systems. In: Computer Communications, vol. 97, pp. 1-14, 2017.

[20] M. Nitti, R. Girau, and L. Atzori, Trustworthiness management in the social Internet of Things. In: IEEE Trans. Knowl. Data Eng., vol. 26, no. 5, pp. 1253-1266, May 2014.

[21] H. Al-Hamadi, I. R. Chen, Trust-based decision making for health IoT systems. In: IEEE Internet Things J., vol. 4, no. 5, pp. 1408-1419, Oct. 2017.

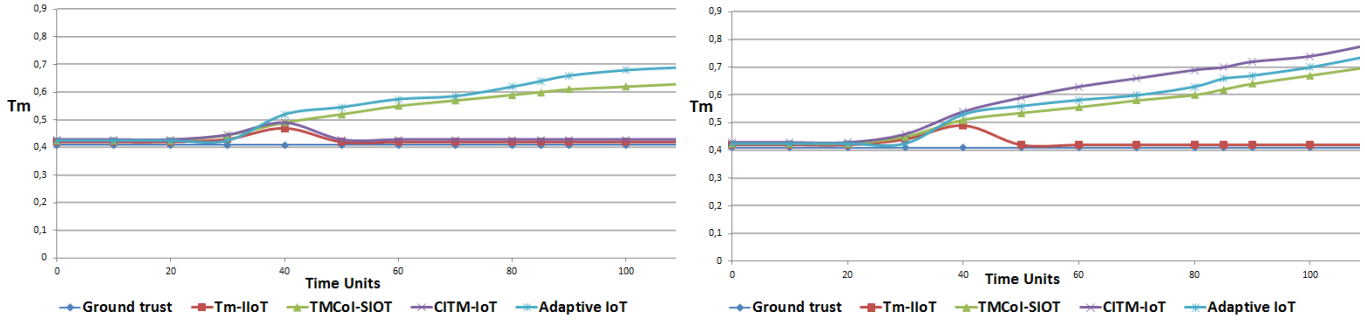
[22] Antonio L. Maia Neto, Yuri L. Pereira, Artur L. F. Souza, Italo Cunha, Leonardo B. Oliveira, Demo Abstract: Attributed-Based Authentication and Access Control for IoT Home Devices. In: 2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp: 112-113.

[23] 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.In: Comput. Security, vol. 39, pp. 351-365, Nov. 2013.



(a) Pm=20% (b) Pm=50%

Fig. 17: Coalition attack: Bad-mouthing attack



(a) Pm=20% (b) Pm=50%

Fig. 18: Coalition attacks: Ballot-stuffing attack

[24] M. D. Alshehri, F. K. Hussain, and O. K. Hussain, Clustering-driven intelligent trust management methodology for the Internet of Things (CITM-IoT), Mobile Netw. Appl., vol. 23, no. 3, pp. 419431, 2018.

[25] O. B. Abderrahim, M. H. Elhdhili, and L. Saidane, TMCoi-SIOT: A trust management system based on communities of interest for the social Internet of Things. In Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), 2017, pp. 747-752.

[26] W. Abdelghani, C.A. Zayani, I. Amous, F. Sdes: Trust Management in Social Internet of Things: A Survey. In: Proc.15th IFIP WG 6.11 Conference on e-Business e-Services and e-Society (I3E 16) 2016, vol. 9844, pp. 430-441.

[27] Xiao, H., Sidhu, N., Christianson, B., Guarantor and reputation based trust model for social Internet of Things. In: International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, pp. 600-605 (2015).

[28] F. Bao, Ing-Ray Chen, Guo, J., Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems. In: Proceedings of the IEEE eleventh International Symposium on Autonomous Decentralized Systems (ISADS); 2013. pp. 1-7.

[29] M. Nitti, R. Girau, L. Atzori, A. Iera, G. Morabit, A Subjective Model for Trustworthiness Evaluation in the Social Internet of Things. In: Proceedings of the IEEE 23rd international symposium on Personal Indoor and Mobile Radio Communications (PIMRC); 2012. pp. 18-23

[30] Chaimae Boudagdigue, Abderrahim Benslimane, Abdellatif Kobbane, Mouna Elmachkour, A Distributed Advanced Analytical Trust Model for IoT. In: ICC 2018, pp: 1-6

[31] Jose-Vicente de Los Mozos, Industry 4.0, production plants shaped by the future, (available at https://group.renault.com/en/innovation-2/industry-4-0-production-plants-shaped-by-the-future/).

[32] Groupe psa, Industry 4.0 and automotive excellence by groupe psa, (available at https://www.groupe-psa.com/en/automotive-group/industrial-performance/).

[33] Volkswagen, Industry 4.0: We make it happen! (2019), (available at https://www.volkswagen-newsroom.com/en/stories/industry-40-we-make-it-happen-4779).

[34] Mercedes-benz, Industry 4.0: Digitalisation at Mercedes-Benz, (available at https://www.mercedes-benz.com/en/mercedes-benz/innovation/industry-4-0-digitalisation-at-mercedes-benz-video/).

[35] A. Karati, S. H. Islam, M. Karuppiah, Provably secure and lightweight certificateless signature scheme for IIoT environments. IEEE Trans. Ind. Inform., vol. 14, no. 8, pp. 3701-3711, Aug. 2018.

[36] T. Gazdar, A. Rachedi, A. Benslimane, A. Belghith, A distributed advanced analytical trust model for VANETs. In: IEEE. IEEE GLOBECOM'2012, Dec 2012, Anaheim, California, United States. IEEE Press, pp.219-224, 2012.

[37] Thingsquare. (2016, June, 2016). Contiki: The Open Source OS for the Internet of Things, Available: http://www.contiki-os.org/download.html

[38] W. Abdelghani, C.A. Zayani, I. Amous, F. Sdes, Trust management in social internet of things: a survey. In: Conference on eBusiness e-Services and e-Society. pp. 430-441, 2016, September.