MACQUARIE
University
SYDNEY·AUSTRALIA

Faculty of Science and Engineering
Computing Department

Trust-based Key Management Scheme for Wireless Sensor Networks

Master of Research Thesis

Daniel Onoja
Student ID: 45007624

Supervisor:
Michael Hitchens
Rajan Shankaran

25 November 2018

Statement of Originality

I hereby certify that to the best of my knowledge; the content of this thesis is the product of my own work. Where any material that has not been originated by me has been properly referenced. This thesis has not been submitted for any university degree or other purposes.

Daniel Onoja, Monday 25th November 2018

Abstract

Security for wireless sensor networks (WSNs) is a significant challenge. Several trust-based key management schemes have been proposed for securing WSNs without considering the limitations associated with sensor nodes in WSNs. Sensor nodes are small in size with limited capabilities such as, storage, energy and computation. Securing data communication channels requires the establishments of shared encryption keys. This thesis presents the design of a lightweight trust based key management scheme for WSNs. The proposed scheme provides data communication security through an efficient key distribution model, ensuring the generation and distribution of keys using an elliptic curve key encryption and a Diffie-Hellman key exchange protocol. The proposed novel trust based key management model provides a degree of security by enabling a sensor node to estimate a trust value associated with a target. The proposed model is a lightweight model, with the aim to provide security for WSNs, reduce computation overhead and save energy consumption in the process. The model ensures that only a small amount of a node resources is required for an efficient implementation by not recording unnecessary information and also avoiding multiple computation where possible. Simulation results demonstrate the performance of the proposed model.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Wireless communications, in recent years have experienced rapid growth. This has resulted from a growing need for efficient communication and constant connection between users and/or the Internet. People are making more significant use of wireless devices, such as laptops and personal digital assistants. However, these devices are unable to read the physical information in their environment by themselves (Lopez *et al.*, 2010a). With a need for wireless surveillance, environmental detection, information gathering from hostile or hazardous environment, health care body detection and aircraft monitoring, the introduction of sensor nodes makes it possible to collect data from the physical world.

A Wireless Sensor Network (WSN) is a collection of a number of sensor nodes, which could have similar capabilities or a heterogeneous mix of nodes with different capabilities. These nodes are often battery-operated sensors with limited communication, computing and data processing capabilities (Kaur, Gill and Dhaliwal, 2016). The sensor nodes are capable of sensing and reporting information to an aggregation head that can then forward the information to a base station in a centralized WSN approach. The sensor nodes are low cost nodes, which are small in size with limited capabilities such as: storage, energy and computation capabilities. WSNs have been extensively used in surveillance, monitoring environments by collecting data in order to detect any hazardous gas (Pietro, Mancini and Mei, 2003), automation sensors as depicted in *Figure 1,* for transportation systems, military environments for tactical advantage and gathering information from enemy territory, health-care, weather and security (Lu *et al.*, 2008). Wireless sensor networks in recent years have become the most widely used means of collecting information in different kinds of environments, making WSN an attractive area of research for academic and industrial communities (Abdallah *et al.*, 2014).

There are two types of environments for deployment in WSNs: controlled environments such as: homes, offices, border control, hospitals, and uncontrolled environments like hostile terrains such as disaster areas and military environments where it is critical to ensure a high level of security within the nodes and the network (Kumar, Jain and Barwal, 2014).

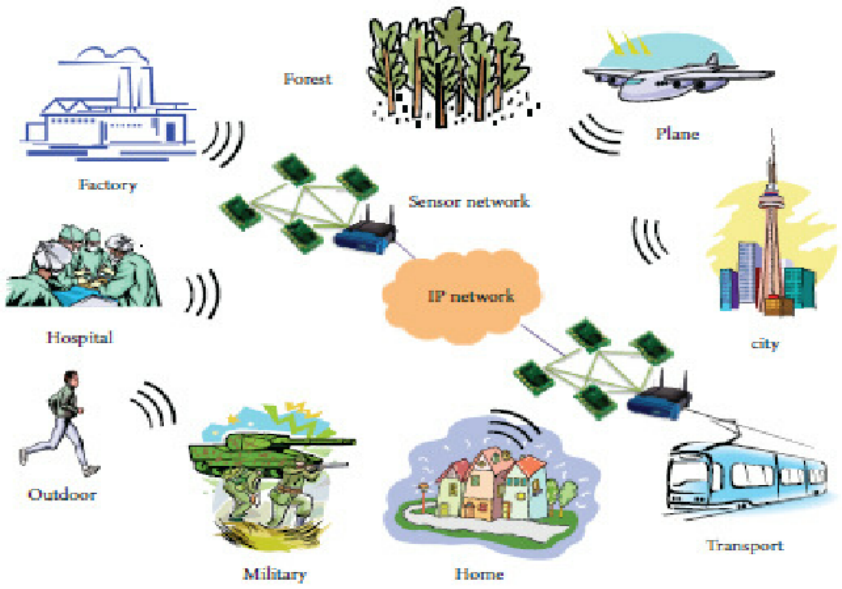

Figure 1: Application areas of wireless sensor networks (Boselin Prabhu and Sophia, 2011)

Security in WSN has been a challenge due to limited resources of the node. This is distinctly different to a traditional wireless network. As the use of sensor networks become wide-spread, security issues become a primary concern, especially in safety-critical tasks (Singh R, Singh D J, 2016), like a secured communication channel between nodes in a hostile environment where sensor nodes can be captured and manipulated easily. Securing data communication channels requires the establishment of shared encryption keys. Consideration of the limitations of such networks it is vital that the underlying computation capabilities of sensor nodes must be taken into account in the development of any new key distribution and management protocol. A key management protocol that could be effective in securing the communication channel between nodes and also providing a lightweight key distribution mechanism is a vital approach for our research. Our research is aimed at providing an efficient model without compromising the resources available to the sensor nodes in the network.

One mechanism which can be used to manage the decision-making in the key management protocol of the sensor node is a trust management system. Trust can be defined as the confidence or firm belief that a party has in the reliability of another party. People will not share information or communicate with other people that they don't trust. A trust management system, which is used to detect the trustworthiness and untrustworthiness of a node will establish the relationship of trust between sensor nodes and also provide an effective security measure for the network (Ishmanov *et al.*, 2013). Trust awareness is an essential tool in preventing malicious or corrupted nodes from effecting a legitimate node or network. These can be achieved either by direct trust or in-direct trust (Jiang *et al.*, 2015) which will be further discussed in chapter 2.

The basic security goals for wireless networks are the same as those for any network, availability, integrity, authentication and confidentiality. These security goals need to be accomplished to maintain an efficient level of security within the nodes or the network. For WSN, these security goals can be accompanied by other requirements such as scalability, efficiency and resilience (Lu *et al.*, 2008).

- Scalability: - The model needs to be able to maintain an efficient level of security to support a network with either a large number of sensor nodes or little amount of sensor nodes.
- Efficiency: - It is of great importance to consider the storage, processing and communication limitations of nodes in wireless sensor networks(Lu *et al.*, 2008). It is key that these limitations do not hinder the ability of the sensor nodes to optimally perform.
- Resilience: - Considering the vulnerabilities of sensor nodes, nodes need to be able to resist against different types of attacks including node capture is essential. The higher the resilience the lower the vulnerabilities.

The remaining chapters of this thesis is organised as follows: In Chapter 2, we discuss the background of wireless sensor networks and security threats. We introduce key management and related work in key management schemes. We introduce trust and a literature review on trust and trust based key management schemes. Chapter 3 provides the architecture and design of our approach and we then develop the model in Chapter 4. Chapter 5 presents simulation results of our model. Finally, Chapter 6 provides some concluding remarks.

Chapter 2: Background

As discussed in the previous chapter, a wireless sensor network is a collection of low-cost sensor nodes, which are often small in size and battery operated. There are different types of wireless sensor networks which are defined by the environment in which they operate.

- Body area wireless sensor networks
- Underwater wireless sensor networks
- Terrestrial wireless sensor networks
- Underground wireless sensor networks

Some of these network environments such as underground and underwater have more difficulty in node deployment and node maintenance than others. The design of sensor nodes for some environments is also more expensive than others. There are different network architectures in wireless sensor networks. The structure of the network architecture depends on the utilization of the network, the tasks required from the sensor nodes and the applications available. Two types of WSN sensor node structure are:

- Homogenous
- Heterogeneous network nodes.

Homogenous wireless sensor network: one where all the sensor nodes have the same capabilities such as: battery power, storage, sensing, procession, and communication capabilities(Uplap and Sharma, 2014). However, (Kumar *et al.*, 2013) explains that homogeneous wireless sensor networks might be difficult to achieve because some homogeneous sensors might not have the same level of initial energy, depletion rate and other capabilities.

In heterogeneous wireless sensor networks the sensor nodes have different capabilities such as, computing power, sensing range, transmission range, storage and battery power. The nodes could be classified into categories based on their capabilities, roles and functions. One heterogeneous architecture is a heterogeneous hierarchical wireless sensor network. In a hierarchical wireless sensor network, the nodes are categorised into three classes of base station, cluster head and sensor nodes. A heterogeneous wireless sensor network is a more complex network than a homogeneous wireless sensor network due to deployment, management, and topology control difficulties (Wu and Chung, 2007). In our research we used a hierarchical wireless sensor architecture. In a hierarchical WSN, Sensor nodes are primarily used for sensing, event detection and transmitting the information to the base station (Tiwari and Kumari Kushwaha, 2017). Sensor nodes may rely on their neighbouring nodes to transfer

data to the base station due to limited transmission range. A node attempting to transmit information to the base station will identify the best path to the base station through its neighbouring nodes. Relying on other nodes in WSN is a vulnerability as an attacker might aim to exploit legitimate nodes by carrying out attacks through deceit. Even though heterogeneous networks could be more secured than homogeneous networks, both networks have security vulnerabilities. WSN nodes have low capabilities and generally have physical vulnerabilities as they are dispersed over an area.

Security in WSN has been a challenge due to the nature of the sensor nodes which are low cost nodes, small in sizes with limited capabilities such as; storage, energy and computation capabilities. The main aim of this research is to ensure a secured key management model for WSNs by developing a novel trust-based model that will provide network security without compromising the resources available to the sensor nodes in the network.

2.1 Security Threats

There are various types of security threats to WSNs. WSNs are vulnerable to many security threats due to the nature of the underlying network, resources available, architecture of nodes and their internal configuration in the environment they operate. Some of the common security threats to key management and trust models in WSN are:

2.1.1 Sybil Attack

A sybil attack is when an attacker uses a compromised node to claim several identities of other nodes in a network (Karlof and Wagner, 2003). This is a significant attack because in WSNs, the sensor nodes are self-organising and rely on other nodes either through a single hop or multi-hop for routing and communication. Even with the different types of key management cryptography implemented to mitigate Sybil attacks, Sybil attacks still remain a threat in WSNs. An attacker can create a fake identity for a malicious node to inject the node into the network. An attacker can also use the identities of legitimate nodes in a network. An example would be when an attacker compromises a node, it then uses this node to obtain several identities of legitimate nodes in the network. A node's identity is required for generating and exchanging a key in most key management algorithms. When a malicious node is trying to establish a connection with other nodes, they will be required to share/exchange identities. The malicious node will be able to obtain the legitimate node identity, and other required information. The attacker will continue to use these false identities through-out the network and might be able to eliminate the legitimate nodes from the network by carrying out other forms of attacks.

These identities can be either fabricated or stolen identities (Singh R, Singh D J, 2016). In fabricated identity attack, the malicious node forges multiple fake identities for itself and pretends to be more than one node. The malicious node could act as a new node or an existing node. Through the stolen identity attack, the attacker obtains the identities of legitimate nodes in the network and uses it to as its identity to establish communications with other legitimate nodes. A malicious node could also use the identity of a damaged node or a node that is no longer active in the network. (Newsome *et al.*, 2004).

2.1.2 Denial of Service attack

Denial of service (DoS) can be an unintentional disruption of a transmission through interference, noise or collision and/or a malicious attack (Alajmi, 2014). DoS attacks are mostly malicious attacks with the sole intent on disrupting a network. A DoS attack targets the network resources and prevents legitimate nodes from using these resources in the network. In the event of such an attack, the attacker would typically send multiple spurious packets into a network to exhaust the network resources available with the sole intention of disrupting the network services (Pathan, Lee and Hong, 2006). It is difficult to detect such an attack (Kaur, Kumar and Bhandari, 2017) (Kumar, 2016). For a key management scheme, a DoS attack can attempt to exhaust the legitimate node resources by sending multiple session key requests or sending multiple packets. Another highly disruptive DoS attack is a distributed DoS attack, where the attacker uses multiple malicious nodes to flood the network with packets. In a key management attack, we can consider multiple key requests which will keep the nodes busy and use up energy resources as a distributed DoS attack.

2.1.3 Node Capture

Node capture in WSNs is a very significant attack. If an attacker gets a hold of a sensor node of a network, this can lead to other types of attacks. (Abdallah *et al.*, 2014), In key management for WSN, sensor nodes store sensitive information such as keys, node identities, routing tables and data. The node and the information could be compromised in a node capture attack (Du *et al.*, 2004). Once an attacker has information on the network keys, they can decrypt messages, generate keys with other legitimate nodes, disrupt the network and carry out other types of attacks such as:

Node Malfunction: A captured node could be damaged or modified to malfunction, and a malfunctioning node could create inaccurate data that could expose the integrity of the sensor network (Kumar, Jain and Barwal, 2014)

Node Replication: A type of Sybil attack that an adversary can undertake after node capture. The attacker tries to introduce the malicious nodes into the network to establish connections with legitimate nodes (Kumar, Jain and Barwal, 2014)

2.1.4 On-off Attack

In an on-off attack, an attacker intends to disrupt the network's performance without being detected or eliminated from the network. (Jiang *et al.*, 2015), The attacker alternates its behaviour between good and bad behaviour in order to avoid detection or prolong its attack life time before being detected. An attacker might perform on-off attack on recommendation activities such as: giving false recommendation and accurate recommendation at different times. For example, showing good or bad behaviour can be done in the context of recommendations, or sharing a session key in such a way that a malicious node acts in a good way for a certain time to lure a legitimate node to trust and want to cooperate with the malicious node, or can be done in the context of normal sensor network activities, such as aggregation, routing, and sensing physical phenomena.

2.1.5 Bad Mouth/Good Mouth

Bad mouth attack is a common attack in Trust management in WSNs. It is essential to consider such an attack in the design of a trust mode as in most cases, a trust management system might rely on recommendations from sensor nodes. (Jiang *et al.*, 2015), Bad mouth attack is when malicious nodes give a negative recommendation on a legitimate node as malicious to eliminate it from the network. Good Mouth attack is when a malicious node labels another malicious node as a legitimate node in other to perform other forms of attacks.

2.2 Key Management

Key management is the process of managing cryptographic keys in a system (Abdallah *et al.*, 2014). The managing system controls the generating and distribution of keys, key storage locations, key revocation and key updates among other key responsibilities in a network (Seo *et al.*, 2015). To mitigate against most of the security threats in WSN, it is essential to have effective communication security. This can be based on cryptographic algorithms that ensure sensor nodes have a set of shared encryption keys. Data communication is one of the core responsibilities of sensor nodes in a wireless sensor network. Therefore, ensuring a secured data communication is the main responsibility of an efficient key management scheme. Several

key management approaches have been proposed for WSNs (Pietro, Mancini and Mei, 2003), (Du *et al.*, 2004), (Lu *et al.*, 2008), (Mansour, Chalhoub and Lafourcade, 2014), (Abdallah *et al.*, 2014) . However, some of these traditional key management systems are not dynamic in nature. Keys are generated and stored in the sensor nodes before node deployment and not refreshed or, if even they are refreshed, there is still no change in which nodes they are shared with. It is essential to avoid using static keys in such situations. Sensor nodes are capable of changing their behaviours which could be as a result of power shortage, an attack, performance or cluster change.

Key management refers to the mechanism of cryptographic techniques in key generation, key distribution and key maintenance such as re-keying or revoking a key (Abdallah *et al.*, 2014). Key management in WSNs enables sensor nodes to generate different types of keys for various tasks and node functions such as: secured data communication, routing, and storing information, etc. In WSNs there are a number of different varieties of keys used between the network nodes to ensure secured data communication, such as network keys, cluster keys, peer-wise-keys. There are different types of cryptographic algorithms: private (symmetric) key cryptography; which requires that the same key is used for both encryption and decryption. (Seo *et al.*, 2015), Symmetric key cryptography has a high computation overhead and uses up a high storage capacity to store the shared secret key. Public (asymmetric) key cryptography; uses different keys for encryption and decryption. It uses less storage space, it is scalable, resilient against some attacks such as node capture and impersonation, and also adapts to mobility. These algorithms ensure that the primary security functions required like confidentiality, authentication, integrity, non-repudiation and key exchange are incorporated. Key management is a challenging issue in the design and development of secured communications in WSNs (Lu *et al.*, 2008). Due to the limited capabilities of the sensor nodes, it is difficult to ensure security in WSNs. Therefore, recent research work has focused on developing key management approaches that requires lesser resources. There have been several proposed key management schemes for WSNs (Simplício *et al.*, 2010)(Pietro, Mancini and Mei, 2003)(Abdallah *et al.*, 2014) for different network topologies like peer-to-peer (Lu *et al.*, 2008), or hierarchical networks. Some schemes are designed for specific functions in wireless sensor networks like static networks, dynamic networks, heterogeneous functions and more in the last few years. Due to time constraints, we decided to select an efficient key management scheme that will be suitable for this research. Previous key management approaches to wireless sensor networks assist with an effective key management selection. In the next section, we will discuss some key management approaches.

2.2.1 Related Work

In the last few years, many key management schemes have been designed for WSNs. Some schemes were designed for specific functions and approaches considering factors like, key distribution, deployment method, sensor node capabilities including mobility, or network topology. Key management can be categorised into a number of approaches by - dynamic architecture, key distribution, pre-deployment knowledge scheme. We shall examine each category along with a number of examples of their use.

Dynamic Architecture: It is essential to consider the network architecture or topology suitable for a model design. A general framework of key management for distributed peer-to-peer WSNs consisting of heterogeneous sensors was developed by (Lu *et al.*, 2008). The topology was a hierarchical WSN design adopted from (Law *et al.*, 2003). The sensor nodes have different classifications with the base station being the most powerful node and the sensor node the least when considering factors like processing capabilities and communication range. In (Lu *et al.*, 2008) the base station acts as a key distribution centres.

Inspired by (Liu, Ning and Li, 2005), Lu et al.'s scheme uses a pool of random keys but with a polynomial share which is allocated to each sensor and each polynomial can generate multiple keys. The scheme addresses key connectivity, scalability and resilience. In terms of resilience a contestable assumption is made, that the nodes in the network can recognise compromised nodes.

(Mansour, Chalhoub and Lafourcade, 2014) evaluated different types of multi-hop authentication protocols and key establishing mechanisms in WSN, with the view that the area of node authentication protocols had only been researched by a few in multi-hop WSNs and most proposed protocols neglect the multi-hop factor. To illustrate some details about multi-hop authentication in WSNs, we consider the methods in (Mansour, Chalhoub and Lafourcade, 2014)

1. Pre-deployed Keys: each node knows the public key of the base station and its own private key before deployment. Using the Diffie-Hellman key exchange protocol, a node can compute a shared key with the base station.
2. Authenticated Multi-hop Join Protocols: there are two protocols *DJB (Direct join to the base station) and IJB (Indirect join to the base station)*. These protocols are for a network join authentication key establishment. Direct join to the base station allows new sensor nodes within the range of the base station to join the network (Mansour, Chalhoub and Misson, 2014). Indirect join to the base station is when a new node wants to connect to

the network through another neighbouring node (Mansour, Chalhoub and Misson, 2014).

After these authentication protocols (Mansour, Chalhoub and Lafourcade, 2014) proposed two major key establishment protocols: multi-hop key establishment protocol using the base station and Multi-hop key establishment protocol without base station.

(Gandino, Montrucchio and Rebaudengo, 2014) proposed a key management system for a static WSN. Static WSN is stationed sensor nodes with no mobility. In this model, a symmetric key establishment model was presented.

Key Distribution: In wireless sensor network, key distribution enables multiple nodes to exchange keys securely over a communication channel (Coles, Metodiev and Lu, 2016). There are several methods of key distribution. (Simplício *et al.*, 2010), presented a random key pre-distribution management model for WSNs. Another random key-assignment scheme for WSN was proposed in (Pietro, Mancini and Mei, 2003). There, the authors developed a pairwise solution for a secured pairwise communication within a peer-to-peer network by storing a set of keys in each node with two protocols. (Pietro, Mancini and Mei, 2003).

The mechanism in (Abdallah *et al.*, 2014) uses ECC for encryption and Diffie-Hellman protocol to establish/distribute the keys in the network. In this model, the keys are uniquely generated and shared by the sensor nodes. The WSN architecture was a hierarchical sensor network with different classification of nodes; base station BS, cluster-head CH and sensor nodes N. (Abdallah *et al.*, 2014) In hierarchical WSN topologies the base stations are usually stationary and considered tamper resistant, these nodes or super nodes have higher capabilities than any other nodes . The difference in the pre-deployment notations in (Mansour, Chalhoub and Lafourcade, 2014) and in (Abdallah *et al.*, 2014) is that in the latter the keys are not known by the sensor nodes pre-deployment. The key management procedures considered issues like; new nodes deployment, nodes elimination and revocation, mobility management and re-keying procedure. This key management scheme is an efficient mechanism which reduces computation overhead with the use of ECC and considers various factors in the different areas of the networks. This key management scheme was selected for this research and is discussed further in Chapter 3.

Pre-Deployment Knowledge Scheme: A few researchers believed that it was possible to estimate in advance the locations of each sensor nodes for a static wireless sensor network. In other words, they could create a cluster pre-deployment and generate keys, store keys and share keys between the node's pre-deployment. Considering aerial deployment from an

airplane, (Du *et al.*, 2004) presents a model that determines the neighbours of each node in the network before deployment. The keys are generated for the sensor node and its neighbouring nodes and are stored in the node's directory before deployment. Sensor node deployments are random and difficult to predict their direct neighbours prior to deployment. With the assumption in this model, that sensor nodes are static once deployed. The random pre-distribution schemes use a key pool system, where, the set of keys are selected from the pool and stored in each nodes memory before deployment.

2.3 Trust

Trust can be defined as the confidence or firm belief that a party has in the reliability of another party. (Hoffman, Zage and Nita-Rotaru, 2009) defined trust as the level at which an entity has total confidence in another entity within the context of an event or decision. There are different definitions of trust depending on the context. In human context, most relationships are built on trust. A person will need to trust the other party to enter any kind of relationship such as customer relation, credit allowance, business partnership, friendship and love-based relationships. Knowing how good an entity's behaviour is against how bad the entity behaves, people are able to evaluate the level of confidence they have that a person would act in the right manner. Thus, we should not focus solely on trust but also consider distrust, which is the opposite of trust and they both apply as either, or. Most people will not share information or communicate with people that they don't trust. Distrust has received less attention in the literature than trust. There are various definitions of distrust (which is also known as disbelief, mistrust or untrustworthy). Given the general understanding of trust and distrust, both apply in WSNs and this research considers the trust and/or distrust relationships from one sensor node to the other.

Trust is an important feature in WSNs and can help solve several related issues in such networks: The uncertainty in interaction (Lopez *et al.*, 2010b). However, a trust association between two sensor nodes can also be used to carry out other functions beyond collaboration. There are various areas of study wherein the notion of trust has been extensively researched (Junqi Zhang *et al.*, 2010) analysed different areas of trust in various trust related contexts such as trust process, trust platforms and computing and trust management. Trust models can be classified into two categories as centralized and distributed models (Han *et al.*, 2014). In a centralized model, a super node (typically the base station) calculates the trust values for sensor nodes, whereas in distributed trust schemes, the sensor nodes calculate their trust values in other nodes by themselves. Trust in WSN has been widely researched as an effective

mechanism in solving security issues by providing a mechanism to safeguard against security threats to the network. Some trust management approaches will be discussed in the next section. Trust management is an essential tool in preventing malicious or corrupted nodes from disrupting the activities of a legitimate node.

There are a number of different trust properties in an efficient trust management system. The trust properties differ according to the trust model in question and must be defined to facilitate the trust calculation or evaluation. Some of the general trust properties are: trust range, direct trust and indirect trust, initial trust value (Jiang *et al.*, 2015).

A common range for the trust value is [0,1] where 0 is untrustworthy and 1 or any value in between is trustworthy depending on the trust threshold. A threshold is a point of reference or a set point that differentiates between positive and negative. An example (Zicari *et al.*, 2017) where trust value is [0,1], the threshold is set at 0.5. Any value below 0.5 is classified as negative and any value above 0.5 is regarded as positive, and if the trust value is equal to the threshold no decision might be taken. Other trust ranges could be [0, 100] (Junqi Zhang *et al.*, 2010) or –[1, +1] (Pirzada and McDonald, 2004).

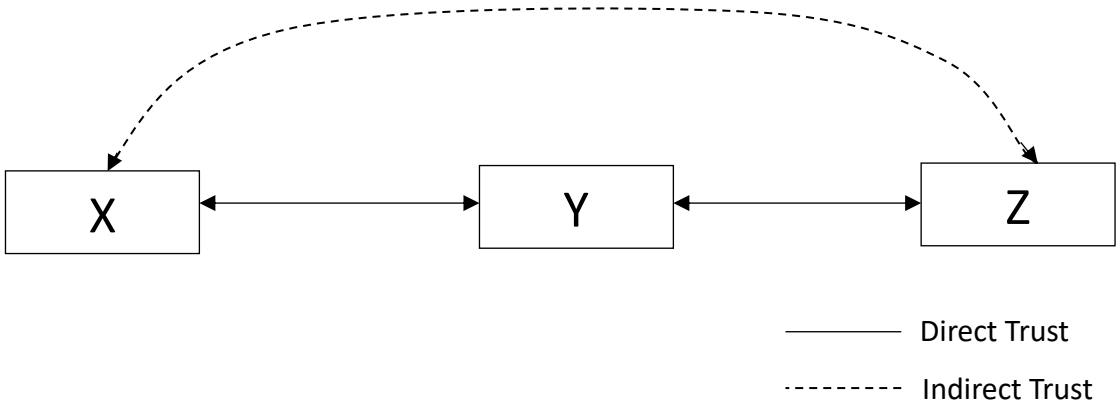

Figure 2: Trust framework

A final trust value can be calculated from a range of different components, each representing a facet of the target node's behaviour. Each component is given a weighting value, which is used in calculating the final trust value from the component values. The weight given to a trust component shows the significance of that trust value in the final trust calculation.

2.3.1 Direct Trust

Direct trust is the direct interaction or experience trust associated with a node that is directly measured by its neighbouring nodes within a single hop range. As shown in *Figure 2*, Node X has a direct relation with node Y but not a direct relationship with node Z, node Y has a direct

relation with both node X and node Z. Node X is able to calculate the direct trust it has for node Y based on their interactions to determine if node Y is trustworthy or untrustworthy. There are a number of methods that have been introduced for calculating a direct trust value (Lopez *et al.*, 2010b)(J. Zhang *et al.*, 2010)(Tiwari and Kumari Kushwaha, 2017). Most are designed depending on network services but some are designed depending on where and how the records are stored or the computation capability of the processing node. (Huang and Nicol, 2009) Basic trust concept denotes B, D and U as belief, disbelief and uncertainty which can be calculated as B + D + U = 1, where 1 is positive and 0 is negative.

2.3.2 Indirect Trust

Indirect trust, sometimes referred to as recommendation trust, is when a subject node requests a recommendation for a target node from a third-party node which has had a direct interaction with this target. As shown in *Figure 2,* node X is the subject node with no previous interaction with the target node Z, Node Y is the third-party node with previous direct interactions with node Z. Node X will request a recommendation from node Y because Y has a direct trust value for node Z. Node X's in-direct trust value for node Z is the recommendation from node Y or the direct trust value of node Y for node Z. There are different models proposed to calculate indirect trust, in (Jiang *et al.*, 2015) calculates indirect trust, considering a multi-hop scenario when there is no direct interaction between two nodes (X and Z).

2.3.3 Initial Trust Value

It is usually necessary to decide where a node's trust value might start from at the time of deployment. The initial trust value assigned to each node could be 0, 1, 0.5 depending on the trust weighting parameters. In many cases 0.5 is assigned as a neutral initial trust value which could either increase to make the target node a trustworthy node or decrease to make it an untrustworthy node. It is not effective to set the initial trust value at 0, or 1. If a node initial value is 0, signifies that the node is untrustworthy, the neighbouring nodes might not want to cooperate with an untrustworthy like node. If in a network the initial trust value for all nodes are set to 0, the network might fail as there will be no communication in the network because all nodes are represented as untrustworthy. If the initial trust value is set at 1, means the node is trustworthy and should the node be compromised before any interaction, the malicious node would be able to carry out attacks because other nodes might want to interact with it. A common challenge with an initial trust value set at neutral (Uncertain) of 0.5, is a situation

where a node remains idle without interactions. A few trust management approaches address the issue of idle or selfish nodes, using time factor to determine if it is trustworthy or untrustworthy.

2.3.4 Related Work

(J. Zhang *et al.*, 2010) considers the deployment of a large WSN. The research models trust management based on direct and group trust to address the security issues in the deployment of a large number of sensor nodes. In this research the root of the hierarchy is a base station which is assumed to be the strongest node with more or unlimited computation power and resources. The next set of nodes are the cluster heads which are each the head of a group of sensor nodes. The sensor nodes are the least powerful nodes in the network. For this model, it is assumed that the network nodes are in a fixed location where their location and communication range are known.

The trust management is divided into two level of trust: a node level trust management; which is the level of trust between individual nodes and, a cluster head level trust management; which is the level of trust the cluster head has in the nodes belonging to its cluster (Intra-cluster trust management). The model considers the time factor, where it gives more value to the most recent communication or corporation in the newest time frame. The time frame is crucial for trust evaluation. (The trust value with the longest time will depreciate in weight in the overall trust value and expire eventually). It is also imperative that a trust management system in wireless sensor network is scalable and can adapt to sensor nodes mobility. (Junqi Zhang *et al.*, 2010) advanced the previous research by adding a dynamic trust management with flexible nodes and a time window information of several time units. The dynamic trust management accounts for an Inter-cluster trust management, where a node can move from one cluster to the other and the cluster head will be required to send the trust records of that node to the new cluster head. In the model, the cluster heads also record the energy level of each sensor nodes in the group cluster. The time units record the interaction history within a certain time unit. Due to limited resources, recording all historic interactions in a time unit will exhaust the storage capability of a sensor node.

Considering a remote deployment environment (Kaur, Gill and Dhaliwal, 2016) developed a trust based routing model in a secured trust based key management routing framework (STKF) to prevent against passive and active attacks. Here a neighbour node is responsible for neighbour selection which is dependent on the past and present node to node cooperations for routing data from source to destination to guarantee a secure and trustworthy route. By node

to node selection based on past and present behaviour, a node can also eliminate a malicious node from its routing table by updating the direct trust values.

Also in (Kaur, Gill and Dhaliwal, 2016) indirect trust is considered as the trust relation between distributed nodes who do not have direct interactions with a target node. The indirect trust/recommendation for any node is could be collected from its neighbouring nodes.

(Che *et al.*, 2015) is a lightweight trust management based on Bayesian and Entropy model, developed with consideration for computation overhead. Computation overhead is a major challenge in wireless sensor networks because in most models, the more efficient the model becomes the higher the computation overhead. In this model, to minimise the computation overheard, there is provision to consider only the direct trust as a satisfactory trust value without computing the indirect trust. A valuable addition to this model is the direct trust confidence level of a node. Direct trust confidence is a unique attribute in a trust management. It is a lightweight approach because it reduces computation overhead. A node can measure the confidence level it has in a trust calculation (Patel *et al.*, 2005). (Che *et al.*, 2015), if the direct trust confidence is above a network defined threshold, the direct trust value is sufficient enough and considered as the final trust value, thus there will be no indirect trust calculation, and this will reduce the computation overhead. However, if the trust confidence is below the threshold, an indirect trust calculation would be required for the final trust calculation.

2.4 Trust Based Key Management

Most key management systems as discussed earlier are focused on securing data communication channels, by generating and establishing keys between sensor nodes. This provides a level of security for data communication but could be vulnerable to other forms of security threats to the sensor nodes as discussed earlier. Trust based key management enables the key management model to incorporate a decision mechanism that identifies and eliminates malicious nodes from the key establishment process. Several trust based key management schemes have been proposed over the years to ensure a secured data communication in WSNs such as (Khatri, 2014) which uses the identity of a node and the trust model in the key management approach to establish keys. In (Kaur, Gill and Dhaliwal, 2016), a secure trust based key management routing protocol was proposed to provide a secured and trustworthy data communication channel. In this paper, the trustworthiness of a node on a route depends on the present and past interactions of one node to another. The routing protocol provides an update mechanism where a route update is required to identify and remove compromised nodes from the route. A unique channel link is created between nodes using a “q” composite random key

pre-distribution to guarantee complete data communication. (Feng, Kuan and Hao, 2010), proposed a combined protocol using trust computing technology and secure node authentication. This paper introduced a multicast framework believed to be secure to mitigate potential threats in routing or route maintenance.

2.5 Chapter Summary

In this chapter, we discussed some major security threats to key management in wireless sensor networks and how they can affect the network. We explained the importance of an efficient key management protocol and evaluated existing key management schemes which we believed were relevant to this research to identify the most suitable model which we will discuss further in the next chapter. Additionally, we discussed some of the previous trust models within the literature review. As illustrated, some models were designed for a static network where the nodes were assumed to be in a fixed location, while others had a high computation overhead. A few models use up the node storage and energy without considering the limited resources in the nodes. An important issue is to provide a secured scheme with a low computation overhead while considering the sensor nodes limited resources in a scalable and adaptable manner. We will be discussing the network architecture of our model in the next chapter including the trust model framework.

Chapter 3: Architecture and Design

Trust based key management is significant for sensor networks as it provides a collaborative mechanism to isolate and exclude malicious or corrupted nodes from the network. In this chapter we describe the architecture of a heterogeneous hierarchical wireless system. We then provide an overview of our proposed key management architecture, after which we discuss the trust model that helps monitor the sensor nodes' behaviours in the network and dynamically evaluates their trust values.

3.1 Network Architecture

As discussed in Chapter 2, there are various forms of WSNs. These includes peer-to-peer networks, homogenous static networks, and heterogeneous hierarchical networks. For this research we have chosen to use a heterogeneous hierarchical topology as shown in *Figure 3*. Unlike a flat homogenous network, a heterogeneous network consists of different types of nodes with different capabilities such as transmission range, computation power and sensing capabilities. The most common heterogeneous architecture is a heterogeneous hierarchical WSN. In a hierarchical WSN, the nodes are categorised into three classes of base station, cluster head and sensor nodes. We consider this network architecture to be scalable in adapting to a larger number of sensor nodes.

Base Station: - The base station, also known as the super node, manages the cluster head and the sensor nodes. The base station acts as the network administrator and has a wide communication range enough to reach all the cluster heads and sensor nodes in the network. It could add a new node to the network and remove a malicious node or cluster head from the network. We assume that the base station is the highest capacity node with unlimited resources and can compute with the highest capabilities. The base station is assumed to be tamper resistant and considered to be the most secured and trusted node in a wireless sensor network. The base station will update the network key and inter-cluster key if it eliminates a node or at a system defined time.

Cluster Head: - After deployment, the sensor nodes are divided into groups or clusters based on their communication range. A cluster head is in the second category of nodes with less capabilities than the base station. These nodes have more power and resources and wider transmission range when compared to normal nodes in the cluster.

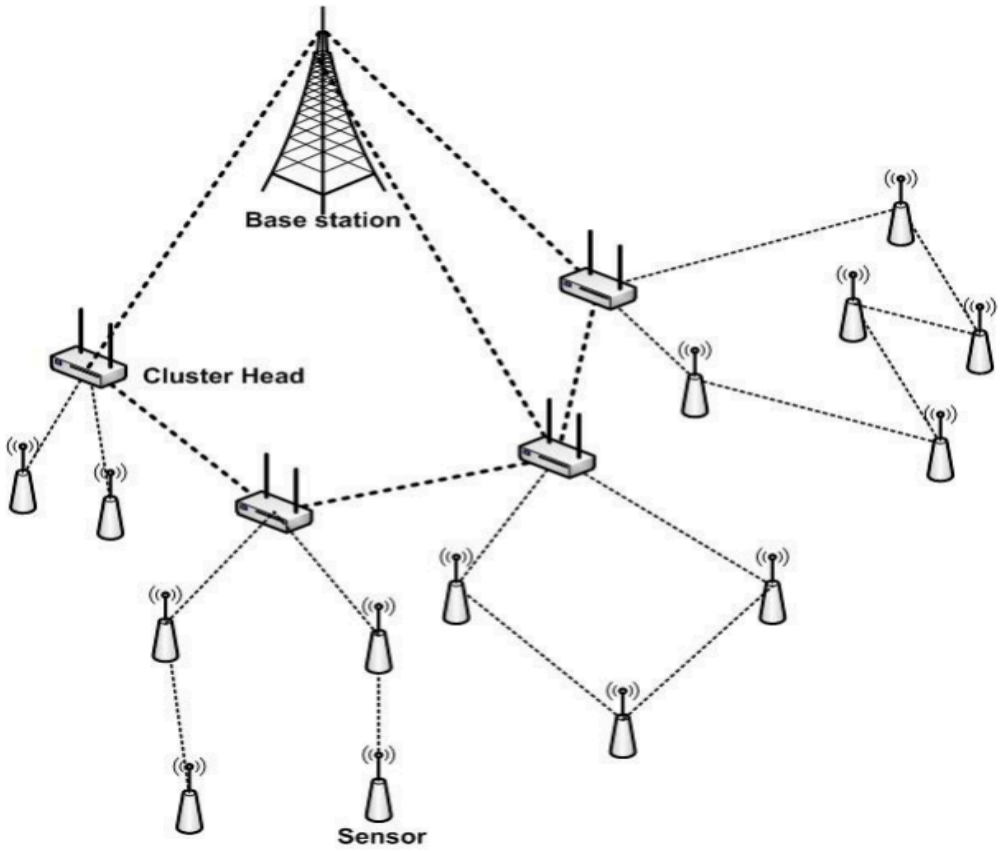

Figure 3: Hierarchical Architecture (Abdallah et al., 2014)

There can be multiple clusters in a network and each cluster has a cluster head who manages the sensor nodes within its cluster group. We assume that the cluster heads are appointed by the system, the transmission range of the cluster head is wide enough to reach the base station, neighbouring cluster heads, and to cover its cluster area. The CH keeps record of the trust values of sensor nodes in its cluster. The record is obtained when there is a need to change the cluster key or when a malicious node is identified/reported by a sensor node. It could eliminate a malicious node from its cluster, allow a node to join the cluster, revoke/change the cluster key and evaluate the trustworthiness of another cluster head.

Sensor Node: - This category of nodes primarily has two main tasks: sensing and communicating/relaying of data gathered. The sensor nodes gather information/data from its environment, then transmits the data collected to its cluster head in a single hop or multi-hop (through other sensor nodes). The sensor nodes are in the lowest category of the network. These nodes have limited capabilities compared to the nodes in the higher categories. Sensor nodes might rely on their neighbouring nodes to transfer data to the cluster head and rely on the cluster head to transfer data to the base station due to their limited transmission range. The sensor node is where most of the trust calculation occurs. The sensor node evaluates the trust of its neighbour nodes before establishing keys for secured data communication.

For example, a node attempting to transmit information to the base station will identify the best path to the base station through its neighbouring node by evaluating the trust it has in the nodes. A sensor node can only be attached to one cluster at any given time, also the sensor nodes in the network are mobile and has the ability to move from one cluster to another cluster.

3.2 Proposed Key Management System

As this research is focused on trust management, not key mechanisms, we decided to select an existing key management scheme to which we would add trust in (Abdallah *et al.*, 2014). The key management scheme takes into consideration, the limited resources associated with sensor nodes and it is adaptive to node memory size, node energy, and limited computation capabilities, by implementing an efficient and less complex approach compared to existing key management schemes. The network architecture was designed for a hierarchical sensor networks with different classification of nodes; base station BS, cluster-head CH and sensor nodes N. The mechanism uses an elliptic curve public key cryptography (ECC) and Diffie-Hellman key exchange protocol which uses less computation overhead to ensure efficient and secure key generation and exchange in different node categories of the network. The keys are not generated or stored in the nodes pre-deployment (Abdallah *et al.*, 2014). The key management scheme provides a level of secured data communication by addressing some security threats like node capture in cluster head, node replication at deployment phase and energy depletion attack during key establishment procedures between sensor nodes. In this model the cluster heads are nodes with higher processing capacity with a large storage and long-lasting batteries. Cluster heads can have inter-cluster communications where, a cluster head can communicate with other cluster heads and an intra cluster communication where, a cluster head can communicate with sensor nodes within its cluster. The cluster head can also relay data collected from sensor nodes to the base station. It is assumed that the cluster heads are tamper resistant and an automatic destruction or auto memory eraser functionality can be implemented to provide resistance against node capture attack. The base station is assumed to be in a well secured location which might be localized anywhere around the network. It is considered to be the most secured node and the most trusted and reliable node of the WSN. For more information on ECC see (Koblitz, 1987). ECC provides an efficient public key cryptography with a high security standards as other cryptosystem technique. Equation 1 is a form of Galois finite field in ECC see (Koblitz, 1987).

$$E(F_p) = \{(x, y) \in F_p^2, y^2 = x^3 + ax + b\} \in \{O\} \quad (1)$$

a and b will need to satisfy $4a^3 + 27b^2 <> 0$

Key Establishment (Abdallah *et al.*, 2014)

A: Prior to Deployment

1. The base station randomly picks a prime number p, the elliptic curve $E(F_p)$ and a group point G ∈ F_p. (Miller, 1986)
2. The base station then generates its private key $pi_{BS} \in Z_P$, where $2 \leq pi_{BS} \leq p-1$.
3. The BS then computes its elliptic curve public key, $PU_{BS} = pi_{BS}G$.
4. The elements p, $E(F_p)$, G, PU_{BS} are stored in each sensor node along with a temporary key K_o prior to deployment.

The temporary key will be used to authenticate the deployed nodes.

B: Individual Key

The individual keys are the sensor node private and public keys used to secure data communications between the base station and the sensor nodes.

Once deployed, after network integration and cluster establishment, each node will compute its private key $pi_x \in Z_P$ by:

1. Hash function $pi_x = Hash\ (id_x \,||\, K_o \,||\, N_i)\ mod(p)$ *where x is a sensor node and* N_x *is a nonce generated randomly, ensuring that all the private keys are not the same.*
2. Once the private key x_i is known, the sensor node computes its own elliptic curve public key $PU_x = pi_xG$. With these the sensor node will be able to compute an individual pairwise secret key with the base station, $K_i = pi_x pu_{BS} = pi_x pi_{BS} G$.
3. The sensor node x sends its public key PU_x to the base station for validation and to be stored in the public key's storage. Using the temporary key K_o the message is authenticated to ensure that it is from a genuine sensor node by a Hash Message Authentication Code (HMAC).
4. The base station verifies the identity of the sensor node, validates and saves the public key of the sensor node in its public keys storage, then establishes the individual pairwise secret key $K_i = pi_x pu_{BS} = pi_x pi_{BS} G$
5. Finally, the base station sends an acknowledgement message to the sensor node that is authentication by a MAC calculated using K_i. The sensor node will immediately delete K_o (the initial key) from its memory

C: Intra-Cluster Key

Using the same method, (Abdallah *et al.*, 2014) presented Intra-cluster pairwise keys and cluster key management, to secure data communication between sensor nodes and their cluster head. The sensor nodes can setup a pairwise key with its neighbouring nodes in other to relay

information to the cluster head. The sensor node pairwise keys can be established in the same way as the individual keys described above. The difference in this case is that the public keys must be requested from the base station and each node is to verify its validity before key setup.

1. The base station, using the secret key shared with a requesting node will calculate a MAC of the public key
2. Sensor node x and y can set up a pairwise key $K_{x,y} = pi_y pu_x = pi_x pu_y = pi_x pu_y G$
3. Similarly, the cluster head and sensor node pairwise key can be set up where, $K_{CH,x} = pi_x pu_{CH} = pi_{CH} pu_x = pi_{CH} pu_x G$

D: Cluster Key

There is also a cluster key in the model which is a general key shared within a cluster to enable in-network communication and reduce usage of resources. The cluster key is created once a cluster is established. Each cluster have different cluster key. For the cluster key the model proposed a group communication private sharing key derived from (Shamir, 1979) which is assumed to be more efficient than exiting procedures. For each node i in a cluster, its cluster head calculates and sends its public key

$$PU_{CH,i} = pi_{CH} \sum_{n=1, n \neq i}^{M_i} PUn \quad (2)$$

where M_i is the number of nodes in the cluster

The cluster key CK can be computed by each node simply by adding the intra-cluster pairwise key and its public key

$$CK = K_{CH,i} + PU_{CH,i} = pi_{CH} \sum_{n=1,}^{M_i} PU_n \quad (3)$$

where $K_{CH,i}$ is the intra − cluster pairwise key

E: Inter-Cluster Key

With the same method as the cluster key, the cluster heads and the base station can establish an inter-cluster key IC

$$IC = pi_{BS} \sum_{CH=1}^{M_{CH}} PU_{CH} \quad (4)$$

The Inter-cluster key can be used to ensure a secured message broadcast between cluster heads and base station. This could also be referred to as the second level of the network hierarchy.

F: Network Key

The network key is a key shared by all the participating nodes of a network from the base station to the sensor nodes. This is key is generally used for a secured network message broadcast by the base station. It is an alternative to broadcasting first to the cluster head (Using the inter-clusters key) then having the cluster heads broadcast to their clusters (Using an intra-cluster key).

The network key NK was distributed in two procedures, which is a combination of the Inter-cluster key and the Cluster-Key.

1. The base station randomly generates the network key
2. Encrypts the network key with IC and transmits the network key to all the cluster heads
3. Each cluster head decrypts the message and encrypts the NK with CK, before transmitting the message to all members of its cluster.

The key management procedure considered issues like; new nodes deployment, nodes elimination and revocation, mobility management and re-keying procedure. The key management scheme is an efficient mechanism which reduces computation overhead.

G: Key Expiry and Refresh

Keys may expire and need refreshing under two circumstances. Firstly, keys are allocated a system defined life-time. If the keys expire, the nodes are required to refresh the keys. The second scenario is, when too many node's trust values drops below a system defined threshold, the key management requires the sensor nodes to refresh their keys in-order to establish a new key. Not all the keys would be refreshed. To refresh a key or establish a new session key, the node will evaluate the trust value of the target node. The keys will be refreshed for only trusted nodes. The intra cluster key is refreshed once a cluster head is corrupted. The refreshing procedure is explained in (Abdallah *et al.*, 2014).

3.2.1 Mini Summary

This is an efficient key management system and a reasonable dynamic scheme which takes into consideration, the limited capabilities of sensor nodes in a wireless sensor network. The key management approach addresses some security threats like node capture and node replication to some degree but no other significant attacks as discussed in Chapter 2. We believe that the key management could be usefully enhanced to address the security issues discussed in Chapter 2, provide a scalable and efficient scheme for wireless sensor network by introducing a trust mechanism.

3.3 Trust Architecture

As mentioned in the previous section. The trust model is to enhance the decisions of a node's key management factors such as: when to generating a key, when to share a key, and when to revoke or refresh a key. Here are some decision parameters needed to be address:

Base Station	Cluster Head	Sensor node
Is a node trustworthy	Is a sensor node trustworthy	Is a sensor node trustworthy
Is a cluster head trustworthy	Should I share a key	Should I share a key
Should a sensor node be allowed to join the network	When to revoke a key	Should I revoke a key
When to eliminate a node	Allowing a node to join the cluster	Should I update a key
When to update the network key	Eliminating a node from the cluster	
When to update the inter-cluster key	When to change the cluster key	
	Is another cluster head trustworthy	

Table 1: Node trust key management decisions

In order to achieve this, we will need to define the trust components needed to efficiently produce a trust outcome.

3.3.1 Design Principles

Our approach is to develop a lightweight trust management scheme for wireless sensor network. With consideration for the limited resources in sensor nodes, the model should use

the minimal possible amount of energy, computation, communication and storage. The trust management system is to provide a collaborative mechanism to identify, isolate and exclude malicious or corrupted sensors from the network. With relation to key management, it guides critical decision-making processes in key management functions. These principles are essential in developing an efficient trust scheme.

Trust can enhance key management in WSNs by allowing nodes (including cluster heads and the base station) to decide with which other nodes to establish keys. Trust depends on examining the past behaviour of nodes to anticipate their likely future behaviour.

We define the trust model as $TM_{WSN\text{-}KM} = (E, TR, OP)$ where E represents the entities of the trust model, TR represents the trust relationships between the entities and OP represents the operations that manage the trust relationships.

Our WSN trust model encompasses three types of entities: base stations, cluster heads and general nodes. The functions of these within the WSN have been discussed above. More detail on their functioning in terms of trust will be given below but, in brief, the base station maintains trust values in the cluster heads and determines, using the trust values, whether a node is fit to remain as a cluster head. If a new cluster head is required, then the base station will select them from other nodes based on recommendations from the remaining cluster heads. Cluster heads manage their clusters and use trust values to determine cluster membership, represented by distribution of the cluster keys. Individual nodes will use trust value to determine pairwise keying arrangements.

More formally,

- BS is the Base station
- CH is the set of clusters heads ch
- N is the set of general sensor nodes n
- Therefore $E = \{BH, CS, N\}$

TR represents the set of all trust relationships between entities of the system. A trust relationship is a 5-tuple $tr = (e_1, e_2, DT, \theta, CT)$ which represents the trust entity e_1 has in entity e_2 and where:

- $e_1 \in$ E and $e_2 \in$ E,
- DT is the value of the direct trust that e_1 has in e_2
- θ is the time at which DT was calculated
- CT is the set of trust components (ct) from which DT was calculated

As will be discussed in Chapter 4, trust for key management in WSNs is best calculated based on a number of components, such as direct key events, communication and energy. CT is a set of component tuples consisting of (CT_{ID}, *evidence*) where

- CT_{ID} identifies the trust component type
- *evidence* is the set of evidence from which the trust value for that component is calculated.

Example trust component types are discussed in Chapter 4. As is also described there the form of evidence for each trust component type may vary. For example, for communication trust, it could be observations of when a node did and did not deliver a message. For energy it could be observations of energy expenditure.

OP is the set of operations that is used to calculate and manage trust. Each evidence type may require a different calculation. There are also the operations used to group the component trust together and calculate a single trust value. These are also discussed in Chapter 4.

3.3.2 Trust Properties

As discussed above our trust model consists of trust relations (tr) between entities (e_1 and e_2). The trust value (DT_{e1e2}) is calculated from a set of trust components (CT). The final trust value that e_1 has in e_2 is calculated form the direct trust value DT_{e1e2} and, potentially, from recommended trust, which is the direct trust value other nodes (e_x)have in the target node (DT_{exe2}). Whether indirect trust is required depends on the confidence that e_1 has in DT_{e1e2}. These concepts are further discussed below:

Direct Trust: One of the best ways to evaluate a trust in a person is through direct contact. You could assess a person's character from previous interactions with the person. This also applies to sensor nodes. Direct trust is comprised of categories and each category ct has a series of events. There are various categories such as communication trust, routing trust, packet trust, energy trust and others that make up direct trust. However, for this model we evaluated our direct trust by considering communication and energy trust as examples of categories.

Data communication is one of the core responsibilities of sensor nodes in a wireless sensor network. Sensor nodes rely on their neighbouring nodes to transfer data to the base station due to their limited transmission range. A target node examines the communication trust of an object node with the Bayesian formula. If the trust value is above the system defined threshold, the node will be classified as trustworthy or untrustworthy if the value is below the threshold.

Energy is part of the main resources available to nodes in WSN. The nodes extremely rely on their battery energy available to them in-order to carry-out tasks in the network. It is important to access the capability of a node before establishing a key and starting communication. A malicious node can either be very active which will lead into a high level of energy consumption or a malicious node can also be highly inactive which will lead to less or no energy consumption. In a case of where a node is highly inactive or selfish, might make the node communication trust value very low. The topic of direct trust will be further discussed in section 4.1:

Direct Trust Confidence: As discussed above, "the best way to know a person is through direct interaction". However, it is also crucial to be confident or sure of your assumptions of the person's character. We employ trust confidence, which is a unique mechanism to reduce computation overhead. This mechanism is the level of confidence a node has in its direct trust evaluation in another node. A general human example is: If Alice and Bob have been friends for 2 years and Alice and Tom have only been friends for 1-week, ideally Alice will trust Bob more than Tom by knowing Bob more than Tom. However, even though this example is about length of relationship, it could be number of interactions with wireless sensor nodes. Direct trust confidence ensures that a subject node might not need to compute an indirect trust evaluation on a target node to evaluate the trustworthiness of the node. There is a system defined confidence threshold which defines if a node can be confident in its trust evaluation or not. The confidence trust level of Alice's direct trust in Tom might be reasonably above the system defined confidence threshold in which, Alice might still be confident in her judgement of Tom. If the direct trust confidence of the subject node is higher than the threshold, the direct trust will be sufficient for the final trust decision otherwise if the direct trust confidence level is lower than the threshold, the subject node will request for recommendations from neighbouring nodes with direct interaction with the object node to evaluate the indirect trust value.

Indirect Trust: Indirect trust computation is an essential part of a trust model approach. With the example in direct trust calculation, a person might not know the other person very well because they have had very few or no interactions. An example would be references. For instance, in a job scenario a referee is required for an applicant to establish that the applicant can be trusted or is capable of carrying out the required duties. The indirect trust in our approach will only be required when a direct trust confidence level is low or when there has been no interaction between the subject node and the target node.

3.3.3 Notion of Time Related Trust

A legitimate node with a high trust value due to previous good interactions might start to alternate between good and bad behaviours. This might be because the node has been corrupted or is experiencing a lack of resources. A malicious node might alternate behaviours not to be detected easily. A malicious node might act as a legitimate node during an event and behave maliciously in another event. We employ time and trust update to mitigate against this issue. Time is divided into time frames and, a time frame is the portion of time within which an event/activity takes place. A sensor node is able to observe the behaviour of another node in a given time frame. The trust value *dt* calculated within a time frame is recorded as the most recent calculation. The trust value calculated within the current time frame *i* is combined with the overall trust value calculated in the previous time frame *i-1* (DT_{i-1}) to get the new overall trust DT_i. A node's historical trust value should be considered to measure its current trustworthiness. In this mechanism, the only historic trust value is the previous updated trust value. The trust update is not at random and neither is it frequent. This is to reserve energy consumption and computation overhead. The trust value is updated when there is a trust value computation. The new updated trust value becomes the most historic trust value (recent trust value). This is recorded, and the previous trust value is discarded. This mechanism uses less storage capacity compared to storing all the previous historic trust value. We will further discuss trust update in chapter 4.

3.4 Chapter Summary

We proposed a network architecture suitable for designing a model that ensures security at every level of a wireless sensor network. In a hierarchical architecture, there is a peer-to-peer network between sensor nodes as well as communication between different category of nodes such as the base station and the cluster heads. The functions of the BS, CH and SN have been discussed in this chapter. The sensor node calculates the trust properties to get the overall final trust of another node. We intend for a scenario where you can apply our approach to different system requirements. In this chapter we discussed the overview of the network architecture. We introduced the model framework and approach. In the next chapter we elaborate more on the model algorithm and techniques.

Chapter 4: Model

As discussed in Chapter 3, the proposed trust based key management model provides a degree of security by enabling a sensor node to estimate a trust value associated with a target: - the probability that a transaction with a target node yields satisfactory results. The proposed model is a lightweight model, with the aim to reduce computation overhead and save energy consumption. The model also ensures that only a small amount of a node resources is required for an efficient implementation by not recording unnecessary information and avoiding multiple computation. The model is designed to provide a level of security in key management for wireless sensor network. In this chapter, we present the trust calculation process in detail. The trust principles are direct trust, direct trust confidence, and indirect trust.

4.1 Direct Trust

Every sensor node has a record management capacity where trust records are stored. Trust records are represented by a tuple R = (O_{ni}, DT_v, R_T) for a subject node X that computes a trust value for node, where O_{ni}, is the object node identity which is node Y. DT_v, is the final trust value and R_T is the recorded timestamp.

As discussed in the previous chapter, the overall direct trust is a function of a number of attributes such as communication trust, energy trust, activity trust, routing trust, key usage and others. The attributes are categories of activity that are relevant in calculating the trust value.

The trust value for each category $\{ct_i, \dots \dots \dots, ct_n\}$ that a subject node X has in the target node Y are calculated and recorded to compute the direct trust value. Where for a category ct_i, $0 < ct_i \leq 1$

The weighting for each category is different and are defined by the system authority. The weighting vector will be defined by the network authority $\langle w_1, \dots \dots \dots \dots, w_n \rangle$

Where

$$\sum_{i=1}^{n} w_i = 1 \,,$$

(5)

To calculate the direct trust node X has in node Y $dt_{x,y}$ is:

$$dt_{x,y} = \sum_{i=1}^{n} w_i * ct_i$$

(6)

We will examine the calculation of a few of these attributes as examples. Other attributes could be included in a real-world implementation.

4.1.1 Communication Trust

In WSNs the nodes often communicate with each other to carry out certain tasks. These nodes share crucial information in most cases and might have to relay data to the base station. In this trust model a node x would check the communication habits of its neighbouring node y to determine if this neighbour node has been acting in a good or a bad manner. A node is considered to be behaving appropriately if it carries out various communication related functions such as accurate routing, complete packet transfer, secured key distribution in a non-malicious way, while a node is considered to be behaving in a malicious or a selfish way if it is dropping packets, presenting false route, re-using shared keys multiple times, and carrying out other malicious behaviour. These functions could also be classified as examples of trust attributes. In the communication trust of this model we use a beta distribution general concept which has two parameters α and β using the Bayesian formula. Communication Trust $ct_{C_{x,y}}$ can be calculated using the Bayesian beta distribution formula. The total number of good communications is represented by α and β represents the total number of bad communications as binary events.

$$ct_{C_{x,y}} = \frac{\alpha + 1}{\alpha + \beta + 2} \tag{7}$$

Where

$$f\ (p \mid \alpha, \beta) = \frac{\Gamma\ (\alpha + \beta)}{\Gamma(\alpha)\ \Gamma(\beta)}\ p^{\alpha-1}(1-p)^{\beta-1}\ , \qquad where\ 0\ \leq p\ \leq 1, \alpha, \beta > 0, \tag{8}$$

p is the probability density parameter
(Zhou, Varadharajan and Hitchens, 2015)

4.1.2 Energy Trust

Energy is an essential resource for sensor nodes. It is important to consider energy as a direct trust category because corrupted nodes often have an abnormal energy consumption rate while carrying out an attack. A malicious node can either be very active which will involve a high level of energy consumption or a malicious node can also be highly inactive which will lead to less or no energy consumption.

In energy trust, we assume that a node is able to observe the activities of its neighbouring nodes that it has a direct interaction with. In this scenario the subject node monitors and records the energy consumption rate of the object node in a particular time frame. For every time frame the energy consumption rate could vary, such that in some situation it might be high. Others might be consistent or very low. We set the highest energy consumption observed for any time frame (time frames were discussed in chapter 3) as E_{max}. E_{max} is the highest energy consumption rate a node has reached throughout the observation period. E_{max} will only change if in the new time frame the consumption rate is higher than the previous E_{max}. We propose two possible energy trust approaches for two scenarios in this attribute:

In the first scenario; a malicious node might attempt to conceal its remaining energy E_R to manipulate the subject node from not knowing the limit of its energy resources. E_c is the full energy capacity of a node. A subject node observing the energy consumption rate is able to estimate the average of the energy consumption rate using the previous observed energy consumption of sensor nodes in a time frame (E_{ave}). E_u is the energy used by the object node in that particular time frame.

$$E_c = E_{ave} * 2^Y \qquad where\ E_{ave} \ll E_c \qquad (9)$$

To calculate ct_E

$$ct_E = \begin{cases} 1 & if\ E_u < E_{ave} \\ 1 - \left(\dfrac{\log_2 \left(\dfrac{E_u}{E_{ave}} \right)}{Y} \right) & otherwise \end{cases}$$

$$where\ Y = log_2 \left(\frac{E_c}{E_{ave}} \right) \qquad (10)$$

If the energy used in the most recent time frame is less than the estimated energy consumption average, we assume that the expected energy consumption in the next time frame would be less than or equal to the consumption average. Thus, making the target node capable of taking carrying out the expected task.

The second scenario is when the subject node knows the remaining energy E_R of the object node. We assume that a node can accurately and precisely estimate the residual energy level of a target node when $E_c = E_R$. In every time frame where energy is used, the energy used is deducted from the remaining energy and only the remaining energy is recorded. In time t

$$E_r = E_r - E_u \qquad (11)$$

A node is considered to be trustworthy to carry out task with enough energy resources when;

$$ct_E = \begin{cases} 1 & E_r \geq E_{max} \\ \frac{(E_r - E_{ave})}{2*(E_{max} - E_{ave})} + 0.5 & E_{ave} \leq E_r \leq E_{max} \\ 0 & otherwise \end{cases} \quad (12)$$

4.1.3 Direct Trust Confidence

The calculation of direct trust by a subject node on a target node may be sufficient for a final trust decision on a node. In most research, such as (Uikey and Bhilare, 2017), requires that the calculation of a direct and indirect trust make up the final trust value of a trust calculation, with the direct trust value carrying more weight than the indirect trust value. The approach in (Uikey and Bhilare, 2017) consumes more computation energy. Other researchers propose that normally a direct trust value associated with a target node would suffice and an indirect trust value is only required when there is no direct communication between the subject node and the object node (Che *et al.*, 2015). This is susceptible to a problem; a node that has had for example, three or less interactions with a target node might not be confident in its judgement of the target node. But it might be fully confident in its direct trust in a target node that it has had 100 interactions with. We propose a direct trust confidence derived from statistical analysis. A threshold for the confidence level is set and if the confidence value is higher than the threshold, the direct trust will be sufficient for the final trust decision otherwise if the confidence level is lower than the threshold, the final trust value would be calculation from both the direct trust and the indirect trust value.

In confidence calculation, the error level ϵ has an effect on how confident a node is in its direct trust calculation for another node (Patel *et al.*, 2005).

$$\gamma_{x,y} = \frac{\int_{dt_{x,y}+\epsilon}^{dt_{x,y}-\epsilon} p^{\alpha-1}(1-p)^{\beta-1}\, dp}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}\, dU}$$

(Patel *et al.*, 2005)(Che *et al.*, 2015) *(13)*

where p is the probability value of the outcome of $dt_{x,y}$ $\quad p \in |0,1|$

The value calculated is compared to the threshold described above and the node can assess whether it has sufficient confidence in its direct observations. If the confidence value is higher than the threshold, the direct trust will be sufficient for the final trust decision but if the confidence level is lower than the threshold, an indirect trust evaluation will be required.

4.1.4 Trust Update

As discussed in Chapter 3, trust update uses the most recent trust history and the current trust value to calculate the new direct trust value. In this model, we recommend that the trust value should not be updated too often to save energy consumption. A frequent update of trust values when they are not required will create a high computation overhead and a high energy consumption rate. Instead we suggest that the trust value of a sensor node associated with a target node is updated only when a transaction is initiated with the concerned target. The historical trust evaluation of a node should be considered to assess its. A node can be malicious for a long time and begin to act like a legitimate node only in the recent time slot. To solve these issues, we use a time frame concept to update the trust value wherein the historical trust value of the previous time frame from a certain time to recent time is considered. The sensor node only remembers the most recent direct trust value which is in the previous time frame. In the current time frame, the trust value is updated as:

$$DT_{x,y} = w_1\, dt_{x,y} + w_2\, DT_{x,y-1} \qquad (14)$$

$$\text{where} \qquad w_1 + w_2 = 1$$

w_i and w_2 are the weight values of the previous trust and the current trust value. We consider the recent trust value to be of more importance than the previous trust value. An aging factor is defined for trust value attenuation: $\beta = e^{dt_{x,y} - dt_{x,y-1}}$, where $dt_{x,y}$ and $dt_{x,y-1}$ are both the time for the trust calculation of $dt_{x,y}$ and $DT_{x,y-1}$. Therefore, the weight value is $w_1 = \beta$, $w_2 = 1 - \beta$

4.2 Indirect Trust

Indirect trust is an essential trust property as discussed in Chapter 3. There are two scenarios where an indirect trust calculation will be required; If the direct trust confidence level is below the confidence threshold, the subject node will request for indirect trust values from neighboring nodes on the object node because the direct trust is not satisfactory, or when a node *x* wants to interact with node *y* but has no previous interaction history with this node. Node *x* will request for recommendations for node *y* from its neighboring set of trusted nodes *v* who may have previously interacted with this node *y*. Node *x* will then be able to compute an indirect trust with the help of these recommendations. The indirect trust value of node *x* for node *y* is the recommendation from a set of trusted neighbor nodes *v* which is the direct trust value of *v* for node *y*.

We have to consider that, not all the recommenders will give an honest recommendation. There could be malicious nodes utilizing the recommendation system to carry out a good-mouth or bad-mouth attack to manipulate the trust values. Also, not all the recommendations are considered relevant and there will be a decision factor to determine the relevant recommendations. Some integrity checks are needed to combat false recommendations.

Due to the nature of wireless sensor networks, an issue to consider when developing a model is the limited energy resources available to the sensor nodes. A sensor node that computes a large number of recommendations will use up a lot of energy. Some previous research work has suggested that a small number of trusted recommendations could be sufficient, rather than a large number of them which may include malicious or false recommendations (Hasan *et al.*, 2009). In this model a sensor node will be able to select a set of trusted recommenders with reliable recommendations for initial evaluation. The node can gradually add more recommenders in its evaluation if needed. The selection of a set of trustworthy nodes is the first step in the indirect trust calculation. There are several factors to consider in the selection of such nodes. The first factor in the recommendation process is the number of interactions a recommending node has had with the object node. The subject node broadcasts the recommendation requests on the object node with specific instructions; a number of interactions parameter, where only the recommenders who meet the interaction parameter will respond to the broadcast message with their recommendations for the object node. Some nodes might have had 100 interactions with the object node while others might have only had less than 10 interactions with the object node. In Steps:

- The subject node broadcasts a request with an interaction threshold single-hop of its broadcast range.
- If the subject node receives a low number of recommendations, it will re-broadcast the message across its cluster to reach a satisfying number of recommendations from trusted recommenders'.

The interaction threshold is adjustable by the system (we assume that, if the required number of interactions are 80, the threshold could start at 100 depending on the required parameters). This might not be the quickest approach, but it saves energy by not having to broadcast across multi-hop at first. If there are no responses, considering that it might be new node, the cluster head will respond to the cluster broadcast with the node details including the starting trust value of the sensor node. The cluster head can also request information from other cluster head or the base station if it does not have any record of the network node.

Once the subject nodes receive the recommendations from the recommenders, it will evaluate the recommenders based on the value of the direct trust the subject node has in its recommenders. The subject node will select only the recommendations from trusted nodes. There are two steps in the selection process for the set of trusted recommenders: The credibility of the recommenders is the first step, which is the overall trust a subject node has in the recommending node while the reliability check looks at the reliability of the recommendation from the credible recommending nodes. It is relevant to consider that some might be giving wrong recommendations for other nodes. This could be a form of attack or miss-information.

4.2.1 Credibility:

An example of recommender credibility is where, the subject node x evaluates its direct trust in the recommenders. If the direct trust of x in the recommender is above the trust threshold parameter θ, represented by

$$DT_{x,v} > \theta \quad (15)$$

Then the recommender is considered as a trusted recommender.

Because the subject node is not considering recommendations from all the neighbouring nodes, the computational overheads are lower and do not significantly impact the energy consumption at the node. In addition, not all the neighbouring nodes have to respond to the broadcast because the interaction threshold also reduces energy consumption of the nodes in the network. An attacker that is able to create multiple new nodes won't be able to have those nodes meet the requirement above for the direct trust in the recommender. This also avoids Sybil attacks by not pulling all or a-lot of recommendations. Only trusted nodes are considered for evaluation. The recommendations from the trusted nodes are also accessed in recommendation reliability.

4.2.2 Reliability of Recommendations

The second step as discussed is to evaluate the reliability of the recommendations from the selected credible recommenders. Once the subject node has selected the credible recommenders from the nodes that responded with a recommendation it will then check the reliability of the recommendations from these credible recommenders. It is essential to verify recommendations and be able to isolate false recommendations as not all recommendations from credible recommenders could be considered reliable. A simple checking method for

calculating the reliability of the recommenders is to compare the recommendation from each node to the average of other recommendations as:

$$R_{v_i} = 1 - |DT_{v_i,y} - DT_{ave,y}| \quad \text{where } DT_{v_i,j}\text{, is the value of the recommendation from recommender } v_i \text{ for object node } y \text{ and } DT_{ave,y}\text{, is the average value of the recommendations.} \tag{16}$$

For the evaluated recommendations where recommendation reliability is greater than the system defined parameter $R_v > \delta$, the recommendations are added to the set of trusted recommenders $V = V_1, V_2, V_3 \dots \dots \dots \dots V_n$.
The indirect trust calculation of node x for node y is calculated as

$$ID_{x,y} = \frac{\sum_{v=1}^{n} ID_{v,y}}{n} \tag{17}$$

Where

$$ID_{v_i,y} = \begin{cases} DT_{x,v_i} * DT_{v_i,y} & if\ DT_{v_i,y} < \gamma \\ \gamma + (DT_{x,v_i} - \gamma) * DT_{v_i,y}, & else, \end{cases} \tag{18}$$

Where DT_{x,v_i} is the direct trust of node x for recommending node v_i and $DT_{v_i,y}$ is the direct trust of recommending node v_i for object node y. γ is the threshold. The indirect trust is computed for all n number of nodes in V

4.2.3 Badmouth/Good-mouth Recommendation Detection

In this model we implement a recommendation density model which was adopted from (Noor *et al.*, 2016). This model deploys feedback with occasional collusion detection, where a feedback threshold was defined for the number of feedbacks from a user for a particular service provider to prevent a feedback collusion. In this model we introduce a different approach which is to compare the previous recommendation to the current recommendation denoted as the delta Δ of the recommender. This approach isolates any nodes that may have recently been corrupted. A new recommendation is required when a pair of sensor nodes are refreshing their keys in situations wherein the direct trust confidence level is low between these two nodes. The subject node remembers the most recent recommendation v_1 and compares it with the new recommendation v_{1+1} . The delta of each sensor nodes will be compared to the average

delta of the other trusted recommenders to evaluate the changes. If a recommender is acting differently to other recommenders, a recommendation reliability will be computed again as;

$$\Delta_{v,y} = | DT_{v,y} - DT_{v,y+1}| \quad (19)$$

and

$$\Delta_{ave,y} = \frac{\sum_{i=1}^{n} \Delta_{v,y}}{n} \quad (20)$$

For the evaluated recommendations where $|\Delta_{v,y} - \Delta_{ave,y}| < \delta$ are added to the set of trusted recommenders $V = V_1, V_2, V_3 \ldots\ldots\ldots V_n$.

4.3 Trust Contribution to Key Management

As mentioned in Chapter 3, the trust model is to enhance the decisions of a node's key management functions such as when to generating a key, sharing a key, key distribution and when to revoke or refresh a key. The key management scheme has been enhanced to a trust driven scheme. In this approach, a sensor node can evaluate the level of trust associated to a subject node before establishing any of the key management functions. We have introduced a lightweight direct trust framework which establishes a trust value based on previous interaction using a simple Bayesian beta distribution formula. This could include assessing the way a target node used previously shared keys. A node will only establish key functions with trusted nodes in the network. The trust value for a target node will need to be updated for any period of a new key management function. The lightweight direct trust process ensures that a node can assess the confidence level it has in its trust evaluation before establishing a key. This means the trust driven key establishment process is not time consuming and uses less resources. The indirect trust model has a security intrinsic design. It avoids some attacks by eliminating malicious nodes and malicious or false recommendations. The model uses node credibility to assess the trust value of a recommending node that have had several interactions with the target node (including shared keys and key usage). It applies a recommendation reliability check to verify the accuracy of the recommendation. The trust management provides a node level security in key management and a cluster head level security. The cluster head can eliminate a malicious node from its cluster and change the cluster key. Cluster heads can evaluate the trust level of other cluster heads with the same approach. A cluster head will relay a malicious cluster to the base station. The base station will eliminate the entire cluster, re-key the inter-cluster key and change the network key.

4.4 Chapter Summary

In this chapter, we presented a lightweight model that can be implemented on key management schemes in wireless sensor networks. We also wanted a trust management model which could assist across different WSNs application. The model uses a Bayesian formula for WSNs in the direct trust calculation. We introduced various mechanisms to ensure that the trust management model is a lightweight model. One of the mechanisms is trust confidence, where the model might not require an indirect trust calculation. In this approach, an indirect trust calculation is only needed either when the trust confidence is low or there are no previous interactions. Another lightweight approach is using the most recent updated trust history to update the current trust record. By not having to store all the previous trust records, we reduce the use of memory space in this model. Another approach if the indirect trust is required is the number of interactions threshold. The trust model identifies and isolates malicious nodes in the recommendation system. In the next chapter, we will implement some of the approaches and simulate them. We will discuss the results of the simulation.

Chapter 5: Simulation

In this chapter, we implement the proposed trust management model in order to analyse the performance of the trust approach in a wireless sensor network. We assess different simulation parameters such as the trust calculations which includes the direct trust model and the indirect trust approach, the energy consumption rate, malicious node detection and malicious node elimination from the network. We initially started the code simulation with MATLAB to verify the mathematical algorithms in the trust calculations. We then simulated the wireless sensor network using NS2.

Direct trust evaluation example

We calculate the communication trust of each node using equation *(7)*. We then evaluate the confidence level of the subject node in its communication trust of the direct trust model in Chapter 4 using equation *(13).* The confidence interval threshold is usually high because it is important for a node to be close to extreme confidence in its trust values. We assume the confidence threshold from the network operator is 0.95.

Node	Successful Communication	Failed Communications
Y_1	12	2
Y_2	2	15
Y_3	2	0

Table 2: Node X past interaction

As shown in *Table 2* and *Figure 5*, the trust value the subject node has in a target node does not influence the confidence interval of the node's confidence evaluation. However, the number of interactions would affect the confidence level of a target node like the example of Alice, Bob and Tom discussed in chapter 3. Node X has only had 2 interactions with node Y_3, compared to the other nodes. Given that the confidence level is below the confidence threshold, node X will request recommendations from nodes who have had interactions with node Y_3.

Node	Communication Trust Value	Confidence Level
Y_1	0.81	0.97
Y_2	0.16	0.98
Y_3	0.75	0.63

Table 3: Communication trust and confidence interval calculations

5.1 Simulation Parameters

Parameter	Value
Number of Nodes	100
Range	1000 x 1000
Deployment	Random
Application	FTP
Simulation time	10 minutes
Transmission range	250
Interference range	55
Mobility model	waypoint

Table 4: Communication trust and confidence interval calculations

Parameter	Value
Mac protocol type	Mac/802.15.4
Link layer type	LL
Antenna type	Antenna/OmniAntenna
Routing protocol	AODV
Propagation	Two Ray Ground
Data Rate	250 kbps
Freq	2.472e9

Table 5: Wireless parameters

The sensor nodes are deployed at random over an area range of 1000 x 1000 meters. There are 100 sensor nodes within the network range in this simulation as shown in *Figure 4*. The sensor nodes in the network are mobile and might move during simulation from one point to another. The wireless parameters are represented in *Table 5*. The implementation simulation time is 10 minutes.

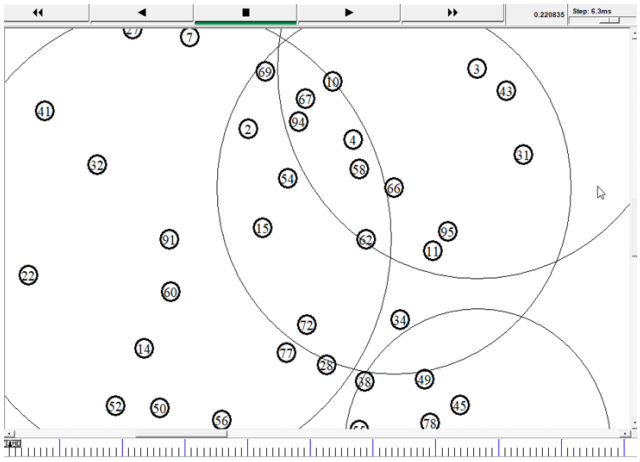

Figure 4: Representation of nodes deployed randomly in WSN.

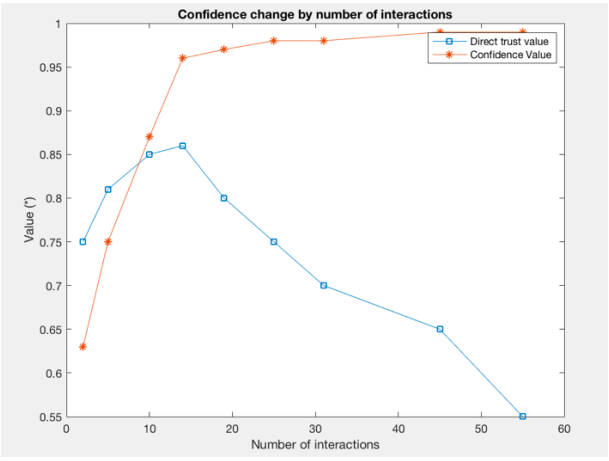

Figure 5: Confidence change by number of interactions

5.2 Simulation Results

We analyse the results of the simulation of the proposed model. The data represented in *Figure 5,* shows the effect the number of interactions has on the confidence level. The number of nodes simulated is 100. The threshold was set at 0.95 for this simulation. The computation as depicted in *Figure 5*, presents that number of interactions required within 100 nodes to satisfy a high level of confidence was 13 or more. We can see that there is also confidence in the rate at which a node can evaluate another node as untrustworthy,

A: Energy consumption rate

In *Table 6,* we present the energy distribution code and the model pseudo code. The data in *Figure 6,* represents the global energy consumption rate of the network over the simulation time (10 minutes). We assume the network life span is set at 1000 joule. In the proposed model the energy consumption required for an efficient distributed network is low. The approach does not require that much energy usage from the sensor nodes. The model might be able to identify malicious nodes by observing their energy consumption in such a lightweight approach.

Energy Distribution Code	**Energy Trust Pseudo code**
energyModel $val(energymodel) -*set* energy (1-99) 10 -*set* initial Energy 10 -*set* rxPower 0.5 -*set* txPower 1.0 -*set* idle Power 0.0 -*set* sense Power 0.2 -agent Trace ON -router Trace ON -movement Trace ON # BS creation *set* BS energy (0) 1000 $ns node-config -initial Energy 1000 -*set* rxPower 0.5 -*set* txPower 1.0 -*set* idle Power 0.0 -*set* sense Power 0.2	**Scenario 1, No Energy Residue knowledge** {Step 1} – Observe the energy used in time frame. Record average -Full capacity = E average * $2^{\text{defined Y}}$ Where E average (Eave) < Full capacity {Step 2} Check energy used, $Energy\ trust = 1\ if\ energy\ used\ (Eu) <\ E\ average$ Otherwise 1-($\log_2$ (energy used/Average use) / Defined Y) **Scenario 2, Energy Residue known** {Step 1} Define E_{max} , Assume Full capacity = Energy residue -energy residue – energy used = new energy residue {Step 2} $Energy\ trust = 1\ if\ energy\ residue\ Er <\ E\ average$ {Step 3} If E$ave\ \leq\ Er \leq E_{max}$ Do $\frac{(E_r - E_{ave})}{2*(E_{max} - E_{ave})} + 0.5$ Otherwise 0

Table 6: Energy code

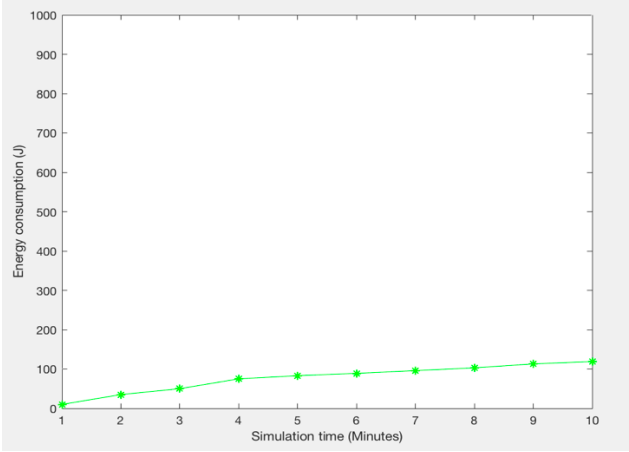

Figure 6: Global energy consumption

B: Malicious node detection rate

In this section we present the results of the rate at which the proposed model could detect malicious nodes effectively. It is essential for the health of a network, to detect malicious nodes before they carry out significant attacks. Some security measures attempt to detect nodes after an attack to identify which nodes carried out the attack. By this time the network might be unrecoverable. We simulated the network with a number of malicious nodes from 2% of the total number of nodes to 50%. We evaluated the average speed/rate at which the malicious nodes will be detected. *Figure 7* presents the result of the simulation. We can see that the proposed model is robust against malicious node as the detection rate is high, even with 50% malicious nodes almost 90% of them are detected.

We can also notice that the increase of malicious nodes within the network reduces the detection rate of the malicious node. This will continue to reduce if the malicious nodes in a network are higher than legitimate nodes in that network. We illustrate this as an example, which is one of the issues the proposed model can avoid from happening. We simulated different percentage of malicious nodes in a cluster. *Figure 9* shows the trust value of the cluster given the degree in percentage of malicious nodes in the cluster. With 10% of malicious nodes, the average trust value within the cluster is high. As the percentage increases so does the average trust value of the cluster. In this trust model, the malicious nodes will be detected at an early stage to prevent further attacks. When the average trust level of a cluster is low, the BS might eliminate the cluster from the network. The BS also eliminates malicious nodes from the network to prevent them from corrupting other nodes in the network.

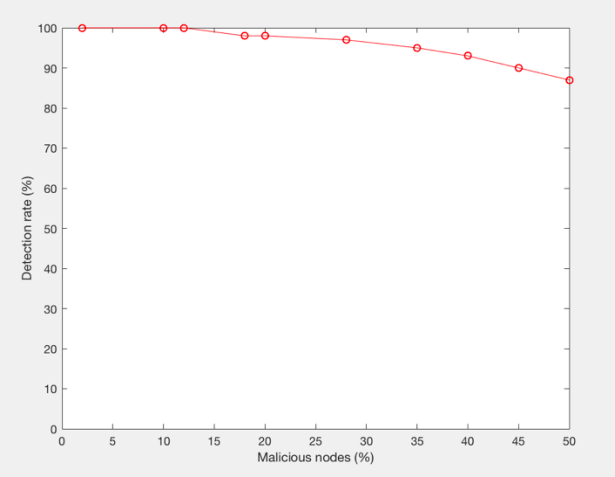

Figure 7: Malicious node detection rate

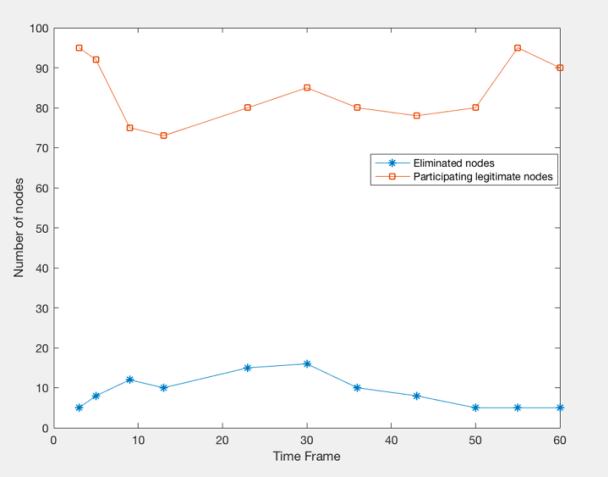

Figure 8: Malicious node Elimination

C: Node Elimination from the Network

We simulated malicious nodes to eliminate them from the network key distribution (re-keying process). In this framework, when a malicious node is detected, the base station prompts a re-

keying system. This system is to revoke the keys associated with the corrupted node and eliminate the node from the network. In the simulation we also implemented the participating nodes in a given time frame of the re-keying process. The simulation result shown in *Figure 8,* shows the malicious nodes in a time frame and the participating legitimate node in that time frame. This means during the node elimination; the network is still healthy and operating efficiently.

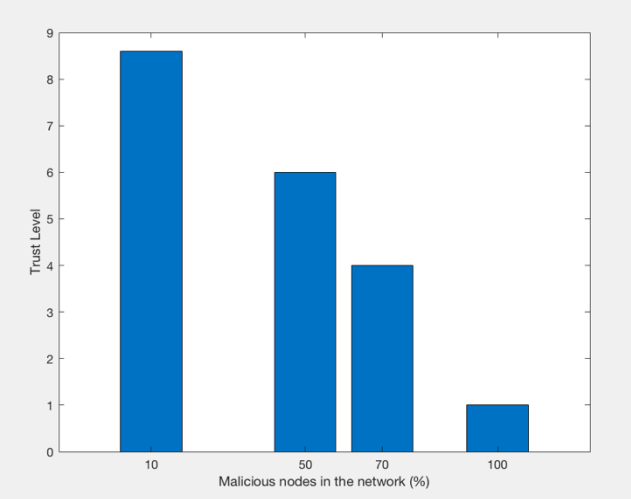

Figure 9: Percentage of malicious node

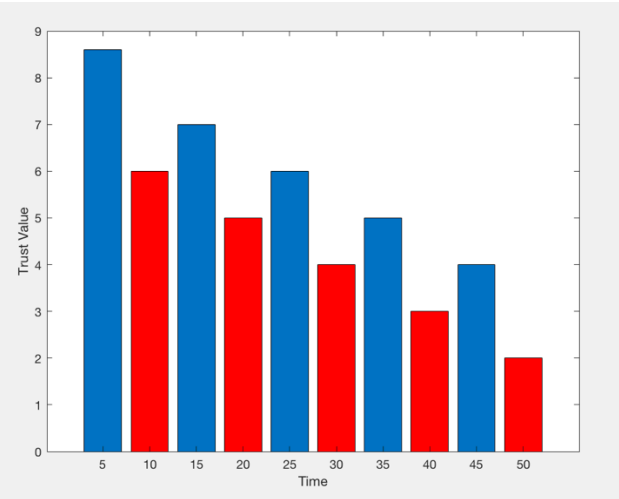

Figure 10: Good/Bad Interactions

5.3 Security Measures

The trust based key management system is essentially meant to provide a level of security for wireless sensor network. The proposed key management mechanism provides a level of security for data communication in WSNs. The trust model is to enhance the sensor nodes to be robust against other forms of attacks discussed in Chapter 2. These attacks are node attacks, network attacks, key management attacks and trust attacks. We will discuss in summary how the trust model achieves the security for WSNs against some of the security threats in Chapter 2. We draw conclusions on this from the model in Chapter 4 and the simulations in this chapter.

A. **Robustness against Sybil Attack:** as discussed in Chapter 2, this is a significant attack in WSNs. In Chapter 4 (indirect trust section), we discussed an interaction threshold. An attacker that is able to create multiple new nodes won't be able to have those nodes meet the requirement above for the direct trust in the recommender. This also avoids Sybil attacks by not pulling all or a-lot of recommendations. Only trusted nodes are considered for evaluation. The recommendations from the trusted nodes are also accessed in recommendation reliability. In the node detection experiment result in *Figure 7,* we simulated nodes with false identities attempting to establish keys with legitimate nodes. We

can see that the detection rate of the model is at an early stage with a high detection rate. The legitimate nodes are able to detect the malicious nodes, and in theory will relay the information to the cluster head, the cluster head eliminates the node from its cluster, change the cluster key, relay the information to the base station. The base station will eliminate the node from the network, then change the network key.

B. **Denial of Service Attack:** This attack and how it works was explained in chapter 2. DoS can be identified using the communication trust model in a direct trust evaluation, where these examples would be classified as bad behaviours which will affect the communication trust value of the malicious node. When detected, the node will be isolated. Sensor nodes will not associate with the target node. The node might be eliminated from the cluster and eventually from the network.

C. **Robustness against Node Capture:** a node capture is a physical attack as discussed in chapter 2. In this key management approach, the keys are not generated and stored in the sensor nodes before deployment. The public and private key is generated after deployment as discussed in Chapter 3. The key management approach is robust against node capture because there is less information stored in a sensor node. The attacker would be unable to establish keys with other nodes. The pairwise keys are only established if the level of trust associated with a target node is sufficient. If an attacker attempts to integrate a captured node into the network or fabricate the identity of the node, it will be detected easily. Firstly, the base station will need to admit the node back into the network, which might not be possible. If it does, the cluster head will access the node before the sensor nodes evaluate the node with all the trust properties for validation.

D. **On/Off Attack:** could be very deceptive in the sense that a node might act good in a time frame and bad in another time frame as discussed in chapter 2 and shown in *Figure 10*. In chapter 3, we discussed time and also the advantages of trust updates in new interaction time frame. This mechanism helps to identify an on/off attack as a node fluctuating in behaviours will also find it difficult to have a high trust value unless it acts in a good manner for a consistent amount of time frames. As we can see in *Figure 10,* the trust value of such node continues to depreciate as the update factor uses the previous and current trust value to update the final trust value. Also, a malicious node performing an on/off attack might receive the same level of good and bad recommendations from other nodes. This will put

the indirect trust value below the system defined threshold which will be insufficient for establishing a key.

E. **Badmouth/Good-mouth Recommendation Detection: -** the attack was explained in chapter 2. The mitigation approach was evaluated and discussed in chapter 4. The indirect trust mechanism evaluates the credibility of a target node by comparing the nodes trust value to a system defined threshold. The model also checks the reliability of a recommendation from a credible node at every given time frame a recommendation is required. This avoids a scenario where, a legitimate node with good credibility in previous time frame becomes corrupted in the next recommendation time frame and attempts to give false recommendations.

5.4 Analysis

In summary, we have listed a number of trust questions to illustrate the performance of the model.

1. *How does the key management react to an attack detected by the trust mechanism*: When an attack is detected, the key management refreshes the keys in all sections of the network. If there is a suspected malicious node or attack from a node, the key management will rely on the trust mechanism to evaluate the level of threat to the network.
2. *How do you differentiate between a legitimate node under pressure and a malicious node*: in this model we assume that, legitimate nodes under pressure can be categorised as compromised node. If a node under pressure begins to act maliciously by dropping packets, or acting selfishly to reserve resources, will be classified as not a network efficient behaviour. In communication trust evaluation, such behaviours will be classified as bad behaviours which will affect the trust value of such nodes.
3. *What are the mechanisms for eliminating, changing a cluster head if compromised*: the base station manages the cluster heads, and maintains the trust values in cluster heads and determines whether a node is fit to remain as a cluster head by using the trust values as discussed in chapter 3. If a cluster head is comprised, the base station will eliminate the cluster head, send out a revocation message to all the cluster heads in the network. The base station will refresh the cluster key and the network key. The base station will select a new cluster head from other nodes in a legitimate cluster based on recommendations from the remaining cluster heads.

4. *How does the cluster head keep track of a nodes trust, how does it identify a malicious node*: as discussed in chapter 3, cluster heads manage their clusters, they have the capability of monitoring the nodes in their clusters. The cluster heads also rely on recommendation from nodes in their network. They use the indirect trust model in chapter 4 to access the recommendations. Sensor nodes might report a node to the cluster head as malicious or legitimate (Good-mouth, badmouth attacks could occur in a case of a malicious node reporting).
5. *How does trust play a role in generating or revoking a key*. The trust model ensures that before a node generates a key with target node, the node must have full trust in the target node. The trust values are not included in the key generation module. If a node shares a key with a target node, and target node starts to behave maliciously. The subject node will be able to detect this from the update module required for every given time frame of a transaction as discussed in chapter 3. Once the malicious node is detected, the subject node will seize communication, update its trust values, send a revocation message, change its keys and relay the information to the cluster head, and the node will be cut out of the cluster and eventually out of the network.
6. *If a node in a cluster is malicious, how do you revoke the key and what role does trust play in this*: Similar but not the same with the previous question the cluster head sends a revocation message, eliminates the malicious node from the cluster and refreshes the intra-cluster key. The trust model plays a role in identifying the malicious node in the cluster by its track records and trust values.
7. *How does communication across clusters/groups work:* As discussed in Chapter 3, there is a cluster key in the model which is a general key shared within a cluster to enable communication within a cluster. The cluster key is created once a cluster is established and each cluster have different cluster key. For communication across clusters, there is an inter cluster-key shared between the cluster heads and the base station.

5.5 Chapter Summary

In this section we implemented the proposed model with a simulation using MATLAB and NS2. We analysed the data and represented them in graphs and tables. We have illustrated the confidence level approach and the effects of malicious nodes in wireless sensor networks. It is essential for the health of a network, to detect malicious nodes before they carry out significant attacks. We simulated malicious nodes to eliminate them from the network key distribution (re-keying process).

Chapter 6: Conclusion

Security for wireless sensor networks has been a significant challenge. The aim of this research was to develop a trust model to provide a high level of security against some security threats and enhance the decisions of a node's key management factors of a key management scheme. In this Chapter we reflect on the works done in this research.

Securing data communication channels require the establishments of shared encryption keys and considering the limitation of such networks it is vital that the underlying computation capabilities of sensor nodes must be taken into consideration in the development of any new novel key distribution and management protocol. The difficulty in implementing key management factors in wireless sensor networks are greater than generic networks with infrastructures. The proposed lightweight trust based key management model provides data communication security through an efficient key distribution model, ensuring the generation and distribution of keys using an elliptic curve key encryption and a Diffie-Hellman key exchange protocol which uses less computation overhead to ensure efficient and secure key generation and exchange in different node categories of the network. Trust management is an essential malicious node detection system especially in wireless sensor networks due to the nature of the sensor nodes. The indirect trust model has a security intrinsic design. It avoids some attacks by eliminating malicious nodes and malicious or false recommendations. The model uses node credibility to assess the trust value of a recommending node that have had several interactions with the target node. It applies a recommendation reliability check to verify the accuracy of the recommendation. The trust model provides a degree of security by enabling a sensor node to estimate a trust value associated with a target node: - the probability that a transaction with a target node yields satisfactory results. The model also ensures that only a small amount of a node resources is required for an efficient implementation by not recording unnecessary information and avoiding multiple computation with an example of the direct trust confidence level in Chapter 4. The model is designed to provide a level of security in key management for wireless sensor network.

The proposed scheme was implemented using MATLAB and NS2 simulation tool, and the simulation results shows the performance of the proposed model including the security measures against the security threats.

6.1 Future Work

There have been a lot of key management schemes proposed for WSNs and a good number of trust models for security in WSNs. The research undertaken for a trust driven key management system are not as much. A significant issue with models designed for WSNs is the resource consumption and computation overhead. It is difficult to develop a lightweight key management incorporated with a lightweight trust mode. We proposed a lightweight trust-based key management scheme in a limited time frame and as such could not design a novel lightweight key management scheme. This research can be enhanced with the design of a lightweight key management approach which is trust driven and tailored specific attacks in wireless sensor networks and or Internet of things (IOT).

Chapter 7: Bibliography

1. Abdallah, W. *et al.* (2014) 'An Efficient and Scalable Key Management Mechanism for Wireless Sensor Networks', 3(4), pp. 480–493.
2. Alajmi, N. (2014) 'Wireless Sensor Networks Attacks and Solutions', *IJCSIS) International Journal of Computer Science and Information Security*, 12(7), pp. 37–40.
3. Che, S. *et al.* (2015) 'RESEARCH ARTICLE A lightweight trust management based on Bayesian and Entropy for wireless sensor networks', (March 2014), pp. 168–175. doi: 10.1002/sec.
4. Coles, P. J., Metodiev, E. M. and Lu, N. (2016) 'Key Distribution', (May), pp. 1–9. doi: 10.1038/ncomms11712.
5. Du, W. *et al.* (2004) 'A key management scheme for wireless sensor networks using deployment knowledge', *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. doi: 10.1109/INFCOM.2004.1354530.
6. Feng, H., Kuan, H. and Hao, M. (2010) 'S-MAODV: A trust key computing based Secure Multicast Ad-hoc on Demand Vector routing protocol', in *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*. doi: 10.1109/ICCSIT.2010.5563941.
7. Gandino, F., Montrucchio, B. and Rebaudengo, M. (2014) 'Key Management for Static Wireless Sensor Networks With Node Adding', *Industrial Informatics, IEEE Transactions on*, 10, pp. 1133–1143. doi: 10.1109/TII.2013.2288063.
8. Han, G. *et al.* (2014) 'Management and applications of trust in Wireless Sensor Networks: A survey', *Journal of Computer and System Sciences*. Elsevier Inc., 80(3), pp. 602–617. doi: 10.1016/j.jcss.2013.06.014.
9. Hasan, O. *et al.* (2009) 'Elimination of Subjectivity from Trust Recommendation', *Trust Management III, IFIP Advances in Information and Communication Technology*, (March), pp. 65–80.
10. Hoffman, K., Zage, D. and Nita-Rotaru, C. (2009) 'A survey of attack and defense techniques for reputation systems', *ACM Computing Surveys*, 42(1), pp. 1–31. doi: 10.1145/1592451.1592452.
11. Huang, J. and Nicol, D. (2009) 'A calculus of trust and its application to PKI and identity management', in *Proceedings of the 8th Symposium on Identity and Trust on the Internet - IDtrust '09*. doi: 10.1145/1527017.1527021.
12. Ishmanov, F. *et al.* (2013) 'Trust management system in wireless sensor networks:

design considerations and research challenges', (June 2013), pp. 107–130. doi: 10.1002/ett.

13. Jiang, J. *et al.* (2015) 'An Efficient Distributed Trust Model for Wireless Sensor Networks', 26(5), pp. 1228–1237.
14. Karlof, C. and Wagner, D. (2003) 'Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures', *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2--3), pp. 293–315.
15. Kaur, J., Gill, S. S. and Dhaliwal, B. S. (2016) 'Secure Trust Based Key Management Routing Framework for Wireless Sensor Networks', *Journal of Engineering*, 2016. doi: 10.1155/2016/2089714.
16. Kaur, P., Kumar, M. and Bhandari, A. (2017) 'A review of detection approaches for distributed denial of service attacks', *Systems Science & Control Engineering*. Taylor & Francis, 5(1), pp. 301–320. doi: 10.1080/21642583.2017.1331768.
17. Khatri, P. (2014) 'Using identity and trust with key management for achieving security in Ad hoc Networks', in *2014 IEEE International Advance Computing Conference (IACC)*, pp. 271–275. doi: 10.1109/IAdCC.2014.6779333.
18. Koblitz, N. (1987) 'Elliptic Curve Cryptosystems', 48(177), pp. 203–209. Available at: http://www.jstor.org/stable/2007884.
19. Kumar, D. *et al.* (2013) 'Clustering Algorithms for Heterogeneous Wireless Sensor Networks - A Brief Survey', *IET Wireless Sensor Systems*, 00(3), pp. 914–919. doi: 10.1016/j.procs.2013.06.125.
20. Kumar, G. (2016) 'Denial of service attacks – an updated perspective', *Systems Science & Control Engineering*. Taylor & Francis, 4(1), pp. 285–294. doi: 10.1080/21642583.2016.1241193.
21. Kumar, V., Jain, A. and Barwal, P. N. (2014) 'Wireless Sensor Networks: Security Issues , Challenges and Solutions', *International Journal of Information & Computation Technology*, 4(8), pp. 859–868.
22. Law, Y. W. *et al.* (no date) 'A Formally Verified Decentralized Key Management Architecture for Wireless Sensor Networks', 34734, pp. 1–13.
23. Liu, D., Ning, P. and Li, R. (2005) 'Establishing pairwise keys in distributed sensor networks', *ACM Transactions on Information and System Security*, 8(1), pp. 41–77. doi: 10.1145/1053283.1053287.
24. Lopez, J. *et al.* (2010a) 'Trust management systems for wireless sensor networks: Best practices', *Computer Communications*. Elsevier B.V., 33(9), pp. 1086–1093. doi:

10.1016/j.comcom.2010.02.006.

25. Lopez, J. *et al.* (2010b) 'Trust management systems for wireless sensor networks: Best practices', *Computer Communications*. Elsevier, 33(9), pp. 1086–1093. doi: 10.1016/J.COMCOM.2010.02.006.
26. Lu, K. *et al.* (2008) 'A Framework for a Distributed Key Management Scheme in Heterogeneous Wireless Sensor Networks', 7(2), pp. 639–647.
27. Mansour, I., Chalhoub, G. and Lafourcade, P. (2014) 'Evaluation of Secure Multi-Hop Node Authentication and Key Establishment Mechanisms for Wireless Sensor Networks', *Journal of Sensor and Actuator Networks*, 3(3), pp. 224–244. doi: 10.3390/jsan3030224.
28. Mansour, I., Chalhoub, G. and Misson, M. (2014) 'Security Architecture for Multihop Wireless Sensor Networks', *Security for Multihop Wireless Networks*.
29. Miller, V. S. (1986) 'Use of Elliptic Curves in Cryptography', in Williams, H. C. (ed.) *Advances in Cryptology --- CRYPTO '85 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 417–426.
30. Newsome, J. *et al.* (2004) 'The sybil attack in sensor networks', *Proceedings of the third international symposium on Information processing in sensor networks - IPSN'04*, p. 259. doi: 10.1145/984622.984660.
31. Noor, T. H. *et al.* (2016) 'CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services', *IEEE Transactions on Parallel and Distributed Systems*, 27(2), pp. 367–380. doi: 10.1109/TPDS.2015.2408613.
32. Patel, J. *et al.* (2005) 'A Probabilistic Trust Model for Handling Inaccurate Reputation Sources', *Third International Conference on Trust Management*, pp. 193–209. doi: 10.1007/b136639.
33. Pathan, A.-S. K., Lee, H.-W. L. and Hong, C. S. (2006) 'Security in wireless sensor networks: issues and challenges', *8th International Conference Advanced Communication Technology*, 2, pp. 1043–1048. doi: 10.1109/ICACT.2006.206151.
34. Pietro, R. Di, Mancini, L. and Mei, a (2003) 'Random key-assignment for secure wireless sensor networks', *on Security of ad hoc and sensor*, pp. 62–71. Available at: http://portal.acm.org/citation.cfm?id=986868.
35. Pirzada, A. A. and McDonald, C. (2004) 'Establishing trust in pure ad-hoc networks', *Proceedings of the 27th Australasian conference on Computer Science*, 26(c), pp. 47–54. doi: 10.1007/s11277-006-1574-5.
36. Seo, S. *et al.* (2015) 'Effective Key Management in Dynamic Wireless Sensor Networks',

IEEE Transactions on Information Forensics and Security, 10(2), pp. 371–383. doi: 10.1109/TIFS.2014.2375555.

37. Shamir, A. (1979) 'How To Share a Secret', *Communications of the ACM (CACM)*, 22(1), pp. 612–613. doi: 10.1007/BF02816138.
38. Simplício, M. A. *et al.* (2010) 'A survey on key management mechanisms for distributed Wireless Sensor Networks', *Computer Networks*. doi: 10.1016/j.comnet.2010.04.010.
39. Singh R, Singh D J, S. D. R. (2016) 'Sybil Attack Countermeasures in Wireless Sensor Networks', *International Journal of Computer Networks and Wireless Communications*, 6(3), pp. 1–6.
40. Tiwari, P. and Kumari Kushwaha, R. (2017) 'A Review on Trust Management Approaches in Wireless Sensor Networks', 3(5), pp. 609–614.
41. Uikey, C. and Bhilare, D. . (2017) 'TrustRBAC : Trust Role Based Access Control Model in Multi-Domain Cloud Environments', *IEEE, International Conference of Information, Communication, Instrumentation and Control (ICICIC*.
42. Uplap, P. and Sharma, P. (2014) 'Review of Heterogeneous / Homogeneous Wireless Sensor Networks and Intrusion Detection System Techniques', *Association of Computer Electronics and Electrical Engineers*, 05(02), pp. 501–507.
43. Wu, C.-H. and Chung, Y.-C. (2007) 'Heterogeneous Wireless Sensor Network Deployment and Topology Control Based on Irregular Sensor Model', in Cérin, C. and Li, K.-C. (eds) *Advances in Grid and Pervasive Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 78–88.
44. Zhang, J. *et al.* (2010) 'A Dynamic Trust Establishment and Management Framework for Wireless Sensor Networks', in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. doi: 10.1109/EUC.2010.80.
45. Zhang, J. *et al.* (2010) 'A trust management architecture for hierarchical wireless sensor networks', *Proceedings - Conference on Local Computer Networks, LCN*. doi: 10.1109/LCN.2010.5735718.
46. Zhou, L., Varadharajan, V. and Hitchens, M. (2015) 'Trust Enhanced Cryptographic Role-Based Access Control for Secure Cloud Data Storage', *IEEE Transactions on Information Forensics and Security*. doi: 10.1109/TIFS.2015.2455952.
47. Zicari, P. *et al.* (2017) 'Controversy-aware hybrid trust inference in online social networks', *Proceedings - 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 11th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Conference on Embedded*

Software and Systems, pp. 610–617. doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.291.