Sensor Network Environments

Subjects: Engineering, Electrical & Electronic | Computer Science, Hardware & Architecture
Contributor: Kealan Mannix, Aengus Gorey, Donna O'Shea, Thomas Newe

New technologies have driven the rise of what is being termed as the fourth industrial revolution. The introduction of this new revolution is amalgamating the cyber and physical worlds, bringing with it many benefits, such as the advent of industry 4.0, the internet of things, cloud technologies and smart homes and cities. These new and exciting areas are poised to have significant advantages for society; they can increase the efficiency of many systems and increase the quality of life of people. However, these emerging technologies can potentially have downsides, if used incorrectly or maliciously by bad entities. The rise of the widespread use of sensor networks to allow the mentioned systems to function has brought with it many security vulnerabilities that conventional "hard security" measures cannot mitigate. It is for this reason that a new "soft security" approach is being taken in conjunction with the conventional security means. Trust models offer an efficient way of mitigating the threats posed by malicious entities in networks that conventional security methods may not be able to combat.

Keywords: trust ; trust model ; security ; networks ; IoT ; IIoT ; Industry 4.0 ; Agri 4.0

1. Introduction

Trust models are increasingly being used in a wide range of sensor networks, such as the IoT (internet of things), VANETs (vehicular ad hoc networks), cyber-physical manufacturing and a wide range of wireless sensor networks. They are not just limited to the sensor network field, but are also applicable to many network topologies, such as online P2P (peer too peer) networks [1] and web services [2]. However, this entry focuses mainly on sensor and IoT networks. The reason for the necessity of this new security measure is the fact that these networks are vulnerable to internal attacks, such as bad-mouthing (an attack where a malicious node falsely reports indirect trust values of another node), collusion, on-off and Sybil, to name just a few. These attacks can pose serious threats to the network operation if not dealt with in a timely manner, which conventional security is not capable of doing.

Conventional security measures such as encryption, digital signatures and authentication measures (such as passwords, tokens or biometrics) are all aimed at protecting the "CIA triad (modified)" (Confidentiality, Integrity and Authenticity (modified triad as in the classical CIA Triad; this is Availability)). The concept of the triad security is aimed at protecting from external attackers, who should not have access to the system or network. However, many of these new network topologies, such as VANETs and IoT networks, are operating on the basis that nodes may enter and leave the network with relative ease, allowing the nodes to send and receive data within the network. In the case of WSNs (wireless sensor networks), many of the nodes will not have high computing capabilities and can be quite vulnerable to cyber-attacks. Due to the potential openness of their deployed environment and transmission medium, the nodes could also be vulnerable to physical attacks, such as tampering or hijacking attacks [3]. All these attacks are internal to the network and cannot be conveniently mitigated by conventional means of security, such as cryptography and authentication.

Trust models are a simple concept used to protect against the aforementioned internal attacks. Trust models are simply a method of interpreting the data provided by nodes in the network to determine if they are trustworthy or not. This concept is similar to, and based on, how people establish trust in their everyday lives. Humans are able to interpret large amounts of data from the world around them and use these data to make decisions, such as whether a person be trusted. For example, if you met a stranger whom you just witnessed stealing something from a shop, you would be much less likely to trust something they said than if you met a friend you had known for years, and who has never, to your knowledge, done anything bad. It is from these simple concepts that trust can also be developed in networks. Nodes that have had good previous interactions with one another will afford one another a higher level of trust than they will afford a new "stranger" node that appears to be sending peculiar data in the network.

The data used to calculate trust can be divided into three main categories: direct trust, indirect trust and physical data. These categories are also what people use in their assessment of trust, and can be thought of in a synonymous way

when applied to the cyber world. The three categories are explained very briefly below, along with analogies of how people tend to use them in their trust calculations of another person.

Direct trust: Your own personal, direct experience with the entity—are they a friend you have known for a while?

Indirect trust: Another entity's recommendation of the entity—do you have a mutual friend? What are their thoughts on them?

Physical data: Observations made of the entity's behaviors or attributes—have you witnessed them misbehaving? What is their demeanor like?

The basic concept of a trust model is simple and familiar; however, there are many different ways of implementing a model and many contrasting use cases in which trust models can be used. This entry aims to provide a comprehensive survey of the various trust models used in sensor networks and to provide an outline of how these models work, along with their strengths and weaknesses. Other reviews on trust models also provide this information; however, they lack an analysis of the measurement parameters of trust. Other works also do not provide a view on how the environment and network types can affect the design and operation of the trust model. This work aims to contribute to these lacking areas by discussing the parameters seen to measure trust in many models and conducting a design study of a trust model on two different industry 4.0-type networks.

2. Trust Model Concepts

Trust models differ vastly from model to model depending on many factors, such as environment, use cases and the level of security needed. However, many key concepts remain the same for all models. In this section, a general overview of the trust models is discussed, including the network types they are being used in, the security features they aim to provide and why these features cannot be provided by conventional techniques. Lastly, the general structure of a trust model is considered.

2.1. Environments

While trust models can be deployed in a number of areas, this entry focuses on models used in sensor networks or similar network types, where the nodes of the network are communicating data back to a central system. There are many types of network topologies and attributes. Featured below is a list of some of the main ones, with a brief definition of each. Networks are not limited to just one of these attributes and are often a combination of a few. For example, you could have a heterogeneous, static, clustered network.

- Homogeneous: a network where all nodes are identical—same function, device, rank, etc.
- Heterogeneous: a network where nodes are not all identical.
- Hierarchical/clustered: networks where some nodes have a higher rank than others. Often, the network is controlled by a base station, which is in command of lower-level nodes or in a clustered network cluster head. These cluster heads are each in command of their own cluster of edge nodes. It is usually the case that the higher ranked the node is, the more computing power and resources it has.
- Static: a network where nodes are neither leaving nor entering—e.g., a smart factory sensor network usually has the same sensors all the time. Note: sometimes static can be referred to in terms of the nodes' physical positioning. In this entry, static refers to the above definition unless explicitly stated otherwise.
- Dynamic: a network where nodes are entering and leaving the network, e.g., a mobile network or VANETs. Note: sometimes dynamic can be referred to in terms of the nodes' physical positioning. In this entry, dynamic refers to the above definition unless explicitly stated otherwise.

Each network type will have different properties and challenges and will require slightly contrasting methods of determining the trust of its nodes. What follows is a description of some of the common types of networks that trust models are currently being used in.

IoT/CPS: Internet of things and cyber-physical systems are similar in terms of their network structures. They are a rapidly advancing area, which can be applied to many different use cases, such as smart homes or cities, smart medical equipment/monitors and industry 4.0. Internet of things networks are, in general, a heterogeneous network comprised of

many different sensors and controllable devices, all communicating with each other across a network. Trust models are applied to IoT-type networks in [4][5][6][7][8][9][10][11][12]. The challenges IoT networks tend to face are very common among most of the network environments. These challenges include potentially low computational power and low power resources (if battery-operated) for the network nodes, as well as the deployment of network nodes in a wide array of areas, which can leave some nodes vulnerable to physical compromise. Nodes within the network can be corrupted due to cyber or physical means, introducing internal attack threats to the network which cannot be effectively diminished by conventional security.

Mobile crowd sensing: [13][14] A subset of IoT applications, mobile crowd sensing is where large groups are used to collect data on some type of metric, such as traffic or noise. Many mobile phone apps use crowd sensing to collect data, such as google maps, which collects data on your location and speed to provide information to other users on traffic flow and journey times. Crowd-sensing applications use trust models to determine whether the data they get are trustworthy and useful or not, as in [14].

Wireless sensor networks: WSNs [3][7][8][15][16][17][18][19][20][21][22][23][24] are perhaps one of the most common use cases for trust models. This is due to the fact that sensor nodes, more often than not, have very limited computing capabilities. Trust models can be developed in very lightweight manners to account for this. Their wide range of deployment and low computing capabilities make them vulnerable to being compromised in a similar manner to the previously mentioned IoT devices. Sensor networks often have a central node. Trust calculations for the sensor nodes are often performed in this central node or base station, which has more power and resources available to it. This type of topology is well-suited to a trust model, and some very good results can be achieved by introducing a trust model to the network. There are many sub-types of WSNs, including homogeneous and heterogeneous, underwater acoustic networks [18][23], large scale networks [24], hierarchical networks [7] and cluster-based networks [3][8]. Each type of network will have slightly different requirements and topologies. Specific trust models can sometimes be used in more than one specific type of WSN, if they incorporate features that allow for flexibility in their implementation.

VANETs: Vehicular ad hoc networks [25][26][27] are comprised of vehicles, such as cars, trucks, buses, etc., which accept and receive data from one another and from roadside units (RSUs). They are used in smart cities for applications such as traffic monitoring and optimizing, accident prevention and early warning systems. These networks are much more dynamic, with vehicle nodes moving in and out of the network freely and the nodes not having a fixed point, which, of course, brings with it potentially malicious nodes entering and leaving freely. With these dynamic networks, some unique aspects can be added to their trust models, such as distance-checking as a plausibility check on the data [25][26], among others.

Unmanned aerial systems: This area is quite similar to a VANET. A trust model is applied to one in [28]. These systems can be thought of a cross between a VANET and IoT-type network, with the nodes, in this case, being static in the network but dynamic in terms of their positions, as they are traveling.

Cyber-physical manufacturing: [29][30] This type of network can be defined as a specific use case of the IoT or WSNs and is comprised of sensors and devices used to control critical infrastructure in factories. Often, these networks are more secure from the physical attacks of nodes, with only authorized personnel allowed enter the factory/manufacturing area and no new nodes allowed to enter the network without authorization. However, the nodes are still at risk of being compromised, and trust models offer an effective lightweight solution suitable to this environment.

2.2. Security Features

Conventional security focuses on providing hard security to a system based on the CIA triad (modified)—Confidentiality, Integrity, and Authenticity (as well as availability and access control). The trust model aims to provide most of these security features both internally and externally on the network.

Conventional security techniques include methods such as encryption, digital signatures, digital certificates and authentication methods (passwords, tokens, biometrics). All these methods mostly assume and are tailored according to the premise that the attacker is an outside entity, as is the case in attacks such as eavesdropping, DDoS or man-in-the-middle (MitM)-type attacks. These methods establish connections between users and assume trust thereafter for the duration of the session. However, in the new network environment types, this is often not good enough, and a new ‘zero trust’ approach must be taken, as nodes within the network can be compromised or gain access legitimately before launching their attacks, effectively launching them from inside the conventional security barriers.

Conventional security methods can still be used within the trust models to provide services such as confidentiality, integrity and authentication, although they must often be adapted to suit low-powered nodes. This is conducted through lightweight or edge cryptography [31][32] and key management [32] methods.

Trust models use the concept of zero trust, in which no entity is trusted until verified. The trust is dynamically updated. This is usually achieved through time-triggered updating or event-triggered updating. This dynamic updating aims to prevent nodes or entities that were once trusted but then became malicious from launching attacks internally on the network.

The main security contribution of a trust model is a form of access control. All trust models should be able to provide and deny access to any node on the network. Depending on the use case scenario, models will decide whether a node can be trusted, whether the data can be trusted and used, or whether the node should be allowed to access some form of resources on the network. Models must be able to discard any untrusted data and eject any untrusted or malicious nodes. This access control provided by the models will successfully mitigate attacks, such as packet modification, bogus data injection or malicious software spreading, which can be carried out through a number of different initial attack methods.

Trust models can also provide authentication if the situation requires. This can be achieved by means of signatures, as seen in [7], where an intrusion detection service provides a signature-based authentication method. As seen in [4], a hardware approach can be taken for the authentication of nodes with the use of radio frequency distinct native attribute (RF-DNA) fingerprinting. This method analyzes the transmitter signal of the node, comparing it to known samples from previous communications. In [4], these data are classified using an SVM (support vector machine). This method is similar to human biometrics (as if the transmitter of the device is the biometric) and definitely has great potential to be used in static hierarchical or clustered network environments, where there is a cluster head or base station with the capabilities of running the required hardware and software.

2.3. Trust Model Structure

For the purposes of this analysis of trust models, a high-level block diagram structure of a trust model was contrived, as seen in **Figure 1**. It is important to note that this is not based on any particular model and by no means defines all trust models. Many models loosely follow this structure, potentially leaving out or adding in components to the structure to suit their specific applications. This high-level block diagram includes the most common and pivotal features of trust models and aims to provide an understanding as to how a trust model typically operates. What follows is a description of each section numbered in the diagram.

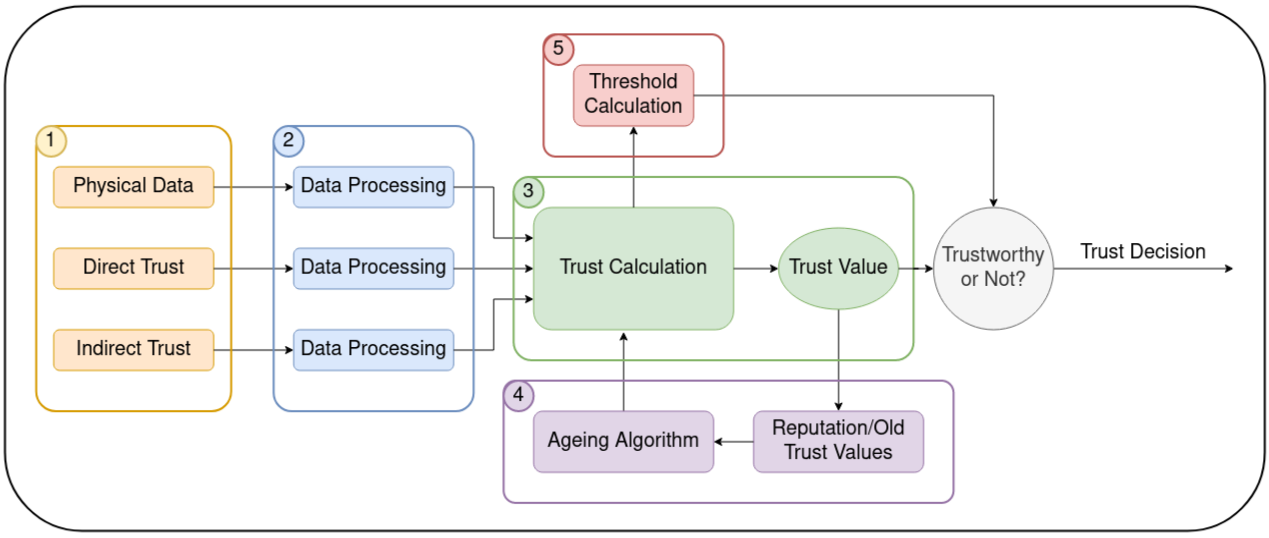

Figure 1. Diagram of a trust model's basic structure.

(1) Data gathering. The data gathering phase is essential to every trust model, as there must be data in order to calculate trust. The data gathering phase mainly collects data from three groups (just one or a combination of multiple groups can be used). These groups are shown in **Figure 1**.

Physical data include data such as energy and resource consumption of the device, the device status (software/OS version) and packet characteristics. Essentially, this includes any physical data about the device. These data can be used to detect odd behavior or anomalies in the device and can also be further used to determine direct and indirect trust.

Direct trust refers to data obtained from interactions between the trustee and trustor. This type of data can be obtained through monitoring the communication behavior of, or the packets being sent by, a device, or through other physical data that is obtained, or by another method of defining whether the interaction was successful or not. It is often determined through probabilistic means, based on the communication behavior of the device.

Indirect or neighbor trust is essentially direct trust obtained from another source. For example, if node A wishes to communicate with node B, node B can calculate its own direct trust with A, but also ask nodes C and D for the direct trust values between themselves and A. B can then take this value into account when calculating its trust of A. Usually, the indirect trust is accumulated from all nearby neighbors and summed or averaged in some of them, and this is taken into account with less weight of effect than the direct trust values.

(2) Data processing. In some models, the data are pre-processed before being used in the trust calculation. In these types of models, the data are pre-processed using a mathematical model individually, and the trust calculation acts as a form of integrating these values together, such as a weighted average or sum. This is not a necessary step; however, it can be useful in some instances. For example, direct trust can be calculated using physical data or potentially by considering previous trust values or more than one parameter. Multiple indirect trust values are obtained from neighbors and need to be amalgamated. These calculations can take place before the trust calculation, if necessary, depending on the model being used.

(3) Trust calculation. This phase is at the heart of the trust model. It is where all the data obtained are converted to a meaningful trust value to determine whether the device/node is malicious or not. This calculation can be performed in numerous ways, using different mathematical models. From the block diagram in **Figure 1**, it can be seen that all the data gathered are used in this phase, and previous trust values can also be fed back and used in this phase. The trust value that is calculated in this phase is updated regularly, and this can be accomplished through time- or event-triggered updating, depending on the model.

(4) Historical trust values. Many trust models, but not all models, take into account historical trust values when calculating their trust. These previous values are stored and inserted into the trust calculation, usually using an ageing algorithm that can be anything from a weighted average to something more complex. These historic trust values are often referred to as the reputation of the node. This step may not be utilized in all models, and the direction arrows in **Figure 1** show this.

(5) Threshold calculation. An area that is often lacking in most models is dynamic threshold calculation. The threshold level in most models decides whether a node's trust value renders it malicious or not. Dynamically calculating the threshold gives the model more flexibility to be applied in more than one specific network, as it will account for differences in network traffic, the number of nodes or other factors that could affect the average value of trust for a node. The majority of models tend to leave this phase out and instead employ a static threshold value method [33]. Dynamic threshold calculation should, however, be considered for the models, as it provides more flexibility in allowing them to be applied in more than a single specific network.

The final stage in the diagram is the simple comparison of the trust value with the threshold value. This stage determines the final decision on the node's trust status. If a node becomes ejected from the network, some models have methods that allow these nodes to build up reputation again over time without having full access to the network and, later, to potentially rejoin the network. Models also sometimes face the 'cold start' issue, whereby trust calculations that rely on data and previous values do not work initially on startup, which must be considered. These points are discussed further below.

3. Attack Types

There are numerous attack types that can be carried out against sensor or IoT networks. Many of these attacks are often launched individually, but they can be used in conjunction with one another. Some attacks are designed specifically to gain access to the network or boost (or diminish) the trust of a given node, while others are designed to spread malware, disrupt routing and drop the critical information in transit. What follows is an overview of the attack types that are typically found in the network-type environments and that trust models are being used to mitigate. The attacks have been classified into six broad categories: access, reputation, payload/active, denial of service (DoS), routing and physical. Often, the access- or reputation-type (and sometimes the physical-type) attacks are used to gain privileges in the network that allow the payload, DoS or routing attacks to be run, which can in turn cause network disruption. A lot of these attack types are carried out by compromised nodes, which can become so through access attacks or device hacking. A review of the methods by which nodes are compromised is beyond the scope of this entry.

3.1. Access Attacks

Counterfeiting. Also known as impersonation, spoofing or identity fraud attacks, counterfeiting involves inserting a malicious node into a network that will pretend to be a legitimate node that is already in the network. It can be carried out using a new node with a fake ID that is inserted into the network, or by changing the ID of a compromised node already in the network to make it appear as a different node. This can be achieved when a relaying node steals the ID of a node from relayed data packets and uses this on its own packets [34]. These attacks can be used to gain access to networks and launch further attacks and, in some cases, divert the mitigation actions of the trust model back to the node that is being counterfeited, causing the legitimate node to be ejected. The authors of [34] propose a detection method for impersonation attacks based on bloom filters and the use of hashes for WSNs.

Man-in-the-middle (MitM). One of the more common cyber-attacks in all areas, the MitM attack involves an attacker injecting itself into an ongoing communication between two legitimate parties [35]. The attacker becomes an intermediary in the communication, whereby both legitimate parties still believe they are communicating directly with one another, but, in fact, they are unknowingly communicating with one another through the MitM. MitM attacks can be passive in nature, simply eavesdropping or running a traffic analysis, although often they are used to launch further attacks, such as those in the payload or DoS categories.

3.2. Reputation Attacks

These attacks specifically affect trust models. They are quite similar in terms of the goals they wish to achieve. Different reputation-type attacks are often used interchangeably or in conjunction with one another to attack vulnerabilities in a given trust model.

Breakout fraud/on-off. Sometimes referred to as a zigzag [25] or garnished attack [17][24], this type of attack involves a node behaving as expected by the network for a period of time. Then, when its trust values are high enough, it will launch an attack. When the model detects suspicious behavior from the node and the trust value begins to fall, the node can stop behaving maliciously and return to normal behavior, which in turn will restore the trust value before the network ejection is initiated. This process can be repeated multiple times, as long as the node does not let its trust value drop below the threshold value. Attacks such as this demonstrate the need for a dynamic and fast trust-updating response to catch attacks early, before they disappear again for a period of time.

Bad-mouthing. Bad-mouthing involves a malicious node falsely reporting the indirect trust values of another node. This will negatively affect its trust value. Collusion or sybil attacks can be used to boost the effectiveness of this attack. Some models use methods where only positive trust values can be reported back to counteract bad-mouthing attacks [15].

Ballot-stuffing. As it is in the physical world, ballot-stuffing involves one user stuffing the ballot with the same vote to change the results of a vote. This scenario is the same in the digital world, where a node can stuff the ballot in a majority voting system or an indirect trust system. This can affect its own or other nodes' trust values. This attack can be achieved through means of sybil, or collusion, and this technique can also be used for self-promoting and bad-mouthing attacks in models that are vulnerable to it.

Collusion. This attack can be considered as a reputation-based attack or a routing attack (or it can also be used for forms of DDoS attacks). In this attack type, multiple attackers work in collusion to modify packets, drop routing packets [36], perform bad-mouthing or self-promoting attacks or compromise majority voting schemes. It is a very broad class of attack that essentially describes any attack where there are multiple attack nodes working together to achieve a common goal.

Sybil. In this scenario, a node presents multiple identities to other nodes in the network [37]. These identities can be forged or stolen from other nodes. This allows the node to have a greater say in the network and can be used to boost bad-mouthing or self-promoting attacks. It can also be used to corrupt data, majority voting schemes and other processes in the network.

Self-promoting. This is an attack where the node simply reports better trust values for itself or another node that it is working with. This attack aims to boost the trust value of the attacker, giving it more access to launch further attacks. This attack can be launched using sybil or collusion attacks, as mentioned above.

3.3. Payload/Active Attacks

These types of attacks are launched from compromised nodes after access is gained and trust is established. These types of attacks can be very destructive to a network and the goal of the trust models would be to detect and eject

compromised nodes before they can gain enough power to launch such attacks effectively.

Malicious software spreading. This is the spreading of software such as worms or viruses through a network from a compromised node. These worms and viruses can compromise more nodes and propagate throughout the network, affecting the operation of compromised nodes in any way that the attacker intended when creating the malware. These attacks can be devastating to a network.

Packet modification. In this scenario, a compromised node modifies all or some of the packets that it is supposed to forward [38]. This allows the attacker to insert the data it wants into the packets, such as false data readings or malware. Data pollution attacks are also very similar to packet modification and involve the injection of corrupted data into the network to disrupt normal operation.

Selective forwarding. Otherwise known as packet dropping, this is where a compromised node drops all or some of the packets that it is supposed to forward [38]. This causes data to be delayed in transit or lost completely. Nodes can also store packets and forward them later in a replay attack. A replay attack can be mitigated using random numbers or timestamps.

4. Trust Data Gathering

The information in this section relates to phases one and two of **Figure 1**. This section analyzes the methods of gathering data and the parameters measured by the models to be used in trust calculation.

4.1. Data Gathering

Direct gathering. Almost all models use some form of direct trust. Direct trust is any trust metric gathered directly between trustor and trustee, without the use of a third-party node or entity. This can include analyzing the data sent in the communications, i.e., packets, communication frequency and the quality of the data [14], or using a system similar to a watchdog mechanism. In [15], a watchdog mechanism is used to monitor the routing, processing and data in nodes that are in direct communication with a given node. Many trust parameters can be gathered directly through this direct gathering and used in trust calculations later on, or to calculate a direct trust score.

Indirect gathering. The opposite of direct gathering, indirect gathering is the method of using third-party nodes or entities to gather data and use these data to calculate indirect trust scores (also referred to as second-hand, recommended or reputation trust scores). This is a very common method of gathering data. Methods for selecting the nodes to gather this data include one-hop neighbors [11][15][33], nodes in the cluster [8][9][19][24][26], cluster head or base station feedback [17], multi-hop neighbors [21] and common neighbors between the trustor and trustee [20].

Broadcasting. This is a method in which a node or cluster head broadcasts information. In [16], all communications are broadcasted from the beacon nodes so that other beacons can listen and check if the location information is correct when compared to their location table. In [30], the gateway information of a provider is broadcast and opened up for auction to the users, and, in [33], trust tables are multi-cast (it is multicast as it is not broadcasting to the entire network, but only its cluster) from the cluster head to its cluster after a table update has taken place. The broadcasting method removes a layer of privacy from the data being broadcast and prevents attacking nodes from fooling individual nodes, as all the nodes are able to check if a malicious node broadcasts inaccurate information relative to the other nodes in the group.

4.2. Trust Parameters

Communications are a vital part of any network and many attacks. Attacks can often change communication behaviors in a network; therefore, it makes sense to use communications as a parameter for measuring trust. Many models use some attribute of the communications between nodes to help to model trust. The message frequency and packet forwarding rate [4][12][27], as well as the packet forwarding behavior [19], are simple ways of determining if a node is behaving unusually. Another common approach is to analyze the ratio of successful interactions/packets sent, or that of unsuccessful or total interactions/packets sent. This type of communications trust is used widely (reference [3] terms it as honesty trust) [9][17][18][20][22][24][26]. This approach requires a method of defining a successful or unsuccessful transmission, which is usually achieved using a simple set of rules defined for the given context. Each model will differ in how it defines successful or unsuccessful transmissions, but a common approach is to use an ACK signal to indicate the received packets. The communication trust score is calculated using simple ratios, equations or probabilities to convert the data collected into a value that can later be used in the trust calculation. The method used to achieve this is slightly different for each model so as to suit its contextual needs.

Data can be attacked through some of the attack methods mentioned in Section 4, and these attacks can modify and change data being sent by nodes in a detectable manner. Data checking can often be performed in simple yet effective manners, adding little computational overhead to the network. The authors of [3][15][19][20][23][39] use simple anomaly-type checks, such as comparing sensor data to the average value of the data from all sensors. If the data is flawed, then there may be a potential compromise. The authors of [25][26] use plausibility checks on the data, both operating on a VANET network and employing methods for determining vehicles' distance from the roadside unit that they are in communications with. If the data indicates that it is being sent from a location outside this range, these checks will catch it and instantly discard the information. These types of checks seem trivial, but they are very effective and add little computational overhead to a model and, therefore, it makes sense to include them.

Other models use methods such as analyzing the expected values of the amount of data being collected [12], analyzing temporal [10] or spatial [18] correlations of the sensor data, and detecting if a packet is malicious based on its signature [7] or successful data interactions [24].

In [14], a quality of data measurement is used to develop a QoD score for nodes in a crowd-sensing network, and this value influences the trust value of the node. There are eight quality dimensions used to attain this QoD score. Six of these data qualities are defined in [40] as accuracy, completeness, reliability, consistency, uniqueness and currency. The remaining two are defined as syntactic accuracy and semantic accuracy, in the case of live sensor data. This method of QoD assessment works well for applications such as crowd sensing but could also be applicable to other applications, such as some IoT or WSNs.

Interaction history. Often, nodes will have had previous interactions with each other. Just as people develop (or lose) trust in each other through more interactions, nodes in a network can too. This can be achieved through counting the number of previous successful or unsuccessful interactions between nodes, and by allowing nodes to be more trusting with entities they have a good history with and more cautious with entities they have bad or no history with. The authors of [3][5][26][27] all consider the number of previous interactions that nodes had with each other. This method is similar and overlaps with the communication method discussed above, which analyzes the ratio of successful to unsuccessful/total packets sent to a node, which is also using information about past interactions to help to measure trust.

Resource consumption can be a clear indicator of a DoS, routing, payload or other type of attack. If a node is consuming an abnormal amount of energy, CPU power or bandwidth, then it is most likely doing something it should not be, such as running CPU intensive processes, downloading programs or running far more communications than usual. Numerous models in the aforementioned communications section can detect increased bandwidth usage by monitoring the communications.

CPU power is monitored in [15] using a watchdog mechanism that allows nodes to observe their neighbors' processing consumption. CPU, network and storage trust is used to help to define trust for service composition in mobile cloud environments in [13] and in the IoT domain in [5].

The authors of [18][20][22][23] all use an energy trust value based on the energy consumption rate of the node. It is important to note that, in these networks (WSNs and underwater WSNs), energy consumption is very important, as the sensors are in difficult-to-access, remote locations with limited battery life, and excessive energy usage due to attacks or malfunction can cause the sensors to be out of action for a prolonged period of time. Therefore, it makes sense for networks such as these to use energy usage as a trust-measuring metric, as energy is such a valuable resource for them.

Device behavior is examined in [5]. The behavior concerns the device resource usage metrics and whether they are within a predetermined range for the device. This information is obtained from MUD (manufacturer usage description) files. MUD files consist of an MUD URL, where the network can find information about the device connecting to it from the MUD file server [41]. This information notifies the network about what this physical device does, and the network can take appropriate access control measures and can get an accurate idea of its normal resource usage for anomaly detection.

Device status encompasses information regarding device integrity and device resilience [5]. The integrity of the device is defined in [5] as the extent to which the device is known to run legitimate firmware/OS/software in valid configurations. The resilience of the device is defined as the known security of its firmware/OS/software versions. The authors of [5] apply this metric to the trust calculation of the device. The device most likely will not change its firmware/OS very frequently. Therefore, this metric may be more useful in helping to calculate an initial trust value for a new node joining a network when there are little to no other available data on the node.

Associated risk is not commonly used in trust models. It is used in [5], where the risk associated with a device is classed as a fuzzy level. The risk is calculated based on the device's probability of being compromised and the damage it could do to the network (based on its access rights and perceived value and also its neighbor's criticality and perceived value in the network). This method involves a great deal of work in classifying the nodes' associated risk accurately, and it may not be useful in many network types (e.g., homogeneous networks). However, in more complex networks, it may be useful for implementing different classes of trust for devices with different priorities and capabilities within the network in order to increase productivity. A similar concept is used in [25], where a role-orientated trust system is used. In the VANET, different weights are given to the information received from normal vehicles, high-authority vehicles (emergency vehicles), public transport, professional vehicles and traditional vehicles (vehicles with little to no travel history, which are assigned a lower weight).

References

1. Yousuf, M.; Kim, S. Coping with bad-mouthing in peer-to-peer file sharing networks. In Proceedings of the 2015 IEEE International Conference on Peer-to-Peer Computing (P2P), Boston, MA, USA, 21–25 September 2015; pp. 1–9.
2. Dragoni, N. A survey on trust-based web service provision approaches. In Proceedings of the 3rd International Conference on Dependability (DEPEND), ACM, Venice, Italy, 18–25 July 2010; pp. 83–91.
3. Zhang, Z.; Zhu, H.; Luo, S.; Xin, Y.; Liu, X. Intrusion Detection Based on State Context and Hierarchical Trust in Wireless Sensor Networks. IEEE Access 2017, 5, 12088–12102.
4. Kandah, F.; Cancelleri, J.; Reising, D.; Altarawneh, A.; Skjellum, A. A Hardware-Software Codesign Approach to Identity, Trust, and Resilience for IoT/CPS at Scale. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 1125–1134.
5. Mathas, C.; Vassilakis, C.; Kolokotronis, N. A Trust Management System for the IoT domain. In Proceedings of the 2020 IEEE World Congress on Services (SERVICES), Los Alamitos, CA, USA, 18–24 October 2020; pp. 183–188.
6. Wang, Y. Trust Quantification for Networked Cyber-Physical Systems. IEEE Internet Things J. 2018, 5, 2055–2070.
7. Meng, W.; Li, W.; Su, C.; Zhou, J.; Lu, R. Enhancing Trust Management for Wireless Intrusion Detection via Traffic Sampling in the Era of Big Data. IEEE Access 2018, 6, 7234–7243.
8. Rani, R.; Kumar, S.; Dohare, U. Trust Evaluation for Light Weight Security in Sensor Enabled Internet of Things: Game Theory Oriented Approach. IEEE Internet Things J. 2019, 6, 8421–8432.
9. Boudagdigue, C.; Benslimane, A.; Kobbane, A.; Liu, J. Trust Management in Industrial Internet of Things. IEEE Trans. Inf. Forensics Secur. 2020, 15, 3667–3682.
10. Karmakar, G.; Das, R.; Kamruzzaman, J. IoT Sensor Numerical Data Trust Model Using Temporal Correlation. IEEE Internet Things J. 2020, 7, 2573–2581.
11. Adewuyi, A.; Cheng, H.; Shi, Q.; Cao, J.Á. MacDermott and X. Wang, "CTRUST: A Dynamic Trust Model for Collaborative Applications in the Internet of Things. IEEE Internet Things J. 2019, 6, 5432–5445.
12. Wang, T.; Luo, H.; Jia, W.; Liu, A.; Xie, M. MTES: An Intelligent Trust Evaluation Scheme in Sensor-Cloud-Enabled Industrial Internet of Things. IEEE Trans. Ind. Inform. 2020, 16, 2054–2062.
13. Li, W.; Cao, J.; Hu, K.; Xu, J.; Buyya, R. A Trust-Based Agent Learning Model for Service Composition in Mobile Cloud Computing Environments. IEEE Access 2019, 7, 34207–34226.
14. Truong, N.; Lee, G.; Um, T.; Mackay, M. Trust Evaluation Mechanism for User Recruitment in Mobile Crowd-Sensing in the Internet of Things. IEEE Trans. Inf. Forensics Secur. 2019, 14, 2705–2719.
15. Ganeriwal, S.; Srivastava, M. Reputation-based Framework for High Integrity Sensor Networks. In Proceedings of the 2nd ACM Workshop on Security of Ad-hoc and Sensor Networks, Washington, DC, USA, 25 October 2004.
16. Srinivasan, A.; Teitelbaum, J.; Wu, J. DRBTS: Distributed Reputation-based Beacon Trust System. In Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, DASC, Indianapolis, IN, USA, 29 September–1 October 2006.
17. Li, X.; Zhou, F.; Du, J. LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks. IEEE Trans. Inf. Forensics Secur. 2013, 8, 924–935.
18. Han, G.; He, Y.; Jiang, J.; Wang, N.; Guizani, M.; Ansere, J. A Synergetic Trust Model Based on SVM in Underwater Acoustic Sensor Networks. IEEE Trans. Veh. Technol. 2019, 68, 11239–11247.

19. Reddy, V.B.; Venkataraman, S.; Negi, A. Communication and Data Trust for Wireless Sensor Networks Using D–S Theory. IEEE Sens. J. 2017, 17, 3921–3929.
20. Wu, X.; Huang, J.; Ling, J.; Shu, L. BLTM: Beta and LQI Based Trust Model for Wireless Sensor Networks. IEEE Access 2019, 7, 43679–43690.
21. Desai, S.; Nene, M. Multihop Trust Evaluation Using Memory Integrity in Wireless Sensor Networks. IEEE Trans. Inf. Forensics Secur. 2021, 16, 4092–4100.
22. Pang, B.; Teng, Z.; Sun, H.; Du, C.; Li, M.; Zhu, W. A Malicious Node Detection Strategy Based on Fuzzy Trust Model and the ABC Algorithm in Wireless Sensor Network. IEEE Wirel. Commun. Lett. 2021, 10, 1613–1617.
23. Jiang, J.; Zhu, X.; Han, G.; Guizani, M.; Shu, L. A Dynamic Trust Evaluation and Update Mechanism Based on C4.5 Decision Tree in Underwater Wireless Sensor Networks. IEEE Trans. Veh. Technol. 2020, 69, 9031–9040.
24. Khan, T. A Novel and Comprehensive Trust Estimation Clustering Based Approach for Large Scale Wireless Sensor Networks. IEEE Access 2019, 7, 58221–58240.
25. Ahmad, F.; Kurugollu, F.; Kerrache, C.; Sezer, S.; Liu, L. NOTRINO: A NOvel Hybrid TRust Management Scheme for INternet-of-Vehicles. IEEE Trans. Veh. Technol. 2021, 70, 9244–9257.
26. Ahmad, F.; Kurugollu, F.; Adnane, A.; Hussain, R.; Hussain, F. MARINE: Man-in-the-Middle Attack Resistant Trust Model in Connected Vehicles. IEEE Internet Things J. 2020, 7, 3310–3322.
27. Xia, H.; Zhang, S.; Li, Y.; Pan, Z.; Peng, X.; Cheng, X. An Attack-Resistant Trust Inference Model for Securing Routing in Vehicular Ad Hoc Networks. IEEE Trans. Veh. Technol. 2019, 68, 7108–7120.
28. Keshavarz, M.; Gharib, M.; Afghah, F.; Ashdown, J. UASTrustChain: A Decentralized Blockchain- Based Trust Monitoring Framework for Autonomous Unmanned Aerial Systems. IEEE Access 2020, 8, 226074–226088.
29. Yu, Z.; Zhou, L.; Ma, Z.; El-Meligy, M. Trustworthiness Modeling and Analysis of Cyber-physical Manufacturing Systems. IEEE Access 2017, 5, 26076–26085.
30. Jeong, S.; Na, W.; Kim, J.; Cho, S. Internet of Things for Smart Manufacturing System: Trust Issues in Resource Allocation. IEEE Internet Things J. 2018, 5, 4418–4427.
31. Maimut, D.; Ouafi, K. Lightweight Cryptography for RFID Tags. IEEE Secur. Priv. 2012, 10, 76–79.
32. Latif, M.; Ahmad, M.; Khan, M. A Review on Key Management and Lightweight Cryptography for IoT. In Proceedings of the 2020 Global Conference on Wireless and Optical Technologies (GCWOT), Malaga, Spain, 6–8 October 2020; pp. 1–7.
33. Chang, B.; Kuo, S. Markov Chain Trust Model for Trust-Value Analysis and Key Management in Distributed Multicast MANETs. IEEE Trans. Veh. Technol. 2009, 58, 1846–1863.
34. Tanabe, N.; Kohno, E.; Kakuda, Y. An Impersonation Attack Detection Method Using Bloom Filters and Dispersed Data Transmission for Wireless Sensor Networks. In Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Washington, DC, USA, 20–23 November 2012; pp. 767–770.
35. Chen, Z.; Guo, S.; Zheng, K.; Li, H. Research on Man-in-the-Middle Denial of Service Attack in SIP VoIP. In Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, China, 25–26 April 2009; pp. 263–266.
36. Kannhavong, B.; Nakayama, H.; Nemoto, Y.; Kato, N.; Jamalipour, A. A survey of routing attacks in mobile ad hoc networks. IEEE Wirel. Commun. 2007, 14, 85–91.
37. Kavitha, T.; Sridharan, D. Security Vulnerabilities In Wireless Sensor Networks: A Survey. J. Inf. Assur. Secur. 2010, 5, 31–44.
38. Wang, C.; Feng, T.; Kim, J.; Wang, G.; Zhang, W. Catching Packet Droppers and Modifiers in Wireless Sensor Networks. IEEE Trans. Parallel Distrib. Syst. 2012, 23, 835–843.
39. Junejo, A.; Komninos, N.; Sathiyanarayanan, M.; Chowdhry, B. Trustee: A Trust Management System for Fog-enabled Cyber Physical Systems. IEEE Trans. Emerg. Top. Comput. 2019, 9, 2030–2041.
40. Askham, N. The Six Primary Dimensions for Data Quality Assessment; DAMA UK Working Group: Bristol, UK, 2013; pp. 432–435.
41. Lear, D.; Droms, R. Manufacturer Usage Description Specification. 2018. Available online: https://tools.ietf.org/html/draft-ietf-opsawg-mud-25 (accessed on 26 January 2022).