Targeted Context based Attacks on Trust Models in IoT Systems

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-06-2021 / 30-06-2021

CITATION

Lewis, Cody; Li, Nan; Varadharajan, Vijay (2021): Targeted Context based Attacks on Trust Models in IoT Systems. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.14863461.v1>

DOI

[10.36227/techrxiv.14863461.v1](https://doi.org/10.36227/techrxiv.14863461.v1)

Targeted Context based Attacks on Trust Models in IoT Systems

Cody Lewis, Nan Li, and Vijay Varadharajan

Abstract—Trust models play an important role in Internet of Things (IoT) as it provides a means of finding whether a given device can provide a service to a satisfactory level as well as a means for identifying potentially malicious devices in the network. Context awareness in trust models allows a trustor to filter and aggregate evidence by their relevance to the current situation. Context awareness is important in the formulation of trust in IoT networks due to their heterogeneity and due to the dynamic changes in the capabilities of IoT devices. In this paper, we have proposed a new type of context-based attack on context aware trust models for IoT systems. An adversary is able to manipulate the context and impact a target group of IoT devices, while other devices in non-targeted groups are not even aware of the attack. We have demonstrated the effectiveness of this new type of attack on six previously proposed trust models. Through practical simulations and theoretical proofs, we show that the adversaries can launch such context-based attacks against a targeted group of IoT devices in the network. The paper also proposes a new trust management system that can mitigate such context-based attacks.

Index Terms—IoT Trust Models, Context Aware Trust Systems, Context-based Attacks

I. INTRODUCTION

TRUST plays an important role in networks as it provides a means of finding whether a given node (e.g. an entity) is able to provide a service to a satisfactory level as well as a means for identifying potentially malicious entities in the network. Trust management systems simulate the act of trust between nodes in a network through the use of filtering, adaptive, and/or cognitive processing of evidences collected on a node. Such evidence shows how a node acted in particular situations, and based on that, it gives an ability to achieve dynamic access control and predict that node's future actions. Evidence that a node has made on another is often used by other nodes in the form of recommendations; this can add perspective to the trust developed within the network. The trust model uses these evidences, and the perspective developed to calculate the trust level of a server node. Moreover, the use of static evidences creates an issue within dynamic environments, as the evidence in one case does not necessarily apply in another case.

Reputation allows a system to numerically determine the expected qualities of the recommendations that other nodes provide. It is calculated in a similar manner to trust, however, it remains independent of trust as it concerns the quality of recommendations rather than the quality of services. A

reputation system can effectively mitigate the attacks that manipulate recommendations.

Trust models are useful in Internet of Things (IoT), as they can be used to evaluate how well a node or a device may perform when serving a client, while using less expensive computations. The trust evaluation requires dynamic calculation and can account for the changes in the IoT networks consisting of heterogeneous nodes and in the capabilities of the nodes based on attributes such as battery power, bandwidth and distance. Furthermore, the IoT nodes or devices can be mobile, moving from one domain to another domain in a network as well as between different networks.

Context awareness is important in the formulation of trust in IoT networks due to their dynamic nature. The contextual properties of an IoT device such as position, signal strength, and battery power can change dynamically. The similarity between the current context and that of the evidence's context has an impact on the relevance of that evidence and can be computed using functions or algorithms like graph similarity and Euclidean distance.

In this paper, we define *context* as the properties of a node¹ that may impact on the node's ability to consume services or its ability to provide services. We will denote **context reports** or simply **reports** in this paper using the tuple, $R_i^s = (C_i^s, T_{i,j}^s)$, formed by the context vector and the node j 's trust on node i 's ability to provide the service s . This trust value is determined by the model. The context can have several parameters such as the location of a device, the type of device, its means of transportation and its battery level.

An adversary can create a fake context to cheat the trust management system. To the best of our knowledge, there are no efficient mechanisms that can check the validity of contextual data for lightweight IoT devices. This makes the manipulation of context open for abuse by an adversary. Furthermore, an adversary can attack the context of a targeted set of devices. The users of the targeted devices will be impacted while the others may not even be aware of it.

For example, in a smart healthcare system, since the privacy of patient data is of paramount importance, the trust management system can play a vital role in determining which devices should be allowed to share data for the data sharing to be secure. The context-based attack could target a particular device type or set of patients and cause them to trust a malicious server thereby leading to leakage of their data.

C. Lewis, N. Li and V. Varadharajan are with the School of Electrical Engineering and Computing, The University of Newcastle, Newcastle, Australia e-mail: {cody.lewis, nan.li, vijay.varadharajan}@newcastle.edu.au

¹In this paper, we will use the generic term node to imply an IoT device or an entity.

A. Challenges and Motivations

The Internet of Things (IoT) have widespread applications, from the battlefield [1] to the automation of entire cities [2]. Extreme heterogeneity is therefore common in these networks, e.g. an RFID device is substantially different in computational power to that of a common web server. Nodes within an IoT network thus require context awareness to be able to discern whether others will be able to perform to a satisfactory level.

In the recent times, IoT has seen extensive malicious exploitation, for example, by being transformed into extremely large botnets through Mirai [3] causing denial of service attacks. Hence, the ability to determine what nodes in the network can be trusted to perform services is of upmost importance. Hence the need for trust models that are efficient enough to allow even lightweight IoT nodes to evaluate trust.

The issue with the lightweight IoT devices is that there are, to the best of our knowledge, no efficient ways to validate the reported contexts from other nodes without resorting to a centralized server. Because a user can report any context, adversaries may target groups of nodes that operate within specific contexts. In the meantime, an adversary may evade detection systems by appearing to be honest on average with respect to other contexts, while acting malicious within the specific contexts.

B. Our Contributions

We summarize the contributions of our work as follows.

Context-based attacks. We propose a new context-based attack that can be applied to any context-aware trust model. An adversary performs attacks such as bad mouthing within the targeted context, which causes the attack to impact only those users that operate within that target context, while the others remain unaffected and even unaware of the attack.

Demonstrate context-based attacks. We analyse the impact and the performance of the proposed context-based attack on six trust management systems using both simulation and theoretical analysis. We have carried out the simulation analysis for three of the trust management systems and performed mathematical analysis of the other three. The results show that the attack is effective on all of these six cases.

Mitigating using Trust Management System. Then we develop a trust management system that is able to mitigate the proposed context-based attack. The system also mitigates the more standard recommendation based attacks.

Open source implementations. We have provided the three simulations as open source software². These are composed of the existing trust models, a simulation of an IoT network, and the adversaries for carrying out the trust model attacks. We have also provided our proposed mitigating trust management system as open source.

The rest of this paper is organized as follows. In Section II, we review relevant related works analysing the various trust models and how they compare with our work. In Section III, we present the mathematical description of the context based attacks and show that they can be applied to any trust model.

In Section IV, we apply the proposed context based attack to six trust management systems. We describe the simulation and theoretical studies that we have carried out and the impact of context attack on these systems. In Section V, we present our trust management system that can mitigate context based attacks and demonstrate our results using simulations. Finally, Section VI concludes this paper.

II. RELATED WORK

A. Trust Models in IoT

Lin and Dong proposed in [4] a trust model for social IoT systems. It evaluates trust by calculating expected gain, expected cost, and expected damage, by looking at how a node has previously performed, and how it is performing now. The trust model is decentralized and it uses the contextual information to modulate the trust on nodes. For instance, if a node is able to perform successfully under an hostile environment, then the trust on that node is increased. However, this would be susceptible to our proposed context based attack, where the attacker is able to manipulate selected contexts, thereby adversely impacting on the trust calculations.

Wu and Li [5] propose a trust model that combines the decentralized and centralized architectures by having domains, where groups of nodes within a domain directly evaluate trust on each other, and then use administration centers to calculate trust across those domains. Such an architecture is beneficial in expanding the benefits of the centralized models into a more flexible, distributed environment. Wu and Li use an adaptation of the Dempster-Shafer Evidence theory for calculating trust; however, they only account for the historical actions of the evaluated node without considering the context of the interaction. We discuss in Section II-B the importance of context awareness in IoT trust calculation. We have calculated trust dynamically [6]. This allows for filtering of old reports, which may become irrelevant. Dynamic trust calculations are important for IoT as such networks can rapidly change, for instance, due to changing battery levels of devices. Furthermore, the trust management system that we have proposed in this paper uses a decentralized architecture.

Several other works in IoT are surveyed in [7]. For instance, they discuss trust aspects in IoT access control and attack detection in IoT infrastructures. Many of these systems will benefit from contextual data and their validation, as they can further enhance the fine-granularity of the access control and attack detection capabilities of IoT devices having different capabilities.

B. Context Aware Trust Models

Uddin, Zulkernine, and Ahamed [8] propose a context aware trust model where trust on a node is dependent on the context. That is, node's trust in a particular context does not imply the same trust in a different context. It is recommendation based trust, where an entity takes recommendations from other entities about an unknown trustee to confirm their beliefs and make informed decisions. Context awareness allows a trust model under constrained computational environments to determine trust of a node more accurately. Despite such

²github.com/codymlewis/cba-on-trust-models-for-IoT-systems.

advantages of context awareness, the attack we propose in this paper allows the attacker to exploit the context to his/her benefit.

Tavakolifard, Knapkog, and Herrman [9] propose an improvement of the context based trust model using an algorithm that measures similarity of contexts to achieve transferability of trust. While this feature allows for a baseline of trust to be achieved for a node across all contexts, we can show that it also allows the context-based attack to affect the overall trust of the target.

Truong, Lee, Askwith, and Lee [10] describe a trust model for the Social IoT environment using reputation, experience, and knowledge (REK) for trust evaluation. These parameters are computed using pieces of evidence obtained from the past transactions along with the context of those evidences. Similarly, [11] proposed a context aware trust model which makes trustworthiness predictions based on the the correlations of normalized and filtered records of the behaviours of nodes on the basis of the clusters of contexts the records belong to. While REK based models are effective for predicting trust reaching across the network, they also allow for unchecked attacks to impact many more nodes indirectly. We observe this effect with context-based attack, where through recommendation, the attack can impact an entire group of nodes in a target context.

Otebolaku and Lee [12] propose a context aware trust-based framework for IoT, which determines similarities of contexts though ontologies and uses cognitive context to reason and relate them. Ye, Wang, and Liu [13] propose a context aware trust model for crowd-sourcing environments, where it models two forms of context aware trust. One is based on the type of task, whereas the other is based on the reward associated with the task. Though these provide a different way of calculating context based trust, still these models are not immune to the proposed context-based attack. As these models do not validate context, they are vulnerable to our proposed attack. For example, an adversary could impersonate devices close to the target in the ontology, or create and complete tasks with carefully selected types and rewards.

Alshehri and Hussain [14] provides a comparative study of trust management protocols for IoT. They discuss dynamic and scalable trust management, such as through community of interest where the trust calculations are distributed into subsets of the network. Rafey, Abdel-Hamid, and El-Nasr [15] also propose a context aware trust model which uses a community of interest system. However, it additionally uses a different calculation for indirect recommendations to mitigate the effects of bias between communities. For our proposed system in Section V, we opt for a distributed architecture rather than community based approach to achieve greater flexibility.

Neisse, Steri, Baldini, Tragos, Fovino, and Botterman [16] show how context aware trust models can be used to control access and flow of user's data to achieve security and privacy. Habib and Leister [17] consider an application of this approach, where context aware trust models can be used to provide authentication for IoT. They show how context can be used to influence security based decisions in a dynamic form, such as for role-based access control, assigning roles

dynamically based on context. In this case, the context-based attack will impact the security and privacy of the IoT system.

C. Trust Management in Other Domains

Web of trust is a widely used trust management system [18]. Here the user elects trusted introducers where if two of them trust another user then this other user is marked as trustworthy. Another domain where trust management plays a key role is in the area of cloud services. Cloud systems offer better facilities for trust management including for context validation due to the availability of computer power and their distributed nature. Our context based attack has been targeted at environments where accessing to cloud services may not be feasible (see Section III-A on Threat Model below). The attacks observed in these other domains often involve alteration of the reported recommendation values. Examples include the bad mouthing attack where benign servers are reported as malicious, good mouthing where malicious servers are reported to be benign, and the on-off switch which toggles between good and bad mouthing for a certain period of time [19], [20]. Often the use of reputation systems and evaluations of recommendation can help to counter such attacks. Another common attack involves the exploitation of identity services or the lack of such services. Adversaries may masquerade as different users or perform Sybil attacks by creating many fake identities [21]. Identity attacks may be countered through verification mechanisms such as signatures. In this paper, our proposed context-based attack acts as an extension to enhance the effects of recommendation-based attacks.

III. THREAT MODELS AND CONTEXT-BASED ATTACKS

Context aware trust management systems in IoT tend to be based on a fusion of the reputation and recommendation based systems. Trust is calculated using recommendations from other users via reports, as mentioned earlier in the form of tuples $R_i^s = (C_i^s, T_{i,j}^s)$. Reports are aggregated and scaled to calculate reputation, which characterizes the quality of reports from the sender. However the issue arises as to whether to trust the reported context. Several trust management systems in the literature apply reputation to the reported trust value by blindly trusting the reported context. Our proposed attack exploits this, allowing an adversary to appear benevolent on average, while targeting a specific group of users within a chosen context with chosen trust fabrication attacks.

A. Threat Model

Our focus is on the formulation of trust in IoT devices to determine which services provided by the IoT devices can be reliably used. We assume that devices do not have reliable computation, communication, or capacity, which in turn affect the choice of service being provided and its context, both of which can be dynamic. Furthermore, this implies that a cloud-centric model can not be reliably and effectively accessed. Also that users may not be able to perform computationally heavy operations such as using some machine learning algorithms for anomaly detection and linear aggregations over many dimensions.

B. Context-based Attacks

The goal of the context-based attack is to modify and enhance conventional trust model attacks such as bad mouthing. The context-based attack deliberately chooses to impact a subset of devices in the IoT infrastructure. The subset of IoT nodes affected can be those that request or provide a service under the selected context that is targeted by the adversaries. The adversaries perform the attack by reporting their spoofed trust in the targeted context, and report their true trust when reporting outside of the target context. If successful, the adversaries are able to manipulate the targeted subset IoT devices in the network while devices outside remain unaware of the attack.

The reason why the manipulation of reported context in recommendations in IoT trust models is possible is due to the restricted computational capability of IoT devices not having context validation mechanisms. Furthermore, often the design of trust management systems are driven by their efficiency and their lightweight computations which further make the context manipulation easier (compared to more computationally intensive cryptographic or machine learning techniques). Hence our threat model is concerned with the capability of the adversary to manipulate the context to his or her benefit.

The impact of a context-based attacks is dependent on how the recommendations are handled in the trust models. Those that filter the recommendation reports on the basis of similarity to the current context, C_c^s , are susceptible to the attacks if the filtration causes the majority of the left over reports to be those from the adversaries. For a target context, C_t^s , the distance d from the adversary reported context, C_a^s , is as follows,

$$d(C_t^s, C_a^s) = 0. \quad (1)$$

There can be an inclusion threshold factor, $D \geq 0$, in which case the adversary's reports will always be included in the calculation of trust under the target context, even after filtration.

Then there are trust models that calculate the impact of the reports based on the similarity between the reported and the current context will be impacted to varying degrees with respect to the distance between the current context and the target. Since the resulting trust will be scaled based on the distance, having a distance of 0 from the targeted context will cause to report to have the maximum impact, that is, an impact value, $0 \leq \eta \leq 1$. Then the recommended trust would be calculated as,

$$T^s = \eta^{d(C_t^s, C_a^s)} = \eta^0 = 1. \quad (2)$$

In this case, there does not need to be a majority of adversaries that fit within the context, especially if this impact factor has an effect on the reputation. With respect to reputation, if the report is compared to others and the impact factor is used, then the group of adversaries will collaboratively raise their reputation due to having agreement with each other and by having a maximum impact value.

C. Combining Context-based and Recommendation-based Attacks

As the context-based attacks by themselves may not have significant adverse impact, the attacker will simultaneously carry out additional recommendation-based attacks. The aim of the context-based attack is to create targeted trust recommendation attack, while, through aggregation, appearing benign and thus evading the trust management system's detection mechanisms. The following are three common recommendation-based attacks that can be combined with the context-based attack.

- **Bad mouthing attack:** Here the adversary constantly reports that the service provider is malicious without regard to the actual quality of the provided service.
- **Good mouthing attack:** Similar to bad mouthing, except here the adversary reports that the service provider is not malicious and provides good quality services.
- **On-off attack:** Here the adversary toggles from bad mouthing to good mouthing and vice versa in a dynamic manner.

IV. CASE STUDIES: CONTEXT-BASED ATTACKS AND VULNERABLE TRUST MODELS

In this section, we analyze the susceptibility of six existing context-aware trust models to the proposed context-based attack. The reason behind the choice of these trust models is that they have different system structures such as centralized and decentralized as well as they employ different techniques such as filtering and scaling. We demonstrate that the impact of the proposed context-based attacks is significant in each of these cases.

A. CBSTM IoT [15]

The first trust model we analyzed was the one proposed in [15], which is representative of a distributed social trust management system with discrete context trust calculations. The context-based adversary only performs the bad mouthing attack when the requested context belongs to the target context, otherwise the adversary behaves honestly. The goal of the adversary is to cause the other nodes to distrust services of the nodes in the target context.

1) *Implementation Details:* Our trust model simulation contained 100 nodes with variable number of adversaries. Friendship between nodes dictated which nodes interacted, which was determined randomly ensuring that the average amount of friends that each node has was approximately 50. We assigned one observer in our implementation, which exclusively uses indirect trust and does not perform any direct transactions.

Nodes in the simulation were implemented to act benevolent within any context value that is less than or equal to the their randomly assigned context, and for contexts above that, they will act as malevolent. A context, c_i is less than or equal to another context, c_j , iff $i \leq j$.

At every time unit, all the nodes except the observer perform a transaction with the targeted node. So each node sends a recommendation to the observed node, allowing the observer

TABLE I
PARAMETERS OF OUR IMPLEMENTATION OF THE CBSTM IoT [15]
TRUST MODEL

Parameter	Value
Weights of contexts (w)	$\{0.5, 0.2, 0.2, 0.1\}$
Weight of friend commonality (w_h)	0.50
Weight of social network centrality (w_d)	0.50
Initial trust value (R)	0.5
Computing Power of devices (CP)	1
Maximum reported feedback for a service (fb_{max})	10
Ratio of reputation change (MD)	0.1
Contexts	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$
Total nodes	100
Transactions	250
Maximum attainable trust for an observer	0.5
$\{f_{r\text{low}}, f_{r\text{high}}\}$	$\{30, 70\}$
Relationship factor (R) between nodes i and j	$1/2^n : n = d(i, j)$

Fig. 1. The trust in the targeted context with varying number of adversaries.

to calculate the trust using every possible recommendation. Since there are transactions within each of the contexts at each time unit, recommendations for each node do not timeout and remain for the entire duration of the simulation.

2) *Results*: In Fig. 1, we observe how the attack maintains the initial state of distrust with logarithmic effectiveness with respect to the percentage of adversaries. With 100% adversaries, the system initializes and retains complete distrust in the service provider. With less adversaries, the trust builds over time as expected of the system, when evaluating a good service

Fig. 2. The trust outside of the targeted context with varying number of adversaries.

provider. However, when there are more than 0 adversaries, the convergence points of the trust are lowered relative to the percentage of adversaries that compose the network at a logarithmic rate. We observe that 20% adversaries causes the system to trust the service provider only very slightly, while by 50% the service provider is no longer trusted.

Fig. 2 shows that outside the target context, no adverse effects are experienced. The average recommendation remains in a trusted state for any amount of adversaries, as honest recommendations outside the context far outnumber the adversary's attacks within the target context.

In this case, we show how an adversary may cause nodes operating under a targeted context to be forced into distrusting an honest service provider. While the attack also has no effect on the nodes that are not targeted. By having localization of attacks, adversaries can selectively attack particular collections of nodes and thus better evade detection. When the minority of nodes in the target group reports the network experience as bad while others report the experience as good, the opinion of the majority is considerably more important. As a result, from the point of view of a trustor, the network simply does not operate well under that context.

B. Saied, Zeghlache and Laurent's Model [22]

Saied, Olivereau, Zeghlache, and Laurent [22] propose a centralized trust management system for peer-to-peer communications between IoT devices. It provides context awareness by taking into account the service providing capability, service provided, and the time for which the service was provided. The system operates through a cycle of four phases: (i) bootstrapping, where artificial transactions are induced to provide observations of the service provided in the network;

(ii) entity selection, where a list of possible service providers, ordered by trust levels, is sent from the trust management system for the trustor to make the appropriate choice; (iii) the transaction phase, where the trustor performs a transaction with the chosen service provider; and then (iv) reputation update phase, where using feedback from the transaction and the quality of recommendation, the reputation is updated. The centralized trust manager runs the algorithms in these phases and informs the results to the nodes in the network.

1) *Implementation and Configuration:* Our simulation was implemented with the parameters shown in Table II. We simulated a network of nodes that are defined with an arbitrary id, a service value in the range $[1, 100]$, a capability value in the range $[1, 100]$, and the accuracy of the recommendation. The accuracy reflects the ability of the recommending node to make accurate recommendations. This is dependent on the observations that the recommending node is able to make. For instance, the recommending node may not have a complete view in which case it is a poor witness. That is, its observations may be incomplete and hence its recommendation may not be accurate. Hence the accuracy is the probability that a node will report an accurate recommendation. Poor witness nodes were assigned a random value in the range $[1, 100]\%$, and other nodes were assigned 100%. The service and capability values were assigned as 100 for non-constrained nodes and a uniformly random value in the range $[1, 100]$ for constrained nodes. The constrained nodes were selected randomly. For the evaluation of the targeted effect, we created a target group, a set of nodes which were assigned a random capability and service in the range $[45, 55]$, and nodes outside of that group were assigned anything that is not in this range. Then we had a specified percentage of the nodes assigned to be malicious. These nodes will provide a malicious service when acting as the proxy serving node.

In each epoch, there is a bootstrapping phase, which forces random transactions between all the nodes in the network. In our simulation, we have set the bootstrapping phase to perform $5 \times |Nodes|$ artificial transactions. For the next two phases, a random client is chosen, and then entity selection is applied to choose the optimal server and a transaction is performed between this client and the server. Finally, the quality of recommendation (QR) and reputation are updated. The QR update is performed on each of the witness nodes, and reputation is updated on the proxy server node. The reports generated from the transactions are as previously defined in the form of a tuple with context and recommendation. The recommendation and by extension trust within this system are in the range $[-1, 1]$, where -1 represents a malicious service, 1 represents a good service, and 0 when the service is provided as expected. The recommendation is what the witness nodes assign; note a poor witness node may assign this incorrectly without purposefully being malicious. The time in this simulation iterates by one for every 60 epochs. Our choice of iteration time is based on the following: if the iterating time is too often, then this induces too much decay on the QR and reputation, whereas not often enough gives little to no time decay effect.

TABLE II
PARAMETERS OF OUR IMPLEMENTATION OF SAIED, OLIVIEREAU, ZEGHLACHE, AND LAURENT'S [22] MODEL

Parameter	Value
Memory parameters (θ, λ)	0.7, 0.7
Report distance acceptance factor (η)	1
Total nodes	200
Transactions	500
Poor Witnesses	20%
Constrained Nodes	50%
Malicious Serving Nodes	10%
Reputation Threshold	-1

2) *Effects of Context Attacks:* We observe that the average trust value substantially fluctuates between transactions. This is due to random services being requested for every transaction and hence a larger value is indicative of a greater percentage of nodes in the network being trusted to perform the service. Majority of the graphs resulting from our simulations show trust to be negative for most transactions. This is due to the adversarial reporters outnumbering the honest reporters. In these simulations, the context-based attack involves setting each context value to 50, except for the time parameter which gets set to 5 subtracted from the current epoch. Context-based adversaries simultaneously perform the bad mouthing attack. The goal of the attack is to fool the target group into believing that all the serving nodes are malicious.

From both Fig. 3 and 4, we can observe the successful targeting effects of the attack. With the target group having capabilities in the midpoint of the possible contexts, we would expect the fluctuations of the average trust to be close to the overall average. However, the adversaries leave the target group's trust to mostly fluctuate in a more limited range of 0.0 and -0.3 .

We further observe in Fig. 3 and 4 that with higher percentages of adversaries, the overall distrust in serving nodes by the target group can be achieved without exceeding the Byzantine fault threshold. In Fig. 3, all users of the system are adversely effected as the unchecked targeted attack allows the adversaries to have sufficient amounts of agreed upon recommendations that their bad mouthing attack is strengthened to apply across all contexts. This is possible due to the systems lightly influencing context distance mechanism in the determination of the importance of the reports.

At greater percentages of adversaries, their influence over the entire system weakens for the sake of better targeting, as shown in shown in Fig. 4. This is also caused by the report distance system; however, due to more reports sitting at a specific point on the plane, the averages of the reports are shifted especially since those closest to the current state have prioritized influence. Those that sit near the target point are more heavily influenced by the adversaries, while those further away are more influenced by others.

Fig. 3. Comparison of the average trust of the target group to the normal group, when 20% of the network are context-based adversaries

Fig. 4. Comparison of the average trust of the target group to the normal group, when 30% of the network are context-based adversaries

C. CA4IOT [23]

Parera, Zaslavsky, Christen, and Georgakopoulos [23] propose an architecture for IoT that automates the task of selecting sensors in the network based on the tasks at hand. It is a layered architecture that records contextual data from the devices in the network, and then processes it to be ready for filtration and analytics. A front end interface enables the nodes to use the stored data to find out what is occurring in the network. The system can define rules where a combination of context values is used to determine whether there is some

property in the network. For example, in a smart farming network, if there are sensors in an area stating that the humidity and temperature are at some dangerous point, then the system could report that the crops in that area are likely to have a disease.

Again, this architecture is susceptible to context-based attacks. In order to demonstrate this attack, consider the network having adversaries, which are impersonating sensors. Let the context values recorded by the sensors be $C = \{c_1, c_2, c_3\}$, the target contexts be $C_t = \{c_{t1}, c_{t2}, c_{t3}\}$. Let the following predicate be used to determine the value of rule, $R \in \{true, false\}$,

$$R = (c_1 \geq c_{t1}) \wedge (c_2 \geq c_{t2}) \wedge (c_3 \geq c_{t3}). \quad (3)$$

The adversary could fool the system into falsely identifying $R = false$ even when it should be $R = true$. First, the adversary would spoof the sensing data with the context values set as $C = C_t$. Then when the end user checks the interface, areas where adversaries are present will report the rule as not being *true* even though in reality it should.

D. Toivonen, Lenzini and Uusitalo's Model [24]

Toivonen, Lenzini, and Uusitalo [24] propose a framework that adds context awareness to existing trust models. They defined equations for modification of trust and reputation based on context, an ontology structure to relate trust and acting as a way for evaluating recommendations. The framework introduced a similarity function, s , to determine the similarity between two contexts. Filtration of recommendations is then based on that similarity, and ind_x and dec_x are used to increase and decrease trust by a factor of x respectively. Let us now consider how context-based attacks are effective against this framework.

1) *Attack Explanation:* The trust model in [24] defines the following binary predicate, which is used to determine the similarity between the elements of the set of contexts of the trustor, $c^s \in C^s$, and the recommender, $c_j^s \in C_j^s$. There is a predicate function applied to the context values, $p_k : c \mapsto \{true, false\}$.

$$c^s \sim c_j^s \iff p_k(c^s) == p_k(c_j^s) \quad (4)$$

The predicate is then used in the similarity function to determine the reputation of the recommendation,

$$r_j = s(C^s, C_j^s) = \frac{\sum_{k=1}^m w_k \cdot (c_k^s \sim c_{j,k}^s)}{\sum_{k=1}^m w_k} \quad (5)$$

When recommendations are taken into account, they form a tuple (r, C^s) , where r is a reputation value and C^s is the context. A node takes the set of recommendations $R = \{(r_u, C_u^s) | u \in S\}$ from a set $\mathcal{R}$ of recommenders and filters it into the set, $R^j = \{(r', C'^s) \in R : s(C^s, C'^s) > D\}$, where D is a compatibility threshold decided by the trustor.

A context setting attack can effectively reduce the trust of a target in the network by matching the context to that of the trustor. This will cause the predicate in (4) to always evaluate to be true, and the function in (5) to always evaluate to be the maximum possible value, which is 1 when there is no

time based decay on the context values. This means that if T is set less than or equal to the maximum attainable value of (5) for the model implementation, then the recommendations made by context setting adversaries will be evaluated to be the most relevant. This leads to the context setting attackers having control over the recommendation component of the model.

2) *Example:* Let us define the following functions, $inc_y(x) = \sqrt[y]{x}$ and $dec_y(x) = x^y$. Let the initial trust $t_0 = 0.6$, and the first three contexts to be below the expected context threshold to the trustor. Also let the trustee have a good reputation according to the trustor. A context setting attacker will then have a substantial influence on the result as its recommendations will always be deemed the most relevant by the trustee. The difference may be observed as follows: The trust before the recommendations are calculated is given by

$$t_4 = inc_{1.25}(dec_{1.5}(dec_{1.5}(dec_2(0.6)))) = 0.16 \quad (6)$$

Then when applying the recommendations with the set of weights for the context $w = \{1, 2, 1.5, 1.5\}$, the adversaries report their context exactly as the target, whereas honest nodes are likely to not report the exact matching context. Then the results for the distance between adversary reports and the target context are as follows:

$$s(C^s, C_j^s) = \frac{2 + 1.5 + 1.5}{2 + 1.5 + 1.5} = 1 \quad (7)$$

Then for the honest nodes where, for example the third context value differs, the similarity function evaluates to be a lower value,

$$s(C, C_j^s) = \frac{2 + 1.5}{2 + 1.5 + 1.5} = 0.7 \quad (8)$$

Therefore in the most likely situations, the similarity function evaluates the adversaries having more relevant recommendations than the honest nodes. Hence they will then have the final influence over the trust model, which they may then use to do good mouthing attack resulting in the following,

$$inc_{1.25}(t_4) = 0.23. \quad (9)$$

Or the adversaries can do a bad mouthing attack resulting in,

$$dec_{1.25}(t_4) = 0.1. \quad (10)$$

The impact of the context setting attack will then depend on the weight given to recommendations by the trustor, since this is the only part over which the adversaries have control. However, as the implementation context is of major importance in IoT, the chosen weight is likely to be large enough to have a significant impact on the resulting trust.

E. Yang et al. Model [11]

The context aware trust model in [11] uses behavioral data analysis. In this model, trust is calculated across a set of behavioral properties, $I = \{I_1, I_2, \dots, I_n\}$, each of these acting as trust evidences and are aggregated to calculate the final trust. The context-based attacks will have significant influence on this trust model in most cases.

1) *Attack Description:* The adversaries will perform the context attack by sending recommendations to all other nodes where the reported trust from node j , $T_{j,I_q}^s : \forall I_q \in I$, is bad mouthed to the value of -1 . These recommendations, (T_j^s, C_j^s) , will be reported within the target context, C_t^s such that,

$$d(C_j^s, C_t^s) = \sqrt{\sum_{c_i \in C_{I_q}} (c_{j,i}^s - c_{t,i}^s)^2} = 0, \quad (11)$$

That is, the context within the recommendations will be exactly the same as that of the target context.

This trust model only uses the recommendations which have similarity to the current context below the threshold parameter, D . So the reports from the adversaries will always be included for all trust calculations under the targeted context. While in most cases this will mean that the context setting attackers will have significant influence over the resulting trust, there are some cases where the adversaries can be outnumbered and they will not have a major effect on the trust model. The influence of the attack on the trust is shown below.

2) *Attack Example:* In order to show the effect of the attack, consider the following simple example. Let $I = \{I_1\}$, $w_1 = 1$, the recommendations from the adversaries have a trust of -1 , and the recommendations from the other nodes have a trust of 1. This simplifies the trust prediction into the following function,

$$T^s = \sum_{i=1}^n w_i \times T_{I_i}^s = 1 \times T_{I_1}^s = \frac{1}{|\mathcal{R}_D|} \sum_{j \in \mathcal{R}_D} T_{j,I_1}^s, \quad (12)$$

where $\mathcal{R}_D$ is the set of recommendations with a similarity to the target context below the threshold parameter.

For the case where only the adversary's recommendations are relevant, the following holds:

$$T^s = \frac{-1 \cdot |\mathcal{R}_D|}{|\mathcal{R}_D|} = -1. \quad (13)$$

From this case, we can also observe that the trust will be negative for all cases where the following inequality holds,

$$\sum_{T_{x,I_1}^s \in \mathcal{R}_{I_1}^a} T_{x,I_1}^s < \sum_{T_{y,I_1}^s \in \mathcal{R}_{I_1}^n} T_{y,I_1}^s, \quad (14)$$

where $\mathcal{R}_{I_1}^a$ is the set of relevant recommendations from the adversaries, and $\mathcal{R}_{I_1}^n$ is the set of relevant recommendations from the non-adversaries. This can be shown by substituting the attributes of (14) into (12), with $\alpha = -\sum_{T_{x,I_1}^s \in \mathcal{R}_{I_1}^a} T_{x,I_1}^s$, $\eta = \sum_{T_{y,I_1}^s \in \mathcal{R}_{I_1}^n} T_{y,I_1}^s$, and $S = |\mathcal{R}_D|$,

$$\mathcal{T} = \frac{-\alpha + \eta}{S}, \quad (15)$$

Since $S \geq 1$, $\mathcal{T}$ will always be less than 0 when $\alpha > \eta$. Therefore, the context attacks will always make innocent nodes appear malicious when either their relevant recommendations outnumber those of the non-adversarial recommendations, or when the adversary's recommendations result in a more extreme aggregation.

F. Li, Varadharajan and Nepal's Model [25]

We have implemented a simulation of the trust model described in [25]. This trust management system uses distributed calculations where reports are scaled by distances between reported context and the weighted average of reported contexts.

1) *Configuration and Implementation:* With this model, we have implemented a practical simulation involving IoT based tracing of animals moving in a field, and the resulting network where connections are routed between devices. The parameters of the simulation were set to match those presented in [25]. Each context value, excluding time, were normalized according to (16), where C^s is the input context value, C_{max}^s is the maximum possible value for the corresponding context. This places them within the range $[0, 1]$, while time remains a natural number.

$$Norm(C^s) = \frac{C^s}{C_{max}^s} \quad (16)$$

For every time slice, every node in the network moves towards its goal and performs a transaction with the service provider. At this point, each node calculates the trust to determine (a) whether it should perform the transaction, and (b) whether it should send a recommendation to their contacts. If a node performs a direct transaction with a service provider, then a feedback value is set to be $t = 100$, indicating a good trustworthy service has been performed. Trust recommendations in the simulation were always the direct trust feedback combination from the current time slice.

2) *Results:* In this case, our context-based adversaries combine context spoofing with bad mouthing with the aim of causing the target group to distrust the service provider. In order to avoid the repercussions from the reputation system, the adversaries only perform the attack occasionally, allowing their reputation and thus their influence to recover. Another element of the system that the adversaries had to account for is the timeout system. A node that reports a negative recommendation can not report another about the same node until the current reported context is sufficiently distant from the cached context from the last used report (the amount of distance required is determined based on a parameter). To evade this, the adversaries perform a two stage attack. In the first step, the attack has the report with the current time subtracted by the parameterized amount, and in the next step, the attack has the report with spoofed context but with the current time. This shifts the cached context away from the current context to satisfy the timeout condition so the next step can be performed without being mitigated. The second step performs the attack at a context relevant enough to have a significant impact on the targeted users.

TABLE III
PARAMETERS OF OUR IMPLEMENTATION OF [25]

Parameter	Value
Maximum number of contacts	100
Initial Reputation (I_r) of a contact	0.01
Initial trust of a new contact	0
Range of Trust/Reputation	$[-1, 1]$
Number of context items	4
Trust/reputation threshold (τ)	0
Trend threshold (δ_r)	0.01
Trust/reputation adjustment range	$[-0.001, 0.001]$
SP ground trust	1
$\{w_0, w_1, w_2, w_3\}$	$\{0.3, 0.2, 0.4, 0.1\}$
$\{\eta, \eta_1, \eta_2, \eta_3, \eta_4\}$	$\{0.95, 0.7, 0.5, 0.5, 0.7\}$
α, β, γ	0.3, 0.3, 0.8
ρ, δ, δ_a	0.1, 0.8, -0.001
$\{\theta_i\}, \{\mu_i\}, \{\eta_i\}$	$\{0.8\}, \{1.0\}, \{1.0\}$
Transactions	500
Maximum attainable trust for an observer	1
Feedback summation result ($t + d + u$)	100

Fig. 5. Trust values experienced within the targeted context

Fig. 5 shows how the adversaries can successfully influence trust of the targeted nodes once they form half of the network. At this point, the major impact on the resulting trust is in the reputation. When there are sufficient adversaries influencing the system, their spoofed recommendations cause honest recommendations to appear as anomalies and therefore cause the non-adversaries to hold negative reputation. Hence, when the threshold for influence is achieved, there are diminishing results in increasing the adversaries, as the attack is more reliant on the honest nodes than the adversaries. Once the entire network is composed of only the adversaries, their

Fig. 6. Convergence points for the trust by number of adversaries

Fig. 7. Trust values experienced outside of the targeted context

negative recommendations collude to the point where they reduce each other's reputation making them to be close to zero. Fig. 6 plots the discussed changes in average trust on the basis of the number of adversaries. Conversely, Fig. 7 shows the trust values experienced by the normal groups. The only benefit achieved by the attack over normal bad mouthing is in the evasion of the timeout mechanism.

These graphs were generated from the perspective of an observer device, a device that exclusively calculates indirect trust and never performs direct transactions or makes recommendations. The observer had 10 contacts with varying number

of adversaries. The adversaries performed their transactions before the other devices during each time slice, and were in contact with all other nodes in the simulation, so they would directly attack all non-observer nodes.

With the simulations, a new network and map were created for each change in number of adversaries. This means that each line in the graphs in Fig. 5 and 7 reflects its own individual simulation, causing slight random variations between the cases. We chose those cases which were most comparable between the types of attacks in order to create better observation of the differences.

V. A TRUST MANAGEMENT SYSTEM TO MITIGATE CONTEXT-BASED ATTACKS

In this section, we propose a lightweight trust management system that can successfully mitigate context-based attacks. Our system uses a distributed architecture and treats context values as continuous. By lightweight, we mean the most computationally expensive operation in the system, namely the context aggregation, has a complexity $\mathcal{O}(\zeta|\mathcal{N}|)$, where ζ is a system parameter, and $\mathcal{N}$ is the set of nodes in the network.

A. System Design

In this system, each device individually computes the trust and reputation for each other device it communicates with. The primary features that mitigate context-based attacks are the quality of context, observing the context in which the recommendation is received, and having certain elements that reward changing of context while others reward stable context.

We compute trust using (17), where direct trust T_{dir}^s is the trust accumulated based on direct interactions with the service provider, and indirect trust T_{ind}^s is the trust computed using recommendations from other users. Indirect trust has a diminishing influence over the result as the direct trust increases, however, it never reaches zero.

$$T^s = \alpha T_{dir}^s + \beta T_{ind}^s \quad (17)$$

The impact of the direct trust is scaled by (18). It evaluates the strength of the direct trust in comparison to its distance, $d(a, b)$, from the indirect trust and moves the input into the range $[0, 1]$ through the sigmoid function $\sigma(z) : z \geq 0$.

$$\alpha = \sigma \left(\frac{2|T_{dir}^s|}{d(T_{dir}^s, T_{ind}^s)} \right) \quad (18)$$

The impact of indirect trust is then scaled by (19), with the leftover possible maximum value for $T \in [-1, 1]$.

$$\beta = 1 - \alpha \quad (19)$$

Direct trust calculated by (20) is computed by a node every time it interacts with the service provider. It is based on gradient descent, where the gradient is a feedback value, $fb^s \in [-1, 1]$, which is some quantity stating either how close a given output is to the expected service outputs or an anomaly value computed by an intrusion detection system. To ensure the trust remains within the bounds of $[-1, 1]$, $clip(x, a, b)$ is defined as the function $\max(\min(x, b), a)$.

$$T_{dir}^s = clip(T_{dir}^s + \delta f b^s, -1, 1) \quad (20)$$

Indirect trust is calculated using (21). It finds the average recommended trust by the other node while scaling the impact based on the distance between the recommended context and the user's current context, $d(c, c_i)$. The impact of a recommended context, c_i , is made non-linear by passing the distance into the the function $\sigma_s = \sigma(z - s)$ and is weighted by a quality of context value QC_i . The reported trust, T_i^s , is then weighted by the reputation, R_i .

$$T_{ind}^s = \frac{\sum_{i \in \mathcal{R}} QC_i \sigma_s(-d(c, c_i)) R_i T_i^s}{\sum_{i \in \mathcal{R}} \sigma_s(-d(c, c_i))} \quad (21)$$

The quality of context of a node, i is computed using (22) by a different node that has computed trust and possibly performed a transaction. The calculation rewards deviation of the reported context c_i , from the aggregated context $\mathbf{C}_i$, as this shows the node is actively reporting for differing situations.

$$QC_i = \sigma_s(d(c_i, \mathbf{C}_i)) \quad (22)$$

Context aggregation and reputation updates are also performed at the same time as the quality of context on the same nodes i . Context aggregation is the most expensive computation of the model with complexity, $\mathcal{O}(\zeta \mathcal{N})$, linear to the number of connected nodes scaled by a system parameter. This system parameter, ζ , states how many reported contexts from previous epochs (inclusive of the current one) are to be stored in cs_i as defined in (23).

$$cs_i = \begin{bmatrix} \langle c_{i,t-\zeta} \rangle \\ \vdots \\ \langle c_{i,t} \rangle \end{bmatrix} \quad (23)$$

The aggregation of previously reported contexts has the purpose of providing a single context vector that informs a node's previously reported contexts. It is calculated based on a scaled average using (24), where θ is a system parameter indicating the importance of past contexts, $cs_{i,j}$ is the j th row of cs_i , $\odot$ denotes the Hadamard product, and $\oslash$ denotes Hadamard division.

$$\mathbf{C}_i = \sum_k \left[\left([\theta^{d(c_i, cs_{i,j})}]_j \odot cs_i^\top \right)_k \right]_k \oslash \sum_j \theta^{d(c_i, cs_{i,j})} \quad (24)$$

Reputation is calculated using (25) and is based on gradient descent. However, for this case, the reputation from the previous update, R_i' , is decayed according to the distance between the last allowed recommendation context, c_i^a , and the previous update's aggregated context, $\mathbf{C}_i'$. The gradient is the sum of the projections between i 's recommendation and the most recent recommendations of each of other connected nodes.

$$R_i = clip(R_i' \eta^{d(c_i^a, \mathbf{C}_i')} + \Delta_r \sum_{j \in \mathcal{N} \setminus \{i\}} T_i^s T_j^s, -1, 1) \quad (25)$$

When a node reports that the service provider is malicious, it is expected that the node will not interact with the service

TABLE IV
PARAMETERS OF OUR CONTEXT-BASED ATTACK MITIGATING TRUST MODEL

Parameter	Value
Number of nodes	100
Number of epochs	300
s	1
Initial trust	0.01
Number of contexts	4
η	0.95
Δ	0.01
Δ_r	0.02
τ	0.01
θ	0.7
ζ	50
Threshold for trust	0
Sigmoid function ($\sigma(\cdot)$)	Logistic
$\sigma_s(z)$	$1/(1 + \exp(s - z))$
Distance function ($d(\cdot, \cdot)$)	Euclidean

provider for a while. To prevent the flooding of malicious reports, recommendations after a negative recommendation are not accepted from a node i until (26) is satisfied. The timeout mechanism waits until the distance between the current context experienced, c , is sufficiently distant from the context the recommendation was received in, c_i^{recv} . The mechanism is also additively impacted by the extremity of the reported recommendation, T_i , and the distance between the reported context, c_i , and the context reported previous to that, c_i' .

$$\tau \stackrel{?}{=} \eta^{d(c, c_i^{recv}) + T_i^s \sigma_s(d(c_i, c_i'))} \quad (26)$$

B. Simulation Configuration

We evaluate the trust from the perspective of an observer node which never performs direct transactions with the service provider. For each epoch of the simulation, every node calculates the trust of the service provider, performs a transaction with it, updates each trust model value, and sends a recommendation composed of a tuple of the current context and direct trust, (c, T_{dir}^s) , to every other node.

The goal of the adversaries is to make a non-malicious server to appear as malicious to the observer. In the case of the context-based attack, they perform the two stage attack described in Section IV-F2, combined with the bad mouthing attack.

C. Results

The model successfully mitigates the context-based attack for all cases except where all the nodes are adversaries. We observe from Figs 8 and 9 the difference between the effects of the context attack when the observer is a member of the target group or the normal group. Fig. 10 shows how the system can mitigate recommendation based attacks, and that their effect on the system is similar to that of the context-based attacks. In each of the graphs, there are periodic reductions in the

Fig. 8. Trust of the target group observer under context-based attack

Fig. 10. Trust under the bad mouthing attack

Fig. 9. Trust of the normal group observer under context-based attack

calculated trust. This is caused by (26) evaluating to be true for the adversarial nodes and letting in their recommendations for one epoch.

VI. CONCLUSION

In this paper, we have presented a new type of context-based attack on context aware trust models for IoT systems. We have demonstrated the effectiveness of this new type of attack on six previously proposed trust models. The attacks

involve performing a classic recommendation-based attack for a target context and acting honest otherwise; the adversaries may additionally spoof the context to force favourable events in the system. It creates a targeted attack against the nodes that operate within these contexts. Hence the adversaries are able to single out particular target group and reduce the likelihood of their detection. When only a minority set of nodes in the trust management system is attacked while the other nodes are having normal interactions, it becomes difficult for the service provider to determine the cause, whether it is due to an attack or it arises from the context. We have also developed a trust management system that can mitigate the proposed context-based attack. It provides a mechanism to evaluate the quality of reported context independently from that of the reported trust. The system has additional features such as provides context summaries, constantly modulates indirect trust relative to its distance from the direct trust, and excludes repeated negative recommendations for time periods scaled by the reported context and trust. All these characteristics help to counteract the context-based attack and its combination with recommendation attacks.

REFERENCES

- [1] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," *IEEE Transactions on Sustainable Computing*, vol. 4, pp. 88–95, 2019.
- [2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, pp. 60–67, 2016.
- [3] C. Kolias, G. Kambourakis, A. Stavrou, and J. M. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, pp. 80–84, 2017.
- [4] Z. Lin and L. Dong, "Clarifying trust in social internet of things," *arXiv preprint arXiv:1704.03554*, 2017.

- [5] X. Wu and F. Li, "A multi-domain trust management model for supporting rfid applications of iot," *PloS one*, vol. 12, no. 7, p. e0181124, 2017.
- [6] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-aware recommender systems," *AI Magazine*, vol. 32, no. 3, pp. 67–80, 2011.
- [7] J. Guo, R. Chen, and J. J. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.
- [8] M. G. Uddin, M. Zulkernine, and S. I. Ahamed, "Cat: a context-aware trust model for open and dynamic systems," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 2024–2029.
- [9] M. Tavakolifard, S. J. Knapskog, and P. Herrmann, "Trust transferability among similar contexts," in *Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks*. ACM, 2008, pp. 91–97.
- [10] N. B. Truong, H. Lee, B. Askwith, and G. M. Lee, "Toward a trust evaluation mechanism in the social internet of things," *Sensors*, vol. 17, no. 6, p. 1346, 2017.
- [11] G. Yang, Q. Sun, A. Zhou, J. Li, X. Yuan, and H. Tang, "A context-aware trust prediction method based on behavioral data analysis in distributed network environments," in *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 2016, pp. 674–680.
- [12] A. Otebolaku and G. M. Lee, "A framework for exploiting internet of things for context-aware trust-based personalized services," *Mobile Information Systems*, vol. 2018, 2018.
- [13] B. Ye, Y. Wang, and L. Liu, "Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments," in *2015 IEEE International Conference on Web Services*. IEEE, 2015, pp. 121–128.
- [14] M. D. Alshehri and F. K. Hussain, "A comparative analysis of scalable and context-aware trust management approaches for internet of things," in *International conference on neural information processing*. Springer, 2015, pp. 596–605.
- [15] S. E. A. Rafey, A. Abdel-Hamid, and M. A. El-Nasr, "Cbstm-iot: Context-based social trust model for the internet of things," in *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*. IEEE, 2016, pp. 1–8.
- [16] R. Neisse, G. Steri, G. Baldini, E. Tragos, I. N. Fovino, and M. Borterman, "Dynamic context-aware scalable and trust-based iot security, privacy framework," *Chapter in Internet of Things Applications-From Research and Innovation to Market Deployment, IERC Cluster Book*, 2014.
- [17] K. Habib and W. Leister, "Context-aware authentication for the internet of things," in *Elev. Int. Conf. Auton. Syst. fined*, 2015, pp. 134–139.
- [18] L. L. Peterson and B. S. Davie, "8 - network security," in *Computer Networks (Fifth Edition)*, 5th ed., ser. The Morgan Kaufmann Series in Networking, L. L. Peterson and B. S. Davie, Eds. Boston: Morgan Kaufmann, 2012, pp. 633–695. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123850591000089>
- [19] X. Mao and J. McNair, "Effect of on/off misbehavior on overhearing based cooperation scheme for manet," *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, pp. 1086–1091, 2010.
- [20] I. S. I. Abuhaiba and H. B. Hubboub, "Reinforcement swap attack against directed diffusion in wireless sensor networks," *International Journal of Computer Network and Information Security*, vol. 5, pp. 13–24, 2012.
- [21] J. R. Douceur, "The sybil attack," in *IPTPS*, 2002.
- [22] Y. B. Saied, A. Oliveureau, D. Zeglache, and M. Laurent, "Trust management system design for the internet of things: A context-aware and multi-service approach," *Computers & Security*, vol. 39, pp. 351–365, 2013.
- [23] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Ca4iot: Context awareness for internet of things," in *2012 IEEE International Conference on Green Computing and Communications*. IEEE, 2012, pp. 775–782.
- [24] S. Toivonen, G. Lenzini, and I. Uusitalo, "Context-aware trust evaluation functions for dynamic reconfigurable systems," *MTW*, vol. 190, 2006.
- [25] N. Li, V. Varadharajan, and S. Nepal, "Context-aware trust management system for iot applications with multiple domains," in *ICDCS*. IEEE, 2019.