Solving the cold start problem in Trust Management in IoT

Michail Bampatsikos*
National Centre for Scientific Research Demokritos,
Institute of Informatics & Telecommunications
m.bampatsikos@iit.demokritos.gr

Ilias Politis
InQbit Innovations SRL
ilias.politis@inqbit.io

Christos Xenakis
University of Piraeus, Systems Security Lab.
xenakis@unipi.gr

Stelios, C. A., Thomopoulos
National Centre for Scientific Research Demokritos,
Institute of Informatics & Telecommunications
scat@iit.demokritos.gr

ABSTRACT

Internet of Things has a profound effect on everyday life and critical vertical services including healthcare, factories of the future and intelligent transport systems. The highly distributed nature of such networks and the heterogeneity of the devices, which constitute them, necessitates that their users should be able to trust them at all times. A method to determine the device's service trustworthiness is Trust Management (TM), which assigns scores to devices according to their trustworthiness level, based on evaluations from other entities that interacted with it. Often Internet of Things devices that just joined the network, have not interacted with any other entity of this network before, hence there is no way to determine its trustworthiness. Such an event is referred to as the cold start trust score or initial trust score problem. The majority of the trust management approaches address this problem by setting an arbitrary initial trust score, while others will ignore it. Assigning arbitrary trust scores for devices connected to the network for the first time has the potential to disrupt the operation of the entire system, when a high trust score is assigned to a non-trusted malicious device, or lead to unfair policies, when trusted devices are assumed as potential intruders, which also deteriorates the performance of the system. This paper proposes a mechanism, which combines the blockchain based BARRETT remote attestation protocol with a set of device's properties and communication and operational context parameters, in order to determine accurately and assign the initial trust score to each device. Through a set of extensive simulations over different experimental setups, the proposed scheme is achieving to safely distribute initial trust scores to one thousand devices over less than 6ms, while minimising the risk of computational denial of service attacks due to the inherent characteristics of the BARRETT remote attestation protocol.

*Corresponding author.

ARES 2021, August 17–20, 2021, Vienna, Austria

ACM ISBN 978-1-4503-9051-4/21/08…$15.00
https://doi.org/10.1145/3465481.3469208

CCS CONCEPTS

• **Security and privacy**; • **Systems security**; • **Trusted computing**;

KEYWORDS

IoT, Remote Attestation, Trust Management, Blockchain

ACM Reference Format:
Michail Bampatsikos, Ilias Politis, Christos Xenakis, and Stelios, C. A., Thomopoulos. 2021. Solving the cold start problem in Trust Management in IoT. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3465481.3469208

1 INTRODUCTION

In recent years Internet of Things (IoT) networks have gained significant ground in everyday life by being an integral part in vital sectors such as the industry, the public transportation, utilities, and the healthcare. To exploit to the fullest the advantages IoT networks can offer to these sectors, it is necessary for the IoT devices to interact and cooperate with each other securely and efficiently. Inherently IoT networks are comprised of heterogeneous devices with different hardware and software configuration, while different vendors may have built these devices. Therefore, it is imperative for the IoT network provider and the different involved stakeholders to be able to unequivocally determine when a device in the network can trust another device, from a different vendor, cooperate and communicate with it.

In the context of IoT networks, trust refers to the expectation that an IoT device will provide correct, truthful and reliable service to another entity upon the latter's request. To produce an indication regarding the trustworthiness of an IoT device, in the context of correct service provision and cooperation, Trust Management (TM) methods are being deployed. TM methods evaluate, establish, maintain, update and revoke trust between devices of the same or different networks within an IoT ecosystem [1]. A typical TM method works as follows:

- An IoT device, namely the Service Requester (SR), requests a service from another IoT device known as Service Provider (SP).
- After the SP provides the service to the SR, the latter evaluates the provision of service (in terms of quality, correctness, honesty, etc.,) and assigns a score to the former. Multiple

devices may evaluate the trustworthiness of one device and give it a score.
- The TM method then aggregates all these scores and produces a numerical trust score value which in most cases is a real number from the range [0,1] with 0 signifying complete distrust and 1 signifying complete trust.

Such a trust score helps a potential SR to make an informed decision whether to request a service from an SP or not. Nevertheless, there are cases in which an IoT device has just joined the network and no device has any prior interaction with it. Consequently, there is no indication regarding the trust level of the device that just joined the network hence no trust score. This is known as the Cold Start or lack of initial trust score problem. To tackle this problem most approaches simply set arbitrarily the initial trust score of the newcomer device equal to 0.5 [2, 11, 13, 15, 17] and [6], which indicates a neutral trust level, while few approaches set it to 0 [16] and [7]. However, this score is absolute and can be far from the real trust level of the newcomer device and consequently unfair since the device's actual trust score may be close to 0. It is unlikely a newcomer IoT device's trust score is close to 1 since it has not yet provided reliable and truthful service and thus there are not enough data to predict its behavior. Finally, there are solutions that would not consider the cold start problem, thus rendering the entire work open to critique, regarding the initial level of trust an IoT device should bear when requests access to an IoT network for the first time [3, 4, 10, 12, 18], and [20]. Therefore, it is necessary to formulate a mechanism that will produce, by considering reliable certain factors, the initial trust score of the newcomer IoT device. Some TM methods [5, 9], and [21] have proposed solutions to the cold start problem. The TM method in [5] proposes to set as an initial trust score value to a newcomer IoT device the average trust score of all other nodes. In this approach, if the average trust value in the network is high then the initial trust value of the newcomer device will be high as well which is unrealistic. The authors of [9] proposed addressing the cold start problem by attesting a set of platform properties through Remote Attestation (RA), a process through which an IoT device verifies the correctness of its internal state to a remote entity known as verifier upon the latter's request. In this approach, it is not specified which is the set of platform properties that undergo attestation, while it does not consider the risk of Computational Denial of Service (CDoS) attacks that a malicious party can perform by utilizing the RA protocol. In a CDoS attack, the verifier sends multiple Attestation Requests (ARs) to the prover forcing the latter to keep executing the RA process thus keeping it busy and preventing it from performing its duties.

The lack of a coherent method to satisfy the need for establishing trust over the entire IoT network, including devices that may have just requested access without any prior interaction with other devices, is evident and obstructs the efficient and robust deployment and operation of IoT systems. Towards this end, this paper describes a solution to the cold start problem in IoT TM systems, which uniquely to other studies, it utilizes RA in conjunction with a set of weighted, device related parameters and properties to produce the cold start trust score. Moreover, the proposed scheme succeeds to protect IoT devices from CDoS attacks by incorporating the BARRETT RA protocol [19]. It is of the essence to protect IoT devices from these attacks, as they can interrupt services provisioning, while rendering useless the existence of the TM, whose ultimate role is the evaluation of the devices' ability to provide uninterrupted services. After all the purpose of TM is to evaluate the device's services provision.

The rest of the paper is structured as follows. Section 2 provides an overview of the current state-of-the-art for TM in IoT, as well as the solutions offered to the cold start problem. Section 3 presents the architecture of the proposed approach and its components as well as the sequence of operations. Section 4 includes the security assumptions and threat models identified for the proposed mechanism. Finally, section 5 provides insight regarding the performance of the proposed solution and section 6 concludes the paper as well as indicates future work.

2 RELATED WORK

There is extensive literature in the area of trust management for IoT networks, covering a variety of aspects related to protocol design and control plane signalling, security and efficiency. The aim of this literature review is to identify the role of blockchain on trust management schemes and indicate the shortcomings of these solutions in solving the cold start problem in trust management. Therefore, the literature review is split among studies which incorporate blockchain (BC) in trust management and those, which do not.

2.1 Blockchain-based Trust Management

The authors of [13] present a TM framework that detects malicious nodes that provide wrong or extreme trust values for other nodes. The main usage of BC in this approach is to facilitate the exchange of trust data among the nodes. A TM model for industrial IoT applications is presented in [15]. The solution utilises the BC to store trust related data. Each device assesses the trustworthiness of other devices and computes a final trust score for them. Lwin et.al., in [16] propose a BC-based TM system for mobile adhoc networks (MANETs This model is a policy and reputation-based TM framework focusing on constructing a trusted network, rather than on calculating trust. The work in [17] presents a TM framework for IoT which leverages the consistency and security guarantees of BC. In addition, a public BC along with smart contracts for forming secure zones of IoT devices, known as bubbles of trust, is proposed in [18]. In this approach, the BC facilitates the communications in the form of transactions among devices from the same bubble. While in [10] another BC technology, known as obligation chain, allows IoT devices to consume services of other IoT devices after agreeing to the terms and obligations present in the chain. The ultimate purpose of this approach is to establish end-to-end trust among IoT devices without relying on any common roots of trust but by utilizing the obligation chain and its built-in reputation mechanism. The authors of [12] present an anonymous reputation system for vehicular ad-hoc networks (VANETs) which utilizes BC technology as an immutable log that stores all messages exchanged in the network. This approach computes trust relying on the past and current reputation of a vehicle. The work in [11] describes a BC-based TM model to enhance trust relationships among nodes and to eradicate malicious ones in Wireless Sensor Networks. The trust

evaluation process of this approach aggregates a) behavioral-based trust, b) data-based trust and c) feedback-based trust. Boussard et al. propose in a trust assessment framework that computes a trust score for each IoT device in a smart home based on reported history the BC stores [20]. To assess the likelihood of a device behaving maliciously the proposed model considers the baseline behavior of the device the manufacturer has specified, deviations from this behavior and feedback for that device.

2.2 Non Blockchain-based Trust Management

On the other hand non-BC based TM mechanisms are also commonly studied. The authors of [2] present a model in which a node (trustor) bases the trust it has in another on its assessment of their current and past direct interactions as well as on recommendations. The model assesses trust based on a set of criteria while the trustor assigns a weight to each criterion indicating its importance. To produce a trust score the model uses a weighted sum of the specified criteria. The work in [3] describes a graph-based recommender system for IoT ecosystems that utilizes a collaborative filtering recommendation algorithm. To calculate trust among two nodes the model uses a combination of centrality and trust level. Centrality represents how central a node is in the life of another while trust level, in the context of this work, indicates relationships among devices. Furthermore, Frahat et al. designed a fully distributed TM model for IoT by utilizing the holochain technology [4]. The model has two layers, the IoT layer and the fog layer. In the first layer IoT devices provide their trust assessment after each communication among them. [6] proposes a trust approach based on Bayesian inference which detects malicious devices in a healthcare environment. The model uses multiple Intrusion Detection Systems (IDSs) and calculates device's trust values as well as identifies malicious ones via Bayesian inference. In [7], the authors describe a decentralized TM scheme for vehicular networks. To compute trust the scheme considers direct trust and indirect trust. Direct trust originates from direct interactions among two entities while indirect trust is an entity's reputation. Finally, the work in [9] and its extension in [21] propose a TM model for healthcare ICT settings that calculates an entity's trustworthiness based on history, recommendation, context, and platform attestation. Although both works use attestation as means to set an initial trust score to an entity they do not consider CDoS attacks which can prevent an IoT device from performing its duties and thus the provision of services. The approach this paper introduces considers CDoS.

3 THE COLD START TRUST SCORE FORMATION (CSTSF) MECHANISM

3.1 The CSTSF architecture's components and entities

The proposed mechanism comprises two components that participate in forming the cold start score. Firstly, the BARRETT RA protocol, which is a BC based RA procedure, contributes to the formation of a device's cold start trust score (henceforth denoted as cs_{ts}) by attesting its internal state while protecting it from CDoS attacks. Secondly, the context parameter and property-based trust score computation process. This process contributes to the formation of cs_{ts} by utilizing a set of weighted IoT device properties and operational context parameters. Both of these components form their own trust scores which the proposed approach uses to calculate the cs_{ts}. In particular, the BARRETT RA protocol brings together three individual entities.

- A BC network which is the main mechanism that protects the IoT devices from CDoS attacks.
- The Verification Nodes (VNs) which are members of the BC network, act as the verifiers in the context of BARRETT and perform computations relevant to cs_{ts}. There are two types of VNs: i) Full VNs that send ARs, verify R, store a copy of the BC and mine it. ii) Light VNs that only send APs and verify R.
- The provers which are IoT devices and members of the BC. The provers measure their internal state and produce R, which they submit to the BC and send to the VN to verify it. Figure 1 depicts the components and subcomponents of the proposed solution as well as the interactions among them.

The BC network is the core component of BARRETT since it is the one that enables the prevention of CDoS attacks against the provers. It achieves that firstly by imposing a fee for every AR a VN submits thus making CDoS prohibitively expensive in terms of monetary cost. Additionally, smart contracts that run on the BC notify the IoT device about an AR and set a limit to the number of ARs that a VN can submit on the BC regarding a specific IoT device in a predefined interval. With regards to the solution of the cold start problem, the BC contributes by enabling the BARRETT RA protocol. Besides performing the above functionalities, the BC acts as an immutable record that holds information regarding the attestation procedure. More specifically, it holds in the form of transactions the ARs that VNs submit on the BC. Another kind of BC transaction in the context of this approach is the one that contains the attestation report R which indicates whether a device passed or failed the RA process. The IoT devices as well as the VNs are members of the BC network so that they can participate in the overall CSTSF mechanism and interact with it. To this end each one of these entities has BC credentials that allow it to submit transactions to the BC.

The VNs are general purpose computers with high computational resources in terms of processing power, memory, and storage in comparison to IoT devices. They are responsible for initiating the RA procedure and verifying RA reports. They possess BC credentials that uniquely identify them to the BC and enable them to participate in the latter and interact with it. These credentials enable VNs to submit transactions containing ARs and Rs. To submit ARs as BC transactions a VN must pay a fee. To this end a VN's BC credentials have a cryptocurrency or conventional currency balance which enables it to submit ARs in the form of BC transactions. A VN signs these transactions, for authenticity purposes, with a public cryptography key pair that corresponds to these credentials. The transactions regarding R's verification do not require the VN to pay a fee for their submission to the BC. Every VN holds a secret key $\boldsymbol{K}$, which it shares with a prover and uses in an RA session and allows the former to verify the attestation report R the latter sent to it. The

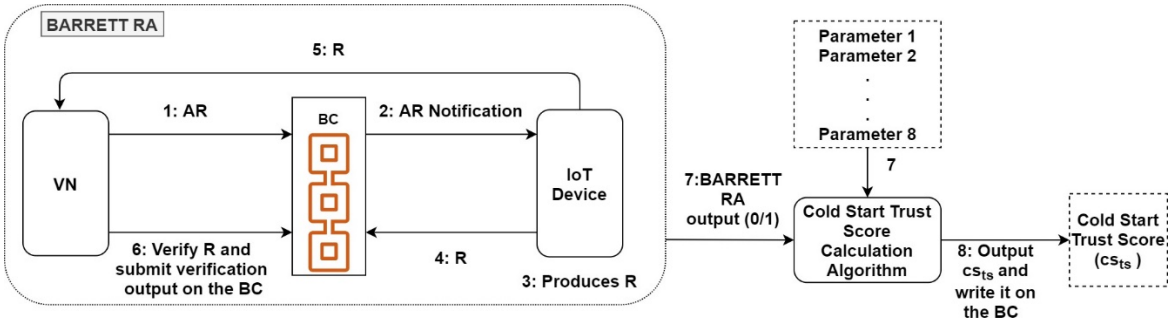

Figure 1: CSTSF's components and interactions

sharing mechanism of $\boldsymbol{K}$ is not within the current work's scope. Since VNs participate through BARRETT to the CSTSF mechanism they are also charged in the context of this work with performing all calculations relevant to the computation of the cs_{ts} of an IoT device. VNs are apriori trusted and the only malicious behavior they may attempt is performing CDoS attacks. As mentioned in the beginning of the current subsection there are two types of VNs, the full VNs and the light VNs. In particular, the full VNs submit ARs in the form of transactions to the BC, verify R a prover produced and sent to them as well as publish the verification's outcome as a transaction to the BC, and perform all computations relevant to the cs_{ts} of IoT devices. Finally, they participate in the BC consensus process and store a local copy of the BC's ledger. On the other hand, the light VNs perform all the functionalities that a full VN does but do not participate in the BC consensus and do not store a copy of the ledger locally. The Full VNs may have higher computational capabilities in terms of processing power, memory, and storage than the light VNs. That enables them to cope with the computational requirements the consensus process and the storage of the local BC ledger copy impose. Finally, although the VNs play a key role in the proposed approach their trustworthiness is not subject to evaluation since the service they provide is not the focal point of this work.

At the heart of the proposed approach lie the provers, which are IoT devices and members of the BC. The provers have BC credentials with a cryptocurrency balance and can submit transactions. As in VNs' case, the BC credentials have a public cryptography key pair that enables the prover to sign the transaction it generates and control its credentials. Moreover, the process that calculates the checksum of the prover's internal state in the context of RA resides in the prover. The prover's ROM stores the RA process's code and the secret key $\boldsymbol{K}$ that the prover shares with a VN to protect them from alteration. The prover's security architecture protects $\boldsymbol{K}$ and the Ethereum account's private key from unauthorized use. After the smart contract notifies the prover that a VN has submitted an AR addressed to it on the BC, the prover computes the checksum of their internal state and produces R. Then the prover submits R to the BC in the form of a transaction and at the same time sends it to the VN who verifies it. In most cases, provers have low-end resources in terms of computational power, storage capacity and memory. Thus, a CDoS attack can easily make them unavailable. Moreover, all trust computations the proposed approach performs pertain to these devices.

Finally, the smart contract is responsible for handling and regulating the submission of ARs to the BC. Specifically, it sets a maximum limit to the total number of ARs that all the verifiers can submit to the BC in a predefined period of time. That happens only in case all these ARs concern the same prover. If the submitted ARs reach this limit, then the smart contract accepts no more ARs concerning that prover to the BC until that period of time elapses. To perform these functions, the smart contract stores the identifiers that correspond to the BC credentials of each prover and each VN. It also stores the identifiers to impose control w.r.t. which BC nodes can participate in BARRETT and send ARs. Details regarding the deployment and implementation of the smart contract are outside the paper's scope.

3.2 Cold start trust score formation (CSTSF) mechanism's sequence of operations

The CSTSF mechanism's sequence of operation has the following three phases.

- Initially the BARRETT RA protocol executes and verifies the correctness of the device's internal state and outputs a binary value signifying the protocol's success or failure.
- Then the generation of the RA based trust score (RA_{ts}) and its assignment to the IoT device takes place. In the same phase the computation of the device's trust score based on context parameters and device properties (Henceforth denoted as $pp_{ts.}$) occurs.
- The final phase is the computation of cs_{ts} which combines the RA_{ts} and pp_{ts}.

The two processes operating in the second phase, namely the processes that generate RA_{ts} and pp_{ts}, are independent from each other and may execute in parallel. Figure 2 provides a high-level overview of the proposed solution's phases and flow of operations.

During the initial phase of the BARRETT RA protocol a VN that wishes to send an AR to an IoT device/prover pays a fee and submits as a transaction the AR via a smart contract to the BC. After the transaction validates, it is inserted to a block, and then the BC consensus process [22] adds that block the BC ledger. Once the block is part of the ledger, a smart contract notifies the prover about the AR. The prover authenticates the AR and invokes the RA process with a key $\boldsymbol{K}$, that shares with the VN node that sent

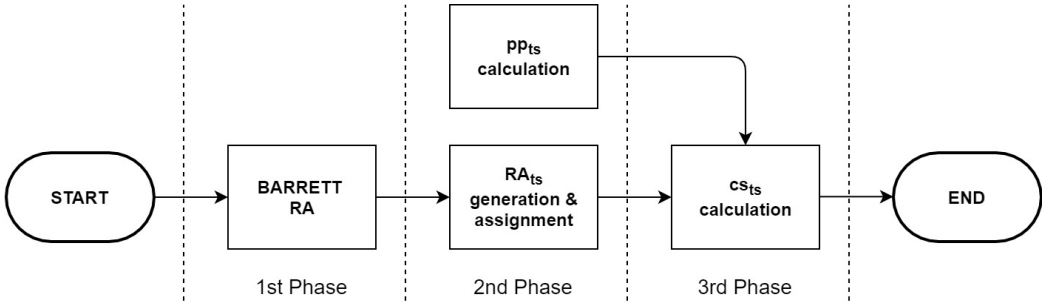

Figure 2: CSTSF's scheme phases of operation

the request, and its internal state as inputs to the process. If R is correct the VN accepts it otherwise it rejects it. Both cases play a role in the formation of cs_{ts}, while the CSTSF mechanism utilizes the outcome of the BARRETT RA process to produce the Remote Attestation-based trust score (RA_{ts}). The RA_{ts} variable can take a real numerical value which is equal to 0 or 1 depending on the attestation's outcome. In case the device is successfully attested $RA_{ts} = 1$ else $RA_{ts} = 0$. After the BARRETT RA completes and the CSTSF mechanism generates RA_{ts} the second phase of operation begins. More specifically, CSTSF computes the second part of an IoT device's initial trust score by considering device properties and context parameters.

In the context of this work the term device properties refers to the characteristics, specifications, and functionalities that may affect a device's security and the services it provides. On the other hand, context parameters refer to the various factors outside of the device that may affect the device's security and the services it provides. Each of these properties and parameters corresponds to a weight, which is a real number from the range [0,1] and indicates the impact each property or parameter has on the formation of pp_{ts}. Thus, the trust score value $pp_{ts} \in [0, 1]$ is the weighted average of the device properties and parameters.

In its third and final phase, the presented approach computes the cs_{ts} as a weighted sum of RA_{ts} and pp_{ts}. As in the case of all trust scores that the proposed approach computes, the weights of RA_{ts} and pp_{ts} (respectively denoted as RA_{wts} and pp_{wts}) in the weighted sum that produces cs_{ts} are in the range of [0, 0.5] and signify the impact of these variables to the formation of cs_{ts}. However, the value of cs_{ts} is a real number in the range of [0, 0.5] since 0.5 is the maximum value that an initial trust score can have. The defined algorithm for the CSTS formation is depicted in Algorithm 1 (Henceforth referred to as "the algorithm"). The use of weighted sum and weighted parameters to produce a trust score can also be found in similar works from the literature [2, 14].

Without loss of generality the device properties the CSTSF formation mechanism considers, include:

- The device's computational resources specifications. These specifications include processing power, RAM memory, storage capacity, and connectivity capabilities.
- Presence of crypto processor (e.g., TPM) in the device.
- Presence of security-related instruction codes (e.g., SGX) in the device.

The context parameters that CSTSF mechanism considers are:

- The device's premise, which indicates the device's location. For example, if the device is in an industrial facility or command center it is more secure, and thus more trusted compared to a device located in a house.
- The context of the application in which the device participates.
- The device's owner and operator. E.g., is the owner a user that just joined the ecosystem or is it an ICT provider that has been in the ecosystem for years.

The proposed algorithm treats the parameters and device properties as a tuple with values of 1 and 0 with 1 signifying that the parameter/property is used/present while 0 indicates that this parameter property is not used/present. However, the computational resources capabilities take a real value from 0 to 1 in this tuple since this value is essentially a computational resources score. The rationale behind assigning this value is that every IoT device has computational resources. To each entry in the parameter and property tuple a weight value corresponds which indicates the significance of each property and parameter to the computation of the pp_{ts} value.

In order for the maximum value of cs_{ts} to be equal to 0.5 the sum of pp_{wts} and RA_{wts} must be lesser or equal to 0.5. So $RA_{wts}+pp_{wts} \leq 0.5$. To this end, in the algorithm and its implementation, the values of RA_{wts} and pp_{wts} are set equal to 0.15 and 0.35 respectively. Moreover, the variables numerical_value_1 and numerical_value_2 correspond to the real and non-binary numbers 1 and 0 respectively.

Tables 1 and 2 provide an example of the values assigned to the parameters and properties tuple as well as the weight. In the algorithm there are no names corresponding to each value in the tuple, but the position of the value indicates to which parameter it corresponds to.

To each entry in the parameter and property tuple a weight value corresponds which indicates the significance of each property and parameter to the computation of the pp_{ts} value. In the event multiple devices join the network simultaneously then there are multiple tuples of both kinds. In such a case the algorithm may treat the multitude of tuples as matrices.

Table 1: Context parameters and IoT device properties tuple

Context Parameter/ Device property	Location	User owner	Orgowner	Application Context 1	Application Context 2	TPM	SGX	Computational Resources
	1	1	0	1	0	0	1	0.9

Table 2: Context parameters and device properties weights tuple

Location Weight	User owner-weight	OrgOwner weight	Application Context 1 weight	Application Context 2 weight	TPM weight	SGX Weight	Computational resources weight
0.1	0.5	0	0.9	0	0.9	0.7	1

Algorithm 1 Cold start trust score calculation

Input: Dev_{state} IoT device's Internal state
K shared key between IoT device and verifier
R BARRETT RA report
pp [] parameters and properties tuple
pp_{weight} [] weight of properties and parameters tuple
pp_{ts} property and parameter-based trust score
RA_{ts} Remote Attestation-based trust score
RA_{wts} weight of RA-based trust score
pp_{wts} weight of property and parameter-based trust score
m: end of pp tuple and pp_{weight} tuple
Output: cs_{ts} initial trust score
begin
  $RA_{wts} \leftarrow 0.15$
  $pp_{wts} \leftarrow 0.35$
/* Attest IoT device*/
  if R = True then
    $RA_{ts} \leftarrow$ numerical_value_1
  else
    $RA_{ts} \leftarrow$ numerical_value_2
  end if
  for j ∈ [0, m] do
    $pp_{ts} \leftarrow$ WeightedAverage(pp[j], ppweight[j])
  end for
/*Computation of cold start trust score as a weighted sum*/
$cs_{ts} \leftarrow pp_{ts}{}^{*}pp_{wts}+RA_{ts}{}^{*}RA_{wts}$
end

4 SECURITY CONSIDERATIONS AND PROPOSED SOLUTIONS

4.1 Security considerations

This section describes the security properties of the proposed approach as well as its threat and adversary model. The proposed approach does not consider TM related attacks since it focuses on the cold start problem of IoT devices that just arrived in the system. These devices had not time to interact with any other entity in the network. Thus, in the context of this work a newcomer IoT device has not managed yet to perform TM related attacks such as bad-mouthing, ballot stuffing, good-mouthing, self-promotion and on-off attacks [1, 8, 23].

With respect to adversaries, a potential type of adversary in the current context are malicious VNs which aim to attack an IoT device to make it unavailable to its legit users. In the context of this work VNs can only do that by performing a CDoS attack. There are two scenarios of CDoS attacks. In the first a VN may conduct a CDoS attack on its own against an IoT device. In the second scenario, multiple VNs may collude with each other to perform a more powerful CDoS attack against the prover by attempting to send multiple ARs simultaneously. The main incentive behind a collusion based CDoS attack is sharing the cost in terms of fees among malicious VNs thus making the CDoS more affordable. The trustworthiness of VNs is not subject to evaluation since, unlike IoT devices, they do not provide any service for consumption.

4.2 Proposed solutions and countermeasures

The most prominent security property of the presented approach is the absence of single points of failure. More specifically, the proposed mechanism achieves that by having more than one entity perform trust score computations. These entities are the VNs. Therefore, in case one VN becomes unavailable another may take over its tasks. Additionally, another defense of the described mechanism against single points of failure is the BC it utilizes. In particular, the BC acts as an immutable log that stores ARs, attestation reports as well as cs_{ts} in the form of transactions. Moreover, since both the VNs and the IoT devices sign the transactions they submit to the BC, the mechanism provides no repudiation of actions.

Another security property of the proposed mechanism is that it provides protection from CDoS attacks against IoT devices. In particular, it achieves that by utilizing the BARRETT RA method and particularly through the latter's AR fees and smart contracts. To perform a CDoS attack a VN must send multiple ARs to an IoT device within a period of time. However, the fees the VN has to pay in BARRETT to send multiple ARs make such an attack prohibitively expensive in terms of monetary cost. Subsection 4.1 mentions another attack scenario in which many VNs may collude to share the monetary cost of performing a CDoS attack to make it more affordable on an individual level. To counter this attack scenario, the presented approach sets through a smart contract a limit to the number of ARs that can reach a prover in a period of time. Thus, preventing concurrent CDoS attacks against an IoT device from multiple VNs. This limitation the smart contract imposes also applies to the case in which one VN sends ARs to one prover.

Another property of BC that plays a role in defending IoT devices from CDoS attacks are the delays that the transaction validation process (mining) incurs. These delays in some cases last a few seconds while in other cases can last many minutes.

5 PERFORMANCE EVALUATIONS

5.1 Experimental setup

The focal point of this work is to produce a cold start trust score computation algorithm for IoT devices. Thus, the algorithm described in subsection 3.2 was implemented in python version 3.8 on the anaconda python SDK, using the Spyder Python compiler and utilized the NumPy python mathematical library. More specifically, the parameters and properties-based trust score (pp_{ts}) calculation function was implemented as a weighted average that receives as inputs the properties and parameters values described in section 3.2 as well as their respective weights. Furthermore, the overall cold start trust score calculation functionality was also implemented which is essentially a weighted sum of RA_{ts} and pp_{ts}.

The implementation assumes that the RA process of the proposed CSTSF mechanism has concluded. Thus, the implementation of the cold start trust score computation mechanism receives as inputs only the verification outcome of the RA report which has a value equal to 0 or 1. The RA algorithm is outside this implementation's scope. The BC infrastructure was not implemented since the implementation of the algorithm starts after the RA concludes.

To evaluate the performance, resources consumption and correctness of the algorithm data were necessary. That is the reason pseudo random datasets were generated with values similar to those the tables of subsection 3.2 contain. Each dataset represents one of those two tables and are stored as csv files. Thus, there is a respective dataset for the weight of each value and for the values (properties and parameters) themselves. However, the size of each of those datasets changes according to the number of IoT devices that join the system. For example, for 500 newcomer devices each dataset has 500 rows. an additional pseudorandom dataset was generated with values which represent the success or failure of the RA process. More specifically, 0 represents failure of the RA process while 1 signifies success. Each value corresponds to an IoT device that underwent the RA process. Although in the algorithm this value is the outcome of an if statement, in its implementation for practical reasons it was decided to use this dataset and to exclude the if statement.

The algorithm retrieves these values from the datasets and handles them as two-dimensional matrices. Since the algorithm considers eight parameters and properties both the weights matrix as well as the parameters and properties matrix have eight columns while the number of rows depends on the number of devices that just joined the network. However, both matrices have an equal number of rows. Regarding the dataset that contains the RA success value, the algorithm treats it as a single dimensional array. The values in the weights matrix are real and belong to the closed range [0,1]. On the other hand, the values in the properties and parameters matrix are mixed. More specifically, the first seven columns of the matrix have binary (0 or 1) values since they indicate whether an IoT device has a particular property and that certain parameters hold true for it. The last column of the properties and parameters matrix contains values regarding the rating/score of the IoT device in terms of computational resources. The rating/score is a real number in the closed range [0,1]. The two matrices have dimensions n x 8 with n representing the number of IoT devices that just joined the network while 8 corresponds to the number of properties and parameters. The tuple containing the RA verification outcomes has n elements.

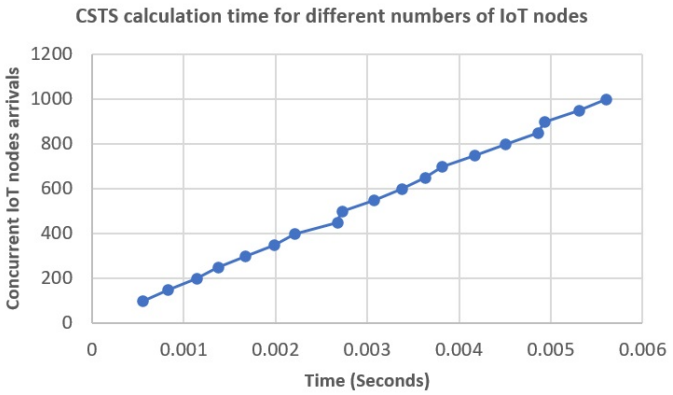

Figure 3: CSTS calculation time for different numbers groups of IoT nodes

5.2 Results and discussions

The algorithm's implementation was tested on a desktop, which in the context of this work acts as a VN, running Windows 10 equipped with an AMD Ryzen 5 CPU, 16 GB of RAM and a 500 GB SSD drive. The focus of the test was to determine the time required for the VN to calculate the cold start trust score for groups of newcomer nodes. The first group comprised 100 IoT nodes while for the remaining groups and for each iteration of the test their population kept incrementing by 50 until 1000 newcomer nodes. Figure 3 indicates that there is an almost linear relationship between the newcomer nodes and the time required to perform the cs_{ts} calculations. The duration of the cold start trust score calculation for almost all groups is in in the order of milliseconds.

The performance of the solution was also evaluated against a memory consumption point of view. More specifically, the algorithm was evaluated in terms of memory consumption for different numbers of concurrently arriving IoT nodes. As seen in Figure 4, for 250 arriving nodes the peak memory consumption reaches 0.44323 Mega Bytes (MB), for 500 newcomer IoT nodes 0.845734 MB, for 750 newcomer IoT nodes 1.2447766 MB and for 1000 newcomer IoT nodes 1.797037 MB. These measurements indicate that there is a linear relationship among the number of arriving nodes and the memory consumed by the algorithm.

The above linear relationships indicate that the algorithm will demand more computational memory resources as the IoT network grows. Nevertheless a carefull study of the obtained results reveals that for 1000 IoT nodes arriving concurrently in the network the solution ensures individual initial trust scores for all of them in less than 6 ms. This tradeoff between the computational and memory resources and the initial trustworthiness of each IoT device allowed

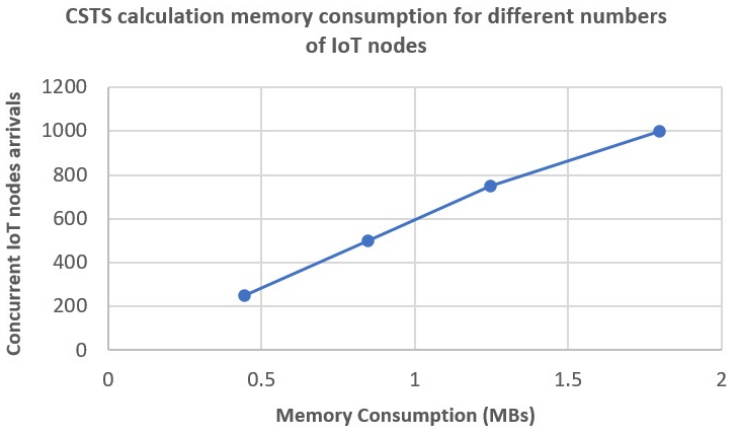

Figure 4: CSTS calculation memory consumption for different numbers of IoT nodes

in the network renders the proposed solution ideal for situations where there are no means of determining the trust level of each IoT device. Additionally, the algorithm was executed locally on a desktop environment while executing it on a cloud computing setting will yield better results with respect to performance.

6 CONCLUSION

This paper proposed a solution combining the BARRETT RA protocol with a trust score computation model to solve the cold start problem in IoT networks. The proposed approach uses an RA mechanism to partially formulate the cs_{ts} that prevents CDoS attacks against IoT devices. Something that no other TM approach that utilizes RA to compute cs_{ts} considers. However, the presented approach inherits the main drawbacks of BARRETT RA which are the high delays owed to its BC infrastructure. Moreover, BARRETT can become prohibitively expensive to honest VNs that wish to frequently monitor the status of an IoT device and thus must send a proportional number of attestation requests. From a performance point of view, the solution showcases an almost linear increase in the time required to compute the trust score for increasing numbers of IoT devices that join the network, as well as in the memory capacity required for the calculations. Nevertheless, the cost of the computational memory resources consumed is dwarfed next to the benefit of achieving a justified initial trustworthiness of each device.

Future work will be focused on implementing the RA protocol and, integrating it to the described algorithm and implementing them to the BC network. Furthermore, the optimization of the algorithm's complexity is going to be addressed through improving its implementation and runtime. The aim is to have a fully working and efficient prototype of the presented architecture that will facilitate the interaction among all components optimally in terms of efficiency and performance.

ACKNOWLEDGMENTS

The research is partially funded by Stavros Niarchos Foundation (SNF) in conjunction with EXODUS Ltd, under Grant No. 12149 "Support of scholarships for industrial PhD's and post doc industrial positions and adjunct industrial researcher" and by the European Union under the H2020 Programme Grant Agreement No. 830929 (CyberSec4Europe) and the Greek state funded Operational Programme Competitiveness, Entrepreneurship and Innovation 2014-2020 (EPAnEK) under the Grant Agreement CityZen-T1EDK–02121.

REFERENCES

[1] Carolina V. L. Mendoza and João H. Kleinschmidt. 2015. Mitigating on-off attacks in the internet of things using a distributed trust management scheme. International Journal of Distributed Sensor Networks, 11, 11, (November 2015). https://doi.org/10.1155%2F2015%2F859731

[2] Adewuyi, A. Anuoluwapo, Hui Cheng., Qi Shi, Jiannong Cao, Áine MacDermott„ and Xingwei Wang. 2019. CTRUST: A dynamic trust model for collaborative applications in the Internet of Things. IEEE Internet of Things Journal, 6,3, (June 2019), 5432-5445. https://doi.org/10.1109/JIOT.2019.2902022

[3] Navya Sri Nizamkari. 2017. A graph-based trust-enhanced recommender system for service selection in IOT. In 2017 International Conference on Inventive Systems and Control (ICISC) (pp. 1-5). IEEE. https://doi.org/10.1109/ICISC.2017.8068714

[4] Rzan Tarig Frahat, Muhammed Mostafa Monowar, and Seyed M Buhari. 2019. Secure and scalable trust management model for IoT P2P network. In 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-6). IEEE. https://doi.org/10.1109/CAIS.2019.8769467

[5] Sarah Asiri and Ali Miri. 2016. An IoT trust and reputation model based on recommender systems. In 2016 14th Annual Conference on Privacy, Security and Trust (PST) (pp. 561-568). IEEE. https://doi.org/10.1109/PST.2016.7907017

[6] Weizhi Meng, Kim-Kwang Raymond Choo, Steven Furnell, Athanasios V. Vasilakos, and Christian W. Probst. 2018. Towards Bayesian-based trust management for insider attacks in healthcare software-defined networks. IEEE Transactions on Network and Service Management, 15, 2, (March 2018). 761-773. https://doi.org/10.1109/TNSM.2018.2815280

[7] Siri Guleng, Celimuge Wu Xianfu Chen., Xiaoyan Wang Tsutomu Yoshinaga, and Yusheng Ji. 2019. Decentralized trust evaluation in vehicular Internet of Things. IEEE Access, 7, (January 2019). 15980-15988. https://doi.org/10.1109/ACCESS.2019.2893262

[8] Fenye Bao and Ing-Ray Chen. 2012. Dynamic trust management for internet of things applications. In Proceedings of the 2012 international workshop on Self-aware internet of things (Self-IoT '12). Association for Computing Machinery, New York, NY, USA, 1–6. DOI:https://doi.org/10.1145/2378023.2378025

[9] Hamed Khiabani, Norbik Bashah Idris, and Jamalul-Lail Ab Manan. 2012, June. Leveraging remote attestation to enhance the unified trust model for wsns. In Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 139-143. IEEE. https://doi.org/10.1109/CyberSec.2012.6246090

[10] Roberto Di Pietro, Xavier Salleras, Matteo Signorini, and Erez Waisbard. 2018. A blockchain-based Trust System for the Internet of Things. In Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies

(SACMAT '18). Association for Computing Machinery, New York, NY, USA, 77–83. DOI:https://doi.org/10.1145/3205977.3205993

[11] Tai-Hoon Kim, Rekha Goyat, Mritunjay Kumar Rai, Gulshan Kumar, William J. Buchanan , Rahul Saha, And Reji Thomas. 2019. A novel trust evaluation process for secure localization using a decentralized blockchain in wireless sensor networks. IEEE Access, 7, (December 2019). 184133-184144. https://doi.org/10.1109/ACCESS.2019.2960609

[12] Zhaojun Lu, Qian Wang, Gang Qu, and Zhenglin Liu. 2018. Bars: a blockchain-based anonymous reputation system for trust management in vanets. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 98-103). IEEE.

[13] Omar Ait Oualhaj, Amr Mohamed, Mohsen Guizani, and Aiman Erbad. 2020. Blockchain Based Decentralized Trust Management framework. In 2020 International Wireless Communications and Mobile Computing (IWCMC) (pp. 2210-2215). IEEE. https://doi.org/10.1109/IWCMC48107.2020.9148247

[14] Axel Moinet, Benoit Darties, and Jean-Luc Baril. 2017. Blockchain based trust & authentication for decentralized sensor networks. arXiv preprint arXiv:1706.01730.

[15] Asma Lahbib, Khalifa Toumi, Anis Laouiti, Alexandre Laube, and Steven Martin. 2019. Blockchain based trust management mechanism for IoT. In 2019 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-8). IEEE. https://doi.org/10.1109/WCNC.2019.8885994

[16] May Thura Lwin, Jinhyuk Yim and Young-Bae Ko. 2020. Blockchain-based lightweight trust management in mobile ad-hoc networks. Sensors, 20, 3, 698. (January 2020). https://doi.org/10.3390/s20030698

[17] Marcello Cinque, Christian Esposito, Stefano Russo, and Oscar Tamburis. 2020. Blockchain-empowered decentralised trust management for the Internet of Vehicles security. Computers & Electrical Engineering, 86. https://doi.org/10.1016/j.compeleceng.2020.106722.

[18] Mohamed Tahar Hammi, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni. 2018. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. Computers & Security, 78. (September 2018). 126-142. https://doi.org/10.1016/j.cose.2018.06.004

[19] Michail Bampatsikos, Christoforos Ntantogian, Christos Xenakis, and Stelios C. A. Thomopoulos. 2019. BARRETT BlockchAin Regulated REmote aTTestation. In IEEE/WIC/ACM International Conference on Web Intelligence - Companion Volume (WI '19 Companion). Association for Computing Machinery, New York, NY, USA, 256–262. DOI:https://doi.org/10.1145/3358695.3361752

[20] Mathieu Boussard, Serge Papillon, Pierre Peloso, Matteo Signorini and Erez Waisbard. 2019. 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) (pp. 841-846). IEEE. https://doi.org/10.1109/INFCOMW.2019.8845126

[21] Hamed Khiabani and Norbik Bashah Idris. 2013. Unified trust establishment by leveraging remote attestation–modeling and analysis. Information Management & Computer Security. Information Management & Computer Security 21,5, (November 2013). http://dx.doi.org/10.1108/IMCS-11-2012-0062

[22] L. M Bach, B. Mihaljevic, and M. Zagar. 2018. Comparative analysis of blockchain consensus algorithms. In 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1545-1550). IEEE. https://doi.org/10.23919/MIPRO.2018.8400278

[23] Ikram Ud Din, Mohsen Guizani, Byung-Seo Kim, Suhaidi Hassan and Muhammad Khurram Khan. 2018. Trust management techniques for the Internet of Things: A survey. IEEE Access, 7, (November 2018), 29763-29787. https://doi.org/10.1109/ACCESS.2018.2880838