TrustLib, a Lightweight Trust Protocol for IoT

***Abstract*—The expansion of the Internet of Things leads to revolutionizing applications in many areas but also brings with it a huge potential of cybersecurity threats. Ensuring trust becomes a major challenge to provide security and privacy of the ecosystem. The majority of studies propose trust models or blockchain-based trust models, designed for the fog level but not suitable for IoT devices with limited computing power, memory, and energy. This paper presents a new lightweight multi-layer protocol to ensure trust both at edge and fog levels, by providing interaction assessments. As interactions occur, each device is able to estimate the trust level of its counterparts from practical metrics. As a result, it can identify and favor interactions with the most trustworthy entities, to improve the quality of service and energy consumption. Also, the protocol is designed as a software library in order to be easily embedded in devices.**

Keywords—Trust, Blockchain, Internet of Things, Decentralized network, Fog computing

I. Introduction

The Internet of Things (IoT) is a global infrastructure, aiming to connect any system of various kinds, to generate new services. It could be a simple RFID tag, a sensor/actuator, or even a smartphone, which can interact with its environment and with its counterparts. While this heterogeneity and this hyperconnectivity offer many opportunities in many areas (*e.g.* health, industry, energy, etc.), IoT deployment represents major challenges in terms of interoperability, data storage, scalability, energy, quality of service but also regarding the security and privacy. Facing the emergence of cyber-attacks, an IoT ecosystem has to integrate security mechanisms to deal with various threats. Besides, its network topology is becoming conducive to decentralization for efficiency reasons. Without a central organization, how can the reliability and security of a highly heterogeneous environment be ensured? More generally, there is a need to ensure trust in this ecosystem.

Ensuring security, without compromising performances of IoT devices is a huge challenge because of their limited computing power, energy, and memory. During last years, Cloud systems have been intensively used to overcome these limitations but the increase in the number of connected devices is such that centralized architectures like Cloud, become unsuitable regarding the real-time and the energy efficiency features of the IoT. Besides, Cloud represents a single point of failure in the security and privacy point of view. In response, the emergence of the so-called "Fog Computing" brings great improvements regarding efficiency and scalability. In this new ecosystem, each entity cooperates thanks to a decentralized Machine-to-Machine (M2M) communication protocol, without a central part. However, the removal of the latter requires a new security mechanism to ensure the trust between machines at any level from devices to the cloud through gateways.

The notion of trust has been widely studied [1]–[11]. Generally, trust is a high-level concept, regrouping behavior and security. In the IoT context, devices do not know each other, thus a system has to be in charge to ensure the reliability and the performance of every device, to detect and isolate unreliable/malicious ones. The trust mechanism aims both to continuously improve the efficiency and security of the ecosystem, from multiple observations. The main challenges are the selection of the best metrics to evaluate the trust and a relevant aggregation method to compute them into a single value, as a global trust view. Also, the information used for this assessment has to be fair and reliable.

Recently, the blockchain system has been well studied by the community, since it provides tamper-proof data storage, avoids a single point of failure, does not rely on a third party, and sets up a validation process done by all participants in the network. Today, blockchain is mainly used in cryptocurrency applications such as Bitcoin [12] and Ethereum [13] but because of its high resource utilization (*i.e.* computing power, storage space, and energy), many efforts have to be done to adapt it to the IoT [14]–[17]. Indeed, blockchain is more suitable to secure the network at fog level, between gateways, but there is still a lack of a solution to secure interaction at the edge level, between IoT devices. Also, even if blockchain ensures fair, transparent, distributed, and reliable information storage, it does not evaluate the trustworthiness of entities. This emphasizes the strength of the association of trust models and blockchains. According to these issues, this work aims to propose a new lightweight and multi-layer protocol, called TrustLib, to improve the quality of service and energy consumption, by ensuring the trust both at the edge level (*i.e.* device-to-device and device-to-gateway interactions), and the fog level (*i.e.* gateway-to-gateway interactions).

The remainder of this paper is organized as follows: Section II is dedicated to related works; then we provide some preliminaries in the third part. Section IV presents the proposed TrustLib protocol and its primary results are described in the next section.

II. Related Works

A. Concept of Trust

As explained in [3], the concept of trust is a complex notion, influenced by several measurable and non-measurable properties. However, it can be standardized as the expectation that a device expectedly accomplishes a task. To correctly handle this concept, it is important to emphasize some characteristics. Trust is dynamic and context-dependent, which means it is true only in a given period of time and in a given context. Also, trust is not transitive, meaning that if device X trusts device Y which trusts device Z, X does not necessarily trust Z. This also underlines the non-mutual relationship and its subjectivity. Indeed, X can trust Y but not reciprocally and X can be trusted by some of them but not by others. Finally, trust depends on history, meaning that past experiences may influence the current trust level.

Three sources of information are required to evaluate the trust: knowledge (*i.e.* own understanding about a device), experience (*i.e.* interactions with a device), and reputation (*i.e.* experience from other devices). While the knowledge is based on the inherent characteristics of the device (*i.e.* information given by manufacturers) as mentioned in [3], experience refers to the evaluation based on its own interactions with devices [5]–[9] ("Direct Trust"). Opposite, reputation refers to the trust evaluation of entities, reported by others ("Indirect Trust").

B. Compute the Trust

By cumulating all previous information about an entity in a given period of time, its trust score can be computed to help a device to make a decision regarding this entity. This operation requires a so-called "Trust Model". Authors in [11] emphasize the most used such as weighted sum, Bayesian models, and fuzzy-based techniques. But the interesting part of a model is its properties used in the trust computation. The social relationship approach in the IoT context [6] (*e.g.* Honesty, Cooperativeness, and Community Interest) remain complex to evaluate, leading to a lack of efficiency and reliability. Therefore, some research focus on more quantifiable metrics as packet size [7], forwarding delay [7], energy [9], quality of wireless signal [9], or even the participation rate of a device (number of its tasks done).

C. Trust & Blockchain

One of the main limitations of the traditional trust management system is the tamper-proof trust storage of information. As previously discussed, the trust evaluation is based on information about connected devices over a period of time, but also on its history (*i.e.* previous experiences). To ensure a reliable assessment, all information needs to be verified, authenticated, and not modified. In other words, data collection has to be secured. This is especially true given the decentralized topology of an IoT network without a central authority, where the risk of data manipulation is amplified. This is the main motivation to combine trust models and blockchain. Several surveys as [14]–[17] explain the concept of blockchain and its integration for the IoT context in detail. In a few words, blockchain is a distributed ledger, composed of blocks containing transactions, and replicated among a network of peers. Blocks are cryptographically linked directly with the previous one, ensuring their immutability. The majority of papers [1], [11], [18]–[20] use blockchain to store the trust level of each device. Authors in [21] save in blockchain information related to the trust computation such as devices' properties, capabilities, and also feedback from others. Finally, [22] proposes a blockchain-based trust management, by storing all authentication information and trust score of each entity.

The association of blockchain and trust models is a very promising system since it provides tamper-proof storage and good dissemination of trust data over the network. However, despite many interesting contributions, the reviewed studies are based on private or existing blockchains (*e.g.* Ethereum), which are not suitable for the IoT context (*i.e.* scalability, energy, storage). Few works provide reward/punishment of devices to incentive good behavior. Also, proposed approaches are focused at the fog level (*i.e.* gateway). Consequently, there is still a lack of a solution to ensure the trust directly between IoT devices but also with gateways.

III. Preliminaries

As mentioned previously, there are two main types of trust: direct and indirect. According to [23], it is impossible to reliably have a complete image of trust with only direct interactions. Moreover, [8] explains that indirect trust is an important way to reduce the risk of malicious attacks. Nevertheless, it is important to analyze the impact of the indirect trust mechanism on performance and security. Table I. summarizes the main threats to trust models. It is interesting to note that majority of attacks are oriented on indirect trust, instead of direct. Indeed, while the latter depends only on the own view of a device, indirect trust requires an evaluation from others. This leads to potential manipulations and increases the risk of bad evaluations. Devices have to verify the authenticity and the integrity of the recommendations, but also to deal with the current trust level of the recommenders. Consequently, indirect trust requires intensive communication between devices and increases the computation and energy costs, leading to a less scalable, less

TABLE I. ATTACKS ON TRUST MODELS

Attacks	Descriptions	Trust
Self-Promoting	Malicious device promotes itself to be seen as honest but delivers malfunctioning data and services	Indirect
Whitewashing	Malicious device disconnects and reconnects to clear its bad trust score	Direct Indirect
Discriminatory	Tempt to fool new device which has not yet enough trust evaluation of others	Indirect
Bad-Mouthing	Tempt to ruin the reputation of a well-behaved device, with bad recommendations	Indirect
Ballot-Stuffing	Tempt to increase the trust score of a bad device by sending good recommendations	Indirect
Opportunistic Service Attack	Malicious device provides good service to increase its trust score, when the latter decreasing because of bad service	Direct Indirect

performant, and even less secure protocol. The trade-off between reliable overall trust view and security is not necessarily significant. Due to the low capacity of devices, the protocol at the edge level should be as simple as possible. For this reason, quantifiable metrics are preferred and only the direct trust evaluation is performed, to avoid the complexity induced by the indirect one.

IV. TrustLib

The incredible plurality of applications offered by the IoT makes it impossible the implementation of a universal protocol. Nevertheless, this work strives to be adaptable to a wide range of applications. This section describes TrustLib, a lightweight trust protocol for both edge and fog levels, to improve the quality of service and energy consumption of an IoT ecosystem. The protocol is designed to be easily implemented and takes the form of a software library to include into the object's source code (*i.e.* sensor/actuator node and gateway).

A. *TrustLib at Edge Level*

A sensor node scans and digitalizes its physical environment and acts as data generator. Regarding a sensor, the trust view relies on the efficiency of other entities in processing its data. For example, in the case of sensitive data transmission, the sensor node has to be sure that the recipient has well received the data. This can be done by the acknowledge procedure. The TrustLib protocol can use the latter to evaluate the communication quality through the success rate, the signal strength (*e.g.* RSSI), and the latency. Thanks to this assessment, the sensor node can favor the transmission of its information to entities which has the best network stability, *i.e.* the level of trust.

In contrast, an actuator uses the data it receives for its decision making. Its reliability depends on the information received (*e.g.* accuracy, integrity, authenticity, etc.). It will give more credibility to the information coming from entities with a high level of trust, thus improving its own reliability. The same applies to the quality of the received signal, which may require data retransmissions and latency. In addition, since the actuator physically acts on its environment, operations must be triggered safely. The use of a trusted protocol is an interesting means of filtering false information, to prevent malicious acts and improve the efficiency of the actuator.

As a proof of concept, this section proposes several formulas to evaluate the trust of an entity seen by a sensor/actuator node, from the following metrics: the operation success (O_S), the latency (L), and signal quality (S_Q) both normalized between 0 and 1. Also, since the power source of a sensor node and actuator is often limited, its own battery level (*Bat*) is taken into account. These elements are then aggregated as the trust assessment. Equations (1) to (4) detail the calculations.

$$O_S = \begin{cases} -1 \text{ when operation fails} \\ 1 \text{ when operation succeeds} \end{cases} \tag{1}$$

$$L = \frac{L_i - L_{MIN}}{L_{MAX} - L_{MIN}} \qquad L \in [0\ ;\ 1] \tag{2}$$

$$S_Q = \frac{S_i - S_{QMIN}}{S_{QMAX} - S_{QMIN}} \qquad S_Q \in [0\ ;\ 1] \tag{3}$$

$$AG_{SA} = O_S \times \left(\frac{1}{\left((1+L)+(2-S_Q)\right)^{O_S}} \right) \times \frac{1}{Bat} \tag{4}$$

For a sensor node, the success of an operation (O_S) entirely depends on the confirmation signal sent by the recipient. If the latter does not acknowledge receipt, the message sent by the node is considered lost (O_S = -1). In the case of an actuator, the success of an operation depends on the ability of the actuator to perform the requested operation without error. The latency (L_i) is measured between the transmission of the message and the reception of the acknowledge (for the sensor node) or between two incoming transmissions during the process of the actuator. The value is then normalized between 0 (best case) and 1 (worst case) according to the minimal and maximal application latency requirements (L_{MIN} and L_{MAX} respectively). The same applies to the signal strength (S_i) according to RSSI minimal and maximal values (S_{QMIN} and S_{QMAX}).

The proposed sensor/actuator node aggregation (AG_{SA}) aims to compute a trust value from the previous metrics. The operation success (O_S) determines the sign of this value. Indeed, when the operation fails (O_S = -1), the reliability of the solicited entity decreases. The central part of the aggregation is used to compute a network score based on latency and quality of the signal. To simulate a social characteristic of trust, namely that "it is harder to gain it than to lose it", this score is raised to the power of O_S (-1 or 1). This means that if the operation fails, the aggregation value will be tenfold, thus strongly penalizing the trust value. Finally, the battery level of the sensor/actuator, the one doing the evaluation, is used to calibrate the sensitivity of the aggregation: when its energy level decreases, the impact of each interaction on the trust is amplified.

B. *Trust at Fog Level*

Gateways embed more resources such as computing power and memory, than sensor and actuator nodes, and interact in many ways. In a local network perspective (gateway-to-object), gateways can perform continuous analysis of the performances of each sensor and actuator. Since the transmissions of the latter are mainly dedicated to monitoring during the operation in progress, it is possible to extract their performances. Regarding sensors, gateways can assess their information and set up efficient and reliable services. By extending this approach on each gateway, the quality of service/experience both on the local and global network can be improved. For example, by promoting the propagation of the most relevant data, or by taking into account the supply and demand of data/resources/services, to diversify the ecosystem. Also, the trust can be monetized, in the sense that an entity with a high trust level can access more data, services, and resources.

This first gateway's trust model is voluntary simple in order to validate the feasibility of such a system. The evaluation of the trust level from the point of view of the gateways is again based on the normalized signal quality (3).

A reliability score, R_{Score}, is also taken into account, reflecting the relevance of the data from the sensors, the performance of the actuators, or the evaluation of the interaction with another gateway. For example, R_{Score} could represent the ratio of Shannon's entropy of an entity over the latency to get the information. In other words, this is an image of the amount of information over the quality of service. Its normalized value is defined by (5). A second score is assigned to indicate the usefulness of the interactions for the ecosystem enrichment, as a quality of experience, noted U_{Score}, and defined by (6). Finally, the calculation of the gateway aggregation AG_G is presented by (7).

$$R_{Score} = \frac{R_i - R_{MIN}}{R_{MAX} - R_{MIN}} \qquad R_{Score} \in [0\,;\,1] \tag{5}$$

$$U_{Score} = \left\{ \begin{array}{l} -1 \text{ when useless interaction} \\ \ \ 1 \text{ when useful interaction} \end{array} \right. \tag{6}$$

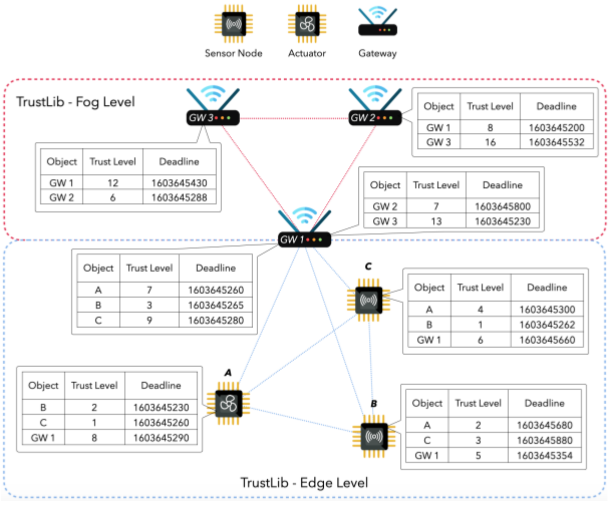
$$AG_G = U_{Score} \times \left(\frac{1}{\left((2-R_{Score})+(2-S_Q)\right)^{U_{Score}}} \right) \tag{7}$$

According to the definition of IoT, the objective is to interconnect everything, to exchange various types of data. A first U_{Score} attribution model is envisaged: following the principles of IoT, the sharing of data on the network is defined as a useful interaction, favoring the enrichment of the ecosystem (U_{Score} = 1). Any other interaction (*e.g.* reconfiguration request, data access request, etc.) is considered as not useful in terms of added value on the network (U_{Score} = -1). The interest of such a model is particularly interesting to promote the dissemination of information on the network while discouraging non-essential exchanges. Through the satisfaction score (R_{Score}), the gateway can analyze and automatically reconfigure the actuator parameters. For example, by temporarily adjusting the frequency of actuator-gateway transmissions, it is possible to obtain a better follow-up of the operations, which, after analysis, can lead to improved performance. Also, the satisfaction score can be used to identify the gateways providing the best services (*e.g.* relevance, latency, availability, etc.).

C. *Structure*

The TrustLib protocol is designed to be embedded in connected objects and must take into account their limited resources. Since the protocol focuses on the environment close to an object, the number of local entities connected over a specific period of time remains low compared to the global or inter-gateway network. Therefore, it is possible to set up a dynamic list of entities with their associated trust level, as shown in Fig. 1. The size of the list is configurable, depending on the storage capacity of the object. Because of their wider connectivity and higher memory capacity, gateways can embed a longer list than sensor/actuator nodes.

As defined in section II.A, trust depends on past and present interactions. Consequently, the trust level that one entity grants to another at a given moment is represented by (8). Furthermore, trust is dynamic and ephemeral. In order to include this aspect in TrustLib, each assigned level has a validity lifetime, at the end of which it decreases. This lifetime, as well as the reduction coefficient, are configurable according to the application.



Object	Trust Level	Deadline
GW 1	8	1603645200
GW 3	16	1603645532

Object	Trust Level	Deadline
GW 1	12	1603645430
GW 2	6	1603645288

Object	Trust Level	Deadline
GW 2	7	1603645800
GW 3	13	1603645230

Object	Trust Level	Deadline
A	7	1603645260
B	3	1603645265
C	9	1603645280

Object	Trust Level	Deadline
A	4	1603645300
B	1	1603645262
GW 1	6	1603645660

Object	Trust Level	Deadline
B	2	1603645230
C	1	1603645260
GW 1	8	1603645290

Object	Trust Level	Deadline
A	2	1603645680
C	3	1603645880
GW 1	5	1603645354

Fig. 1. TrustLib Structure

$$TrustLevel_i = TrustLevel_{i-1} + AG \tag{8}$$

D. *TrustLib Management*

It is important to note that trust is nominative, which means it is attributed to a specific entity. A mechanism ensuring the authenticity of interactions is therefore necessary. This can be integrated at the network and/or application level (*e.g.* using the *Network Session Key* of the LoRaWAN protocol or using public/private key for the application). Algorithm 1. represents the entire TrustLib protocol embedded in sensor/actuator node and gateway.

```
Algorithm 1 : Trust Management
Input: $Interaction_{Entity_A}$
Output: $Updated\ Trust\ List$
$O_S, L, S_Q, R_{Score}, U_{Score}, AG_{SA/G} = \text{Evaluation}(Interaction_{Entity_A})$
for each entity in TrustList do
    if $entity == Entity_A$ then
        if $Entity_A.Deadline \leq Current_{Date}$ then
            $Entity_A.TrustLevel_n = Entity_A.TrustLevel_{n-1} + AG_{SA/G} - Penalty$
        else
            $Entity_A.TrustLevel_n = Entity_A.TrustLevel_{n-1} + AG_{SA/G}$
        end
        $Entity_A.Deadline = NewDeadline$
    else
        if $entity.Deadline \leq Current_{Date}$ then
            $entity.TrustLevel_n = entity.TrustLevel_{n-1} - Penalty$
            $entity.Deadline = NewDeadline$
        end
    end
    if $entity.TrustLevel_n \leq 0$ then
        $Remove(entity)$
    end
end
if $Entity_A$ not in the list then
    if $AG_{SA/G} \leq 0$ and TrustList not full then
        $Entity_A.TrustLevel_n = AG_{SA/G}$
        $Entity_A.Deadline = NewDeadline$
        $Add(Entity_A)$
    end
end
```

Algorithm 1. TrustLib Protocol Embedded in Sensor/Actuator Nodes and Gateways

The first phase consists of evaluating the interactions with an entity, using the previous equations. An examination of the list is then carried out, to recover the previous trust level of this entity and to update it. During the parsing of the list, each entity that has passed the deadline is subject to a penalty, and a new deadline is defined to perpetuate the procedure. This approach ensures the ephemeral property of the trust. If the entity to evaluate is not in the list, the trust level will correspond to the value of the aggregation. To avoid unnecessary use of storage space, only trust levels greater than 0 are kept. This opens the opportunity for new entities to be included in the list, by deleting those with a null trust level. Consequently, there is no negative trust level. This choice is justified by the method of identification of the entities, based on cryptographic keys. This means that simply changing the key would be sufficient to override a negative trust level and restart at the default level of 0.

E. Primary Security Analysis of TrustLib

As a reminder, TrustLib relies solely on direct trust assessments, making it immune to any attack exploiting bad or laudatory reputations. As shown in Table I. the “Opportunistic Service Attack” and “Whitewashing” are the most relevant to TrustLib. The former, which consists of providing reliable data/services only when its trust level is too low, can be mitigated by the severity of the protocol when bad interactions occur. This forces the malicious entity to satisfy its ecosystem more regularly, otherwise, it will be isolated from the system. This provides an incentive for each member of the network to participate in the enrichment of its environment, and not to be a mere user.

The second attack, “Whitewashing”, consists of disconnecting and reconnecting to the network, to erase a bad level of trust. Another approach is to create a new identity. The TrustLib protocol is constructed so that as long as an object is not registered in the list, its trust level is 0, the lowest it can be (no negative value as seen previously). As a result, such an attack shows no advantage. In addition, the approach based on the generation of new identities requires computing resources, and may even penalize the attacker himself.

V. Simulations

The purpose of this section is to illustrate the impact of each metric on the evolution of the trust level, following the model of each entity. Fig. 2 shows the results for the sensor/actuator node and gateway. As expected, it is more difficult to gain trust than to lose it. No less than 200 interactions under optimal conditions (*i.e.* operation success, latency, signal quality, and satisfaction score), are necessary between objects to reach the maximum trust level. Opposite, between 50 and 25 interactions are sufficient to lose trust, depending on the metrics. The loss of trust is the fastest when the sensor/actuator node has a low battery level. After only 3 unsuccessful interactions, with minimum signal quality and maximum latency, the trust that a node attributes to an entity is reduced to 0%. This underlines the promising approach of the TrustLib protocol to improve the quality of service and the energy consumption of an IoT ecosystem.

Besides the trust model, it is important to quantify the storage requirements of TrustLib. As a reminder, the trust list is composed of the device ID (*e.g.* cryptographic key), the trust level, and the deadline. For evaluation purposes, the NIST *secp256k1* elliptic curve cryptography, as used in Bitcoin is chosen. The public key is defined on 33 bytes. The trust level can be stored in decimal form (*float*, on 4 bytes) to maintain accuracy, or on a single byte to minimize storage utilization. The latter configuration offers a range of 256 different levels, which may be sufficient for most applications. Finally, the date is stored as a 32-bit (4 bytes) timestamp. Therefore, up to 41 bytes are required for each stored entity in the list. Fig. 3. shows the required storage space according to the length of the list. At the edge level, with very limited objects, less than 400 bytes allows to evaluate about ten entities. With more storage capacity, about 2 KB is needed to store nearly 50 entities. Since the protocol focuses on a proximity environment, this may be sufficient for many applications. Since blockchain is a trust mechanism and can be used at fog level (*i.e.* on gateways), it is interesting to compare the current platforms such as Bitcoin, Ethereum, Nano [24], and IOTA [25] with TrustLib. Results in Fig. 3 show that the proposed protocol divides the storage utilization by more than 10^4 compared to Bitcoin, and by a 12-factor compared to the smallest blockchain Nano. Gateways, such as a Raspberry board, can easily embed hundreds of GB and manage a list of more than 20 million entities with only 1 GB.

Curves	Metric Configurations	
	Sensor/Actuator Node	Gateway
🟩	$(O_S = 1\ ; L = 0; S_Q = 1; Bat = 1)$	$(U_{Score} = 1\ ; R_{Score} = 1; S_Q = 1)$
🟨	$(O_S = -1\ ; L = 0; S_Q = 1; Bat = 1)$	$(U_{Score} = -1\ ; R_{Score} = 1; S_Q = 1)$
🟦	$(O_S = -1\ ; L = 1; S_Q = 1; Bat = 1)$	$(U_{Score} = -1\ ; R_{Score} = 0; S_Q = 1)$
🟪	$(O_S = -1\ ; L = 1; S_Q = 0; Bat = 1)$	$(U_{Score} = -1\ ; R_{Score} = 0; S_Q = 0)$
🟥	$(O_S = -1\ ; L = 1; S_Q = 0; Bat = 0{,}1)$	-

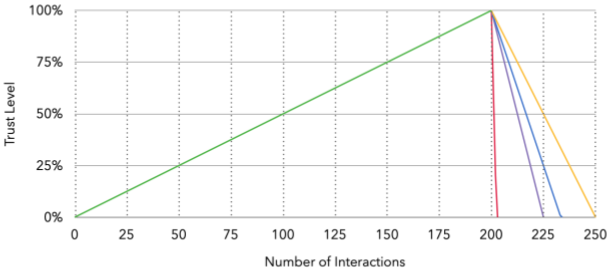


Fig. 2. Evolution of TrustLevel according to Metrics

Finally, the code size and the computation time have to be taken into consideration. A first evaluation was done on Raspberry Pi 3B+, as a gateway. Computation time and code

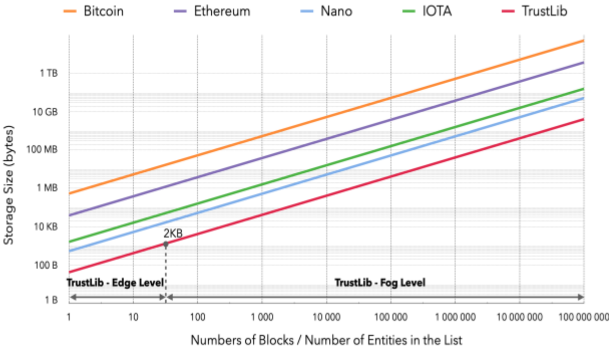


Fig. 3. Storage Size Evaluation

TABLE II. COMPUTATION TIME ON RASPBERRY PI 3B+

Operation	Average Computation Time
Normalizations	0.3 - 0.4 µs
Aggregations	0.1 µs
Initialization of the list (50 entities)	1.5 µs
Recover Trust Level	1 µs
Recover Deadline	1 µs
Get Best Node (with highest Trust Level)	1.2 µs
Trust Management (50 entities)	4 µs

size evaluations are done on Raspberry Pi 3B+ with *gcc Os* flag. As mentioned in Table II, the TrustLib protocol implies very few computation times. Its heaviest operation, the *Trust Management*, needs 4 µs to manage a trust list of 50 entities. Note that results only take into account the TrustLib processes (*e.g.* normalizations, aggregations, etc.). As a reminder, the TrustLib protocol is designed as a software library and requires about 2 KB extra code size for both sensor/actuator node and gateway.

VI. CONCLUSION AND PERSPECTIVES

The expansion of the IoT and its omnipresence raises serious cybersecurity threats. On a global approach, the trust has to be handled in this decentralized ecosystem. Despite many efforts in trust management, and recently in blockchain, there is still a lack of a dedicated solution to the IoT. In this context, we developed TrustLib, a proof of concept of a trust evaluation protocol. It is designed to incentive the quality of service while reducing energy consumption. The proposed approach enables a social characteristic of trust, namely that it is harder to gain it than to lose it, which incentive continuous good behavior. Also, results show the lightweight impact of the protocol in terms of computation times, extra code size (about 2 KB), and storage space requirements (about 2 KB to track nearly 50 entities). Future works will focus on the implementation on a real IoT board, to quantify the gains of the protocol, in terms of quality of service and energy consumption, in real conditions.

REFERENCES

[1] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "An Efficient Architecture for Trust Management in IoE Based Systems of Systems," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, Paris, Jun. 2018, pp. 138–143, doi: 10.1109/SYSOSE.2018.8428732.

[2] H. Xu *et al.*, "Trust-Based Probabilistic Broadcast Scheme for Mobile Ad Hoc Networks," *IEEE Access*, vol. 8, pp. 21380–21392, 2020, doi: 10.1109/ACCESS.2020.2969447.

[3] "Overview of trust provisioning in information and communication technology infrastructures and services." 2017.

[4] Z. Gao *et al.*, "A Credible and Lightweight Multidimensional Trust Evaluation Mechanism for Service-Oriented IoT Edge Computing Environment," in *2019 IEEE International Congress on Internet of Things (ICIOT)*, Milan, Italy, Jul. 2019, pp. 156–164, doi: 10.1109/ICIOT.2019.00035.

[5] E. K. Wang, C.-M. Chen, D. Zhao, W. H. Ip, and K. L. Yung, "A Dynamic Trust Model in Internet of Things," *Soft Comput*, vol. 24, no. 8, pp. 5773–5782, Apr. 2020, doi: 10.1007/s00500-019-04319-2.

[6] F. Bao and 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*, San Jose, California, USA, 2012, p. 1-6, doi: 10.1145/2378023.2378025.

[7] J. Liang, M. Zhang, and V. C. M. Leung, "A Reliable Trust Computing Mechanism Based on Multisource Feedback and Fog Computing in Social Sensor Cloud," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5481–5490, Jun. 2020, doi: 10.1109/JIOT.2020.2981005.

[8] B. Wang, M. Li, X. Jin, and C. Guo, "A Reliable IoT Edge Computing Trust Management Mechanism for Smart Cities," *IEEE Access*, vol. 8, pp. 46373–46399, 2020, doi: 10.1109/ACCESS.2020.2979022.

[9] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "Trust-aware and cooperative routing protocol for IoT security," *Journal of Information Security and Applications*, vol. 52, p. 102467, Jun. 2020, doi: 10.1016/j.jisa.2020.102467.

[10] Y. Hussain *et al.*, "Context-Aware Trust and Reputation Model for Fog-Based IoT," *IEEE Access*, vol. 8, pp. 31622–31632, 2020, doi: 10.1109/ACCESS.2020.2972968.

[11] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Blockchain and Trust for Secure, End-User-Based and Decentralized IoT Service Provision," *IEEE Access*, vol. 8, pp. 119961–119979, 2020, doi: 10.1109/ACCESS.2020.3005541.

[12] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Whitepaper, p. 9.

[13] V. Buterin, "A Next Generation Smart Contract & Decentralized Application Platform," Whitepaper, p. 36.

[14] T. M. Fernandez-Carames and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018, doi: 10.1109/ACCESS.2018.2842685.

[15] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain Technologies for the Internet of Things: Research Issues and Challenges," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2188–2204, Apr. 2019, doi: 10.1109/JIOT.2018.2882794.

[16] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, Nov. 2018, doi: 10.1016/j.future.2018.05.046.

[17] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251–279, Jan. 2019, doi: 10.1016/j.jnca.2018.10.019.

[18] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains," in *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, Jul. 2019, pp. 184–193, doi: 10.1109/Blockchain.2019.00032.

[19] A. Lahbib, K. Toumi, A. Laouiti, A. Laube, and S. Martin, "Blockchain based trust management mechanism for IoT," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1–8, doi: 10.1109/WCNC.2019.8885994.

[20] S. Asiri, "A Blockchain-Based IoT Trust Model," Ph.D. dissertation, Masters thesis, Ryerson University, 2018 p. 102.

[21] M. Boussard, S. Papillon, P. Peloso, M. Signorini, and E. Waisbard, "STewARD:SDN and blockchain-based Trust evaluation for Automated Risk management on IoT Devices," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, Apr. 2019, pp. 841–846, doi: 10.1109/INFCOMW.2019.8845126.

[22] A. Moinet, B. Darties, and J.-L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," *arXiv:1706.01730 [cs]*, Jun. 2017, Accessed: Jul. 17, 2020. [Online]. Available: http://arxiv.org/abs/1706.01730.

[23] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-Computing-Based Trustworthy Data Collection Model in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4218–4227, May 2020, doi: 10.1109/JIOT.2020.2966870.

[24] C. LeMahieu, "Nano: A Feeless Distributed Cryptocurrency Network," Whitepaper p. 8.

[25] "IOTA." [Online]. Available: https://www.iota.org.