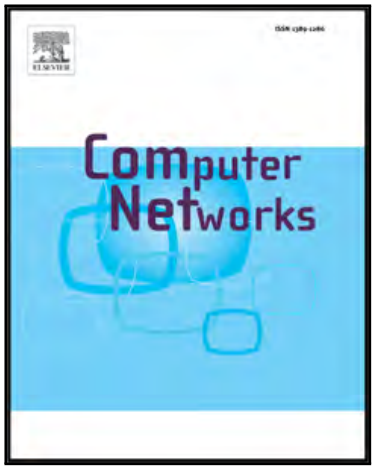
Journal Pre-proof



SUTSEC: SDN Utilized Trust based Secure Clustering in IoT

Kübra Kalkan

PII: S1389-1286(19)31078-3
DOI: https://doi.org/10.1016/j.comnet.2020.107328
Reference: COMPNW 107328

To appear in: *Computer Networks*

Received date: 26 August 2019
Revised date: 29 April 2020
Accepted date: 19 May 2020

Please cite this article as: Kübra Kalkan, SUTSEC: SDN Utilized Trust based Secure Clustering in IoT, *Computer Networks* (2020), doi: https://doi.org/10.1016/j.comnet.2020.107328

SUTSEC: SDN Utilized Trust based Secure Clustering in IoT

Kübra KALKAN

Department of Computer Science, Engineering Faculty, Özyeğin University, Istanbul, Turkey,
ORCID ID: https://orcid.org/0000-0003-1918-8587

Abstract

Internet of Things (IoT) technology consists of huge number of heterogeneous devices that create enormous amount of data. Providing a robust communication for billions of devices is one of the most significant challenges for IoT environment. Thus, cluster based communication is preferable as it promotes scalability. We propose an SDN Utilized Secure Clustering mechanism (SUTSEC) that provides benign cluster heads for the groups by considering mobility, priority, power and trust. SDN's inherited characteristics are leveraged for providing a dynamic secure selection. Additionally, secure key distribution is also considered in trusted clustering. During these security operations we considered several issues related to QoS and QoE such as energy efficiency, reliable communication, lower latency and user preferences awareness. We performed simulations of our proposal in order to show the percentage of compromised cluster heads. Our results suggest that despite half of the nodes are captured in the network, 70% of cluster heads are benign nodes in our model. This means that SUTSEC performs well in preventing the election of compromised nodes as cluster heads. Additionally, we provide analysis for compromised links and connectivity of nodes in order to show the performance of secure communication between clustered nodes.

Keywords: Internet of things, software defined networking, clustering, trust

1. Introduction

Wireless Sensor Networks (WSN) are composed of numerous interconnected micro devices which have sensing capabilities that monitor the envi-

ronment and gather information about the current situation [1]. It has a wide range of application areas that can be effectively deployed in human inaccessible areas such as military area monitoring, underwater and underground applications. However, the large number of sensing micro devices which compose the WSNs are equipped with limited energy resources. Thus, in order to prolong the lifetime of the network, clustering techniques are utilized to reduce the energy consumption in communication. In a clustered environment, a sensor node communicates with a local cluster head instead of a far base station. The main duties of a cluster head is to generate a transmission schedule, to gather data and to transmit it to a base station. Transmission scheduling prolongs the lifetime since the sensors does not have to be awake all the time. Additionally, the scheduling system reduces retransmissions since there will be less collisions.

The Internet of Things (IoT) concept utilizes WSN but it needs additional properties that provides high diversity, usability and mobility. IoT infers the connection of billions of devices at any place at any time [2]. This technology can be utilized in any type of application areas such as smart energy control, smart grid, home automation, industrial automation, health care, elderly care and smart cities [3; 4]. All these application areas require a wide range of distinct communication technologies and also will result in a large amount of data [5; 6; 7]. These facts are accompanied by some challenges.

In contrast to traditional devices with adequate computing, processing, and storage resources, IoT devices such as sensors and mobile devices are resource-constrained. The design of IoT application therefore should consider the different resource capabilities of heterogeneous devices that are part of the IoT environment. The reason that scalability poses a major challenge in IoT is to provide a communication environment for billions of devices. Additionally, we need to store and process large amount of data generated by various devices and systems. Besides, IoT technology needs to consider mobile devices such as cell phones, cars, smart watches etc. Also, one of the most important objective of IoT is to improve personal lives and personal experience [1]. This requires user oriented and context aware design. For instance, the accident detection systems with video supports on smart roads require shorter network delays and higher network bandwidth in a peak hour scenario. Thus, scalability, heterogeneity, context awareness and mobility make IoT technology more challenging than other traditional network systems [8].

Additionally QoS and QoE needs to be considered during migration of

IoT systems to 5G platforms since 5G platforms necessitates higher QoE, QoS and security. QoS refers to the longer network lifetime, more coverage, lower latency and more reliability, whereas QoE refers to the system that is sufficiently intelligent to adapt to the user preferences. In order to provide such a faster, reliable and smart system, secure clustering techniques become more essential since they provide energy efficiency, distributed processing of data and management hierarchy of mobile devices. Thus, it is important to have secure clustering techniques that considers QoS and QoE during their operations.

In theory, clustering the nodes significantly increased the life cycle of IoT devices [9]. However, sensor nodes can be deployed in an hostile environment that the nodes can be captured easily (ex: smart city, smart environmental systems). Once a cluster head is captured, adversary can reach and direct all the communication of this cluster [10]. Thus it is essential that the cluster head is not compromised.

Additionally, since the environment is hostile, it is important to have a secure communication between IoT nodes. As far as IoT consists of heterogeneous devices, it should support security even for simple devices such as sensors. For resource-constrained IoT sensors, symmetric keys are preferable since asymmetric key applications require more computational power and memory. Thus, symmetric keys should be distributed in an appropriate way among IoT devices. However, distribution of symmetric keys is not a trivial problem that many researchers have studied in this area and proposed lots of schemes [11]. In distribution, there is a trade-off between memory and resiliency. If only one pairwise key is used in the whole network, it is obvious that attacker can reach all the communications since it achieved the only key in the whole network. On the other hand, if different pairwise keys are generated for each pair, it is resilient to capture attacks, but this time it consumes huge amount of memory since each sensor needs to hold different keys for each node communication. Additionally, for IoT environment mobility and heterogeneity should be considered during key distribution. But still, symmetric encryption is inherently susceptible to eavesdropping and cryptographic techniques as it does not offer sufficient protection to the network in case of compromised nodes since they are already part of the network with necessary cryptographic materials [12]. If symmetric encryption is utilized and cluster head is compromised in a network, it will reveal lots of symmetric keys. Thus,it is important to have a trust mechanism that prevents to choose the compromised nodes as cluster heads.

In the literature, [12] proposes a trust based clustering mechanism for WSN, that decreases the likelihood of malicious or compromised nodes from becoming cluster heads. However, their work did not consider IoT devices' heterogeneous environment. They do not have any concerns about mobility, key distribution or priority that are related to QoS requirements of IoT. In this work, we focus on the problem of cluster head election for securely connected nodes in an IoT environment at a 5G platform. Our work inspired from [12] and reposition this work according to the current state of the art. We propose to utilize Software Defined Networking (SDN) technology for providing a secure clustering for IoT environment. SDN technology defines a new design and management approach for networking [13; 14; 15]. The main characteristic of this paradigm is the separation of the control and data planes. The SDN controller provides the decision whereas the switches handle data forwarding. Since decision algorithms do not run on network devices, simpler network devices can be used rather than complex routers. Moreover, in traditional networks, each router has its own security, link failure, and forwarding mechanisms. If any of these mechanisms need to be updated, each network device should be managed individually. However, one can manage all these issues centrally within the SDN architecture. There are several works in the literature that utilizes SDN for security solutions [16; 17; 18; 19]. We leveraged SDN to solve secure clustering problem since it provides more dynamic and agile [20; 21; 22; 23; 24] network environment. Its central management property provides a general view of the network. Then it can choose the most appropriate cluster heads at the current view of the network. Additionally, since lots of mechanisms has to be handled simultaneously during the cluster head selection such as mobility, priority (special treatment for the devices that transfers critical data and needs lower latency), trust calculation and blacklisting, all network devices should be informed immediately. SDN helps to change the packet routing according to the clustering. Also, these heavy duty algorithms are handled in the controller and the network devices will be relived. Moreover, SDN is one of the core technologies in 5G, thus it will be an advantage for technology migration through 5G. Our contributions are as follows:

- SUTSEC provides not only a clustering but also a trusted cluster head election property that avoids choosing the distrusted nodes.
- To the best of our knowledge, it is the first work that considers key distribution, trust and mobility together during clustering in IoT.

- To the best of our knowledge, this is the first work that considers priority during trusted clustering.
- While providing security, SUTSEC considers QoS parameters such as lower network latency, more reliability and longer network lifetime. Our work promotes these features by considering power, priority and mobility during trusted cluster head election. Also QoE property which is about user preferences can be provided by our work since it can adapt to the dynamic changes in preferences via SDN usage.

The structure of this paper is as follows. Section 2 discusses related work whereas Section 3 describes our system and threat model. Section 4 presents an overview of our mechanism including motivation and design details. Section 5 describes our simulation details and demonstrates performance evaluation. Then, Section 6 provides discussion and finally, Section 7 concludes the paper.

2. Related Work

There are several clustering methods in the literature. One of early works is presented in [25]. This work proposed a centralized algorithm in which all nodes send their current location and energy level information to the the base station. Then, the base station determines the cluster heads according to their energy level. These cluster heads work for a period of time and then new cluster heads are elected according to their residual energy. The more energy level infers more probability of electing as a cluster head for the next round. This election criteria prolongs the lifetime of the network. Since this method needs communication between the nodes and the base station at each round for election, it has huge amount of extra energy consumption. Additionally, since this method is for wireless sensor networks (WSN) it does not consider mobility or security issues which are inevitable concerns.

In [26], they provide security for cluster-based routing. Their solution utilizes Random Pairwise Keys(RPK)[27] scheme for key distribution. In this scheme, each node has several number of keys and the neighbours can communicate if they have common keys. Thus, neighbors communication is based on a probability of having common keys. This scheme is for flat networks, but they adapted it for hierarchical network and proposed RLEACH [26]. Their results suggest that RLEACH improves connectivity and reduce

memory overhead. However, they did not consider mobility and heterogeneity since their method is viable for WSN.

In [12], they have security concerns in cluster head election. They proposed a trust based clustering (TBC) framework that elects trustworthy nodes as cluster heads. They suppose that each node has a watchdog mechanism that allows it to monitor network events of neighbor nodes. Cluster head election process is revoked when the battery power level of the cluster head falls below a threshold. Each node sends their votes to the cluster head. Each node votes for his neighbors by considering the trust levels. Then, the cluster head decides the winner based on the majority. The second node is also selected as the vice president. Then the winner and the second node is exposed to a challenge-response protocol. If they passes, they are announced as the new cluster head and its assistant, otherwise these nodes are blacklisted. The trust value is also calculated according to the observable and measurable network events. The neighbors observe if the data packets are dropped or retransmitted. They also check if data contents or unique addresses are modified. They simulated their model and their results suggest that it has the ability to prevent compromised nodes from becoming cluster heads. However, this approach involves all nodes in the selection process, increasing the communication and computational overhead [28]. The performance is also effected adversely as communication packet sizes increase in large scale WSNs. Additionally, they do not consider power during the cluster election process. Also, they do not have any concerns about mobility, key distribution or priority that are related to QoS and 5G requirements in IoT environment. Also, their trust calculation mechanism depends on the assumption of each node has a watch dog mechanism that utilizes to monitor network events of other nodes in its rage. However, this could be not affordable for a real IoT environment with constrained requirements. We compare our proposal with this mechanism TBS, since we inspired and improved this work.

Another trust based model is presented in [10]. They utilized a trust model based on ant colony systems. Their results show that their work has high accuracy in preventing compromised nodes from being a cluster head. However, these trust based models does not consider mobility and heterogeneity since they are for WSNs. Additionally, they did not take into account secure communication during their simulations.

In the literature there exist several works that utilizes SDN for clustering in IoT. Al-Janabi et al. proposes a work in [29] which proposes a new clus-

tering protocol based on SDN. It utilizes SDN controller to divide the area into virtual zones to balance the number of cluster heads in each zone considering the node density. There is also another SDN based clustering method in [30; 31]. They introduced the concept of SDN cluster heads (SDNCHs) for multiple SDN domains. A domain is composed of an SDNCH, gateway and sensor nodes. An SDNCH acts as the domain's coordinator, whereas the gateway is the bridge node between the sensor nodes and the SDNCH. Routing functions and security rules are distributed among the SDNCHs. The authors proposed a routing protocol for the distributed clusters and built a test-bed to evaluate the protocol. They also provided another paper to show the test bed in detail [32]. Another work in the literature is provided in [33] which uses Ubiflow framework to combine IoT and SDN. This framework provides an effective flow control and mobility management in different networks using SDN controllers. IoT network is divided into clusters and every division has a local SDN controller. SDN controllers collaborate to provide scalability in different geographic locations [34; 35].

3. System and Threat Model

In this section we give details about our system model and we explain which type of adversary we considered in our protocols.

3.1. System Model

In our model, we assume that there exist a group of connected IoT devices belonging to the same owner. It can be a company's factory, a campus of a university, a building of a company, a smart house or a research center in a forest that deals with environmental issues and distributed sensors over the forest to gather data. For our model, we assume that there is an SDN network which consists of a controller and several switches. Each switch is SDN-enabled and it is called SDN-IoT gateway (SDN-IoT-GW). The controller has several modules which have different roles as illustrated in Figure 1. These modules are defined as virtual functions. It consists of registration, key distribution, trust, priority, path, mobility and CH election modules. The modules can have interactions with each other during the operations. Also, cluster head election steps are explained in Figure 2. Each SDN-IoT-GW operates these steps to generate the clusters and determine the cluster heads. The details of modules and election process steps are explained in Section 4.

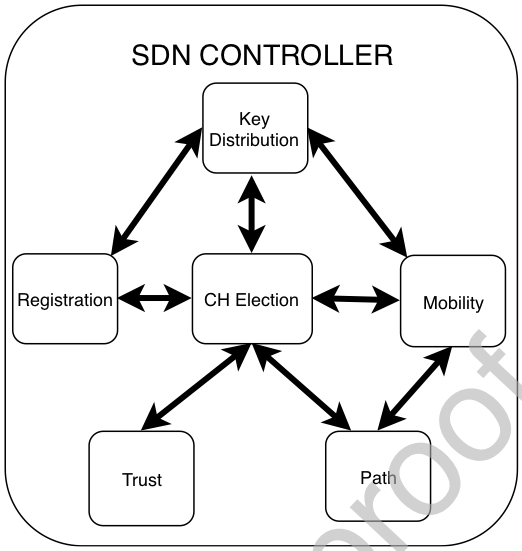


Figure 1: Modules of SUTSEC controller

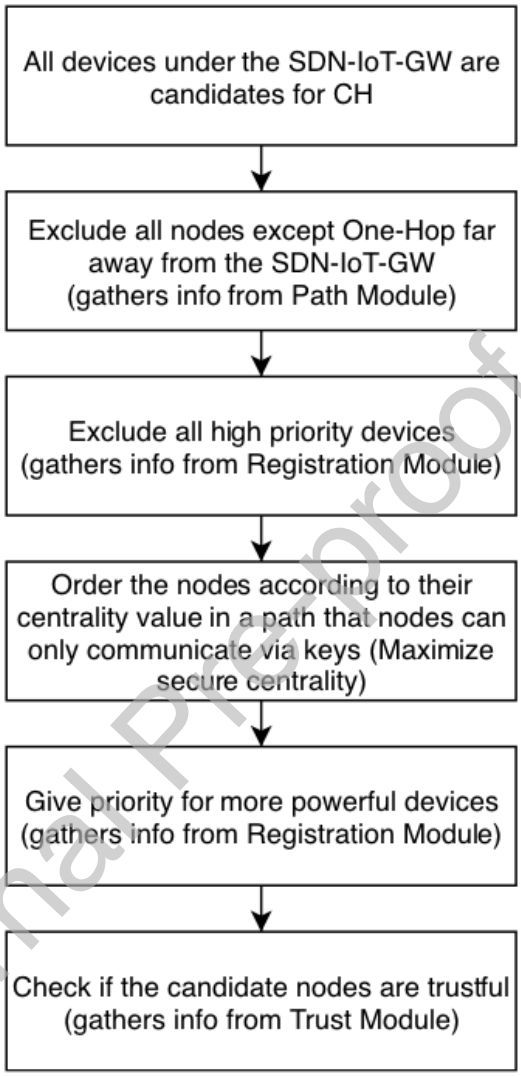


Figure 2: Steps of CH Election executed for each SDN-IoT-GW

There is also a hierarchical topology that the devices are grouped into clusters and each cluster has its own cluster head (CH). Since our architecture SUTSEC proposes to utilize SDN, each cluster head is also under an OpenFlow switch. This hierarchical topology is illustrated in Figure 3.

In this figure, each SDN-IoT-GW has several clusters and each cluster has several number of heterogeneous devices. Different shapes infer devices

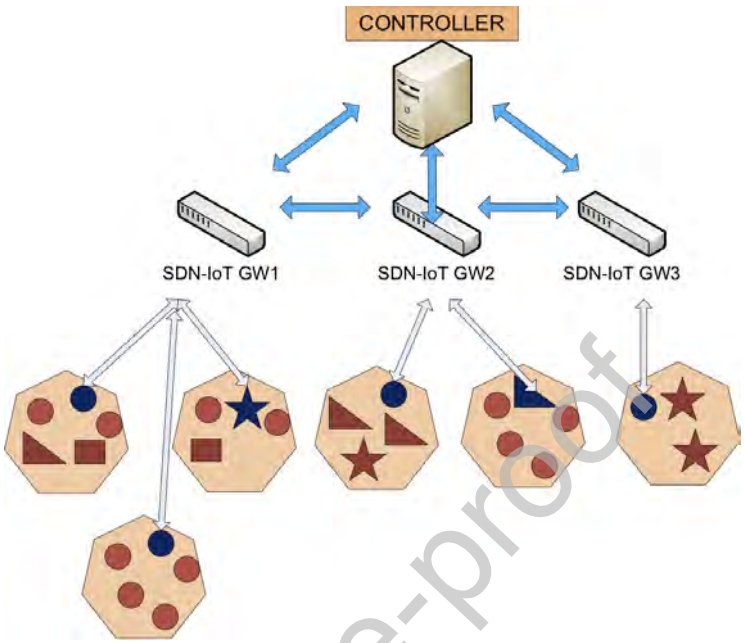


Figure 3: Hierarchical Topology of SUTSEC

with different properties such as different computing power, different abilities, different ranges and different application areas. These can be anything like cell phones, sensors, laptops or any smart devices. Each cluster has its cluster head which is depicted in darker color. These cluster heads communicate with both the gateways and the devices in their own clusters.

3.2. Threat Model

For our analysis we assume that there is an adversary who wants to obtain all the communication in the network. Since all the communications between nodes are encrypted via keys, adversary tries to get these keys by capturing the nodes. He compromises nodes in a fixed rate. When he compromises a node, he captures all the keys and active links of this node. We suppose that the attacker is global that it can reach all parts of the network. Thus, if he compromises a key, then he can capture all the links that this key is utilized in the network.

4. SDN Utilized Trust based Secure Clustering

SDN Utilized Trust based Secure Clustering (SUTSEC) is a novel central mechanism that aims to provide a scalable and secure communication environment for mobile and heterogeneous IoT devices. SUTSEC is a central mechanism due to the heterogeneous structure of IoT environment. Decentralized approach is more challenging due to the hardware limitations of cheap and transmit only devices [1; 36]. Also, it is an adaptive type of CH election scheme that considers several parameters for CH election. Our main aim is to provide secure clustering with choosing the parameters that considers QoE and QoS concerns. In order to prolong the life time of the network our model proposes an energy aware CH election. Additionally, we consider the coverage sensitive longevity defined in [37], which considers not only the lifetime but also the network coverage. If loosing network coverage, the system will lose part of the sensing data, thus we provide connectivity analysis which considers the secure coverage of the network. For transmission reliability, our model provides trust consideration during CH election which is one of the main objectives of our work. Since CHs are the main communication bridges, if they are compromised adversary can drop the packages at will and the transmission will not be reliable. Another QoS metric, low network latency is an essential criterion in some IoT applications such as smart cities as once a car is detected the system needs to turn on the lights. Similarly, there can be critical data that needs faster transmission in emergency situation. We leveraged SDN to provide this property with the help of its central and dynamic management features. The devices which can transmit critical data and needs lower latency will have a priority and will be treated specially. Also, numerous number of switches usage balances the cluster size and reduces the network latency. Via SDN utilization SUTSEC can choose the most appropriate cluster heads at the current view of all the network. During the cluster head selection mobility, securely communicated neighbors of nodes, trust and blacklisting issues needs to be considered. Thus, the central view and immediate briefing to all network devices are essential. Also, as SDN technology provides routing control over the network from the controller it will manage to handle the dynamic path changes in mobility situation. It will send the routing rules to the switches simultaneously via OpenFlow. Additionally, since the controller will cope with all these heavy-duty jobs the network devices will be relived and can do their routing jobs more properly.

The common objectives for QoS in a system include longer network life-

time, more coverage, lower latency and more reliability [1]. While we provide secure clustering, our operations consider the following QoS and QoE concerns with the corresponding properties:

- longer network lifetime: energy aware CH election operation.
- more coverage: coverage sensitive CH election operation.
- lower latency: central and dynamic management with SDN utilization.
- more reliability: CH election with considering trust.
- adaptability to user preferences: priority parameter in CH election and SDN utilization provide this property.

The election process of SUTSEC considers the following properties of the nodes:

- Secure centrality between SDN-IoT gateway and the member of cluster nodes: Since the main objective of clustering is to decrease the communication burden, the nodes who are able to communicate with more nodes securely are the candidates for cluster heads. Thus, centrality is calculated according to the secure shortest paths from the cluster members to the SDN-IoT-GW. The candidate who is in the secure shortest paths of more cluster members are chosen as cluster head. The cluster head should maximize the number of shortest paths from all cluster members to the SDN-IoT-GW that pass through that node securely. This can be achieved after secure communication path is provided between nodes. A secure communication between two nodes can also be provided if they have a common key. Key management and secure communication path construction details are provided in following subsections.
- One-hop far from SDN-IoT-GW: The cluster head must be chosen from the nodes who can reach to the SDN-IoT-GW in one hop. This will decrease the communication burden. If larger hops are allowed then it will consume more energy for each transmission.

  In Figure 4, one hop nodes that are one hop far from the SDN-IoT-GW are illustrated as the candidate nodes $CAN1$, $CAN2$ and $CAN3$. The secure communication paths between nodes are illustrated with arrows.

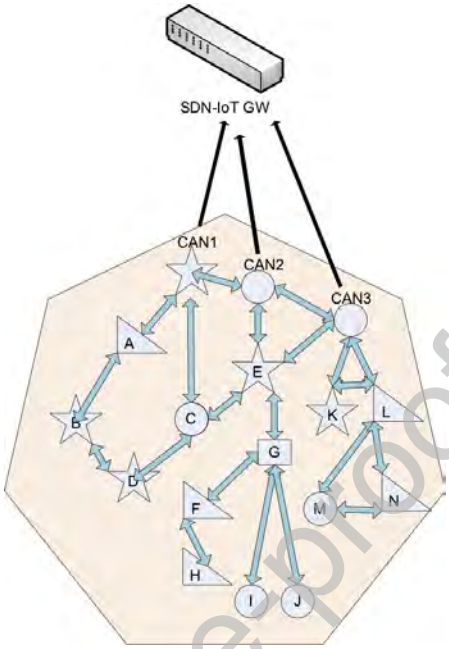


Figure 4: Secure centrality between SDN-IoT gateway and the member of cluster nodes

Their secure centrality can be calculated as the number of nodes whose shortest paths to the SDN-IoT-GW pass through the candidate node. The arrows show the shortest paths for each node to SDN-IoT-GW. For instance secure centrality for $CAN1$ is 4 since A, B, C and D can reach SDN-IoT-GW via $CAN1$ in the shortest way. Similarly, secure centrality of $CAN2$ can be illustrated as $CAN2_{cen} = 6$ since E, F, G, H, I and J can reach through $CAN2$. $CAN3_{cen} = 5$ due to the nodes E,K, L, M, and N. Then the larger secure centrality value for a candidate cluster head will be preferable ($CAN1$ in this example), since it will decrease communication burden during the transmissions.

- Power: Since the IoT environment is heterogeneous, devices have different power capabilities. Power is a significant issue as all communication of the cluster nodes will be provided through the cluster head. It will consume huge amount of energy. Thus in order to prolong the life time of the cluster, it is essential to choose more powerful devices which are connected to the power grid as cluster heads.

- Trust: Trust is the indicator that shows if the node is compromised or not. This objective prevents adversarial nodes from becoming cluster heads. Trust calculation is explained in Section 4.5.

- Priority: The device type is essential to categorize its priority. If the device can hold a critical or low latency required data then these type of devices are marked as prior in the registration phase and they are treated specially during cluster head election. Priority treatment are explained in detailed in Section 4.4.

In SUTSEC the controller has six modules that are responsible for different duties. The main structure of SUTSEC is also handled by these modules as illustrated in Figure 1. The working principles of these modules are explained in the following subsections.

4.1. Registration Module

In order to join the network each device needs to register the SDN network. This is controlled by the module in controller which is working as a registration desk. It records the properties of the device in terms of its power and priority. They are grouped as powerful and weak devices in terms of power. Also, if the device will convey a critical data it is maerked as high priority device. Then, Registration module contacts with the Key-Distribution module and gets the necessary keys (explained in the next subsection) and an identifier number. Afterwards Registration module sends these information to the new registered device.

4.2. Key Distribution Module

This module is dealing with key operations to provide secure communication between the devices. It provides symmetric keys for the weak devices whereas it can generate asymmetric keys for powerful devices. Since IoT environment is an heterogeneous environment, we have utilized random key pre-distribution scheme in order to provide secure communication even with weak devices. We suppose that weak devices are preloaded with symmetric keys from a key pool before they attend to the network (similar to the WSN preloaded key distribution mechanisms [11]). These keys are from a key pool which is in the Key Distribution module. On the other hand, powerful devices can behave more dynamically and they can request a required key from the controller via the gateway to communicate with the weak devices.

Since the powerful devices and SDN-IoT-GWs have their assymettric keys, the communication between a powerful device and SDN-IoT-GW and also communication between two powerful devices are provided by the keys which are generated by Diffie-Hellman key exchange protocol.

Let's suppose that there is a key pool KP which has P number of keys and each key has a unique identity number ID_x. The KP is on the Key Distribution module and it distributes these keys randomly on the weak devices before they are deployed. Each node with id i is illustrated as n_i and it has a key ring kr_i which is composed of r number of keys. Each key of the key ring kr_i is shown as k_i^m. ID of each key k_i^m are shown as $ID\_k_i^m$. If node n_a wants to communicate with node n_b, if the key $k_a^m \in kr_a$ and the key $k_b^n \in kr_b$, then n_a and n_b try to find a condition that ensures $ID\_k_a^m = ID\_k_b^n$. Then a secure communication path is provided between n_a and n_b. This procedure is applied if both of the nodes are weak devices. If these weak devices do not any have common keys, then they will communicate via SDN-IoT-GW. All the terms that are utilized in our system model is shown in Table 1.

On the other hand, if one of the nodes are powerful, then the weak device sends an ID of one of the keys from its key ring kr and the powerful device request this key from the Key Distribution module via the SDN-IoT-GW. Then, this key is provided in a secure way after a challenge response procedure is applied on the powerful device. (We assume that secure communication is already provided between the powerful device and the SDN-IoT-GW).

Namely, when a node changes its place and enters to the range of an SDN-IoT-GW, it requests a key that exists in this node's key ring and they communicate securely. Additionally, since we suppose that the nodes can be mobile, they also need to do all these operations when they change places.

4.3. Path Module

The main duty of this module is to hold the map of the network. It holds the shortest path information from each node to the SDN-IoT-GW and holds a candidate CH list CL for each SDN-IoT-GW, which holds the devices that are reachable in one hop from the gateway. It is important to note that the mentioned path is constructed over the secure communications in which it is not enough to be in the same range to communicate, but also the nodes need to have common keys to have secure transmission. This information is gathered via SDN-IoT-GWs.

Table 1: System symbols

Term	Explanation
KP	Key Pool
P	Total number of keys in KP
ID_x	Identity number of key x in KP
n_i	Node with id i
kr_i	Key ring of node i
r	Total number of keys in a key ring
k_i^m	mth key of the key ring kr_i
$ID\_k_i^m$	ID of mth key of the key ring kr_i
CAN	Candidate cluster head
CL	Candidate list
CAN_{cen}	Centrality between SDN-IoT gateway and a candidate cluster head
λ	Power threshold
$PCCH$	List of powerful candidate cluster heads that is sorted in descending order according to CAN_{cen} values of the cluster head candidates
$WCCH$	List of weak candidate cluster heads that is sorted in descending order according to CAN_{cen} values of the cluster head candidates
RL	List of nodes that are not included in a cluster yet
ρ	Trust threshold
CH	Cluster Head
PL	Prior Device List

4.4. CH Election Module

This module is the core module which gathers information from other modules and runs the following algorithm for each SDN-IoT-GW.

- The list of candidate nodes CL ,which are reachable in one hop from the gateway, is requested from the Path Module.
- Priority device list PDL is gathered from the Registration module.
- Remove prior devices from the candidate list CL.
- The CAN_{cen} value is calculated for all the candidate nodes CANs in CL.
- The power information is gathered from the Registration module. The one hop nodes which are powerful are sorted in descending order according to their CAN_{cen} value in a powerful candidate cluster head list $PCCH$.
- Similarly, the one hop nodes which are weak are sorted in descending order according to their CAN_{cen} value in a weak candidate cluster head list $WCCH$.
- After $PCCH$ and $WCCH$ lists are ready, each candidate node is analyzed until the cluster heads are determined and all the nodes are included in a cluster. Initially, all the nodes are listed in a remaining list RL. When they are owned by a cluster head, they are extracted from this list. This algorithm is explained in the following pseudo code.

Ensure: $PCCH$ and $WCCH$ lists are ready. Loop through the nodes in $PCCH$ and then $WCCH$ in order until the remaining list RL is emptied.
  if $CAN_{tr} > \rho$ **then**
    $CH = CAN$
    Each node in range of CH is included as a member of this cluster.
    Member nodes are extracted from RL.
  else
    CAN is blacklisted.
  end if

Elements of $PCCH$ and $WCCH$ are analyzed respectively as a candidate cluster head until RL list is emptied. Trust value of the candidate node, CAN_{tr} is compared with a threshold ρ. If it is above the threshold, then it is determined as a CH. All the nodes in the range of this node is determined as its cluster members and these are extracted from RL. If CAN_{tr} is also below the threshold ρ, then it is reported to SDN-IoT-GW and to the Trust module of the controller. Then this module informs all the network that CAN is a malicious node and it is blacklisted. It is important that all the network is informed since the nodes can be mobile in an IoT environment.

After the CHs are determined then, the prior nodes in PDL will be distributed. The prior devices are removed from the candidate list at the beginning as it is not salutary to have these node as CHs since CHs needs to deal with extra communication and processing tasks which will consume more energy and can make it busy with these duties and overshadow the main job of the prior device. Thus, each device in PDL is treated specially by considering the following preferences:

- Choose the CH in its range which is more powerful.
- Put the device in a smaller cluster.
- Put as few prior devices in the same group as possible.

It is obvious that the prior device will prefer to be in a more plentiful environment. Thus it will choose to be under a more powerful cluster head and in a smaller cluster. Also it will be better not to share the same area with another prior device since their priorities can be in conflict.

If a node is in range of two CHs, then it will be a member of the candidate CH which is former in the $PCCH$ /$WCCH$. Lets say $PCCH$ list consists A, B, C, D and node k is range of A and B both. All the nodes are in RL so k is also in RL list $(RLa, b, c, d, e, f, g, h...k, l, m, n....)$. Our algorithm will start with A and it will process all the nodes in its range. Then A will add k to its members and remove k from RL. Thus, k will be As member who is the former device in $PCCH$ list.

4.5. Trust Module

Trust module is holding trust values for the devices and blacklist which consists of the nodes that are distrusted. Trust value of a candidate node is determined according to the other nodes reputations about this node. In our

mechanism, the trust value is determined according to the similar parameters utilized in [12]. Each node grades other nodes according to average packet drop rate, average data and unique address modification rate. They send their differentiated grades to their cluster head periodically. The CHs under the SDN-IoT-GWs send the trust value tables of their members to the Trust module periodically. If a CH is captured, it can change the trust values of its members while it is sending it to SDN-IoT-GW. For this reason, trust module changes members trust values not only looking at the newly coming report but also considering historical data. According to these values if a node's state will change, then trust module will send a message to the nodes who are the members of this cluster. Let's suppose CH_m is captured and it changes its members' trust values and due to incorrect scores that it send to trust module, a node N is about to become an untrusted node (its trust value will be less than the trust threshold ρ). Then, trust module will send a request message in an encrypted way for all the members of CH_m. Each message is encrypted with a key of the corresponding member (if it is a weak device it will be a key from its key ring, otherwise it will start a Diffie Hellman key exchange procedure) and will send through CH_m. CH_m cannot read or alter the message as it is encrypted, also CH_m cannot drop the messages since trust module will start to doubt about CH_m as it will be suspicious not to get any answer for this request from the members of CH_m. At the end, if these members send similar values to the CH_m's values about N, then N will become an untrusted node. Otherwise, CH_m will be punished and its trust value will be decreased. So ultimately, via the trust module a central reputation map of the system is constituted in the controller.

4.6. Mobility Module

Since IoT technology provides heterogeneous environment that can consist mobile devices, mobility should also be considered. Mobility will result in changes in clusters, cluster heads and secure communication paths. Thus, each case has different results and needs to be examined in details. Cluster head can migrate and leave the cluster or a node can migrate from one cluster to another. Also, periodically migrated nodes should be reconsidered since majority can break off and a new cluster can be generated. These cases are handled by Mobility module and explained in the following subsections.

- Cluster Head Migration: If the cluster head is a mobile device it can break off the current cluster. In this case, the nearest cluster head is

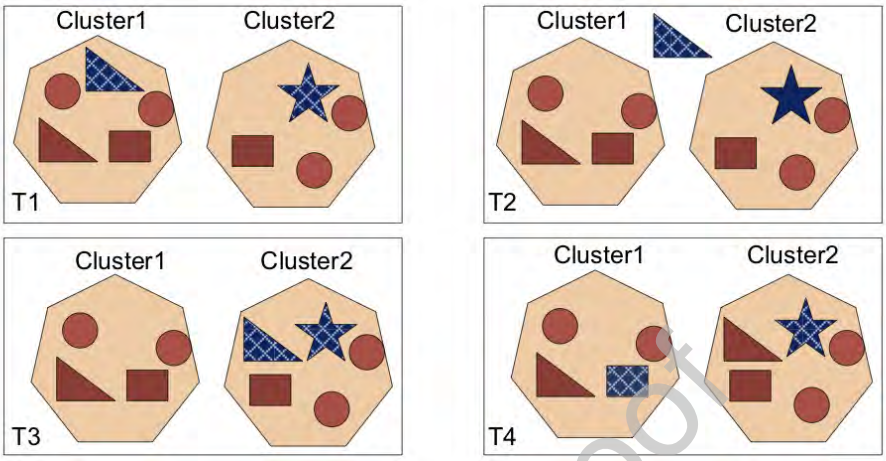


Figure 5: Cluster Head Migration

detected via the help of the SDN-IoT-GW and it will bind as a cluster member to the new cluster. Additionally, the previous cluster should be revoked and a new election process should be started. This is illustrated in Figure 5. Cluster heads are illustrated with dashed shapes. T1, T2, T3 and T4 shows the movements of the cluster head of Cluster 1. The triangle node is the cluster head of Cluster 1 and starts to move. In T3, it comes to a place that is in the coverage area of the cluster head of Cluster 2 and it can create a secure path via a common key with the cluster head of Cluster 2. Then it becomes a member of Cluster 2. Also, a rectangle node in Cluster 1 is chosen as the new cluster head since it has the highest secure centrality value.

- Node Migration: Cluster member can also migrate from a cluster to another. Then, it will try to find a common key to connect to the new cluster head. If they do not have any common keys and the new cluster head is a powerful device then powerful device will request a key (one of the keys from the key ring of the migrated node) from the controller. And then, the new cluster head will add this node as a cluster member.

- Reclustering: If the nodes are not just migrated but also move away too far and cannot connect to a cluster then these nodes are ruptured. After awhile number of nodes that are ruptured will be increased and they may create a new cluster. Then reclustering process should be

started and they will choose their cluster heads. If the raptured node is not become a member of a cluster, then it will be independent and decrease the connectivity value of the network.

5. Performance Evaluation

We perform experiments via simulations for performance evaluation of SUTSEC and an existing trust based clustering method TBC in [12]. We compared our work with TBC as we inspired and improved this work. Also, it is the most similar work that is also based on trust calculation during clustering. TBC does not consider mobility in their simulations but we also applied their model for mobile nodes.

5.1. Performance Metrics

In order to compare the performance of TBC and SUTSEC, the following metrics are considered:

- Compromised Cluster Head Ratio: Since cluster heads have more links, it is essential to have less captured cluster head nodes. This metric shows the ratio of captured cluster heads over all cluster heads.

- Connectivity: This metric shows the ratio of the connected nodes to the network. Since there is common key share issue and mobility, after awhile some of the nodes may left out of the coverage area of any cluster heads and they become independent nodes. The connectivity metric shows the ratio of the connected nodes excluding the independent nodes.

$$Con = 1 - (IndependentNodes/Allnodes) \tag{1}$$

Since TBC does not consider secure communication between nodes, they do not include key distribution processes. However, in our model SUTSEC we also need to consider two additional metrics:

- Compromised Links: When a node is captured, its all communication links are also captured. Thus, this metric shows the ratio of the links that are captured by the adversary over all links.

- Additionally Compromised Links: When a node is captured, its all keys are also captured. Thus, if the same keys are also utilized in other places of the network, then the adversary can easily reveal their communication. For this reason, additionally compromised links is an essential metric that shows the ratio of the compromised additional links over all links.

5.2. Simulation Details

In order to evaluate the works, we utilize Mininet-Wifi [38] as SDN-IoT network simulator. It enables to create a realistic network topology and it is convenient for mobile SDN-IoT environment. It is integrated with Ryu controller. The test environment is Ubuntu 14.04.

There are 200 IoT nodes and nine switches (SDN-IoT gateways) that are managed by a Ryu controller. The simulation area is $200mx200m$. The range of a switch is 30 metres. The heterogeneous nodes are deployed randomly. Half of the nodes are powerful whereas others are weak. The key pool size is 10000 and key ring size of a node is 30. In our simulation, there is an active external adversary that captures nodes in periods as explained in Section 3.2. Also nodes are mobile that moves according to random-walk mobility model.

5.3. Results

Figure 6 shows the compromised Cluster Head Ratio for SUTSEC and TBC during a hundred periods. We suppose that capture rate for the adversary is one node per period. In this case, at the end of a hundred periods, half of the nodes are captured. According to the results, the percentage of compromised cluster heads in SUTSEC is about half of TBC. Also, results suggest that when half of the nodes are captured, 30% are malicious which indicates that 70% of cluster heads are benign in SUTSEC. This result is highly favorable in such a case that most of the network is captured.

In Figure 7, the curves show the connectivity of the network. Connectivity is highly related to the mobility and it is obvious that connectivity decreases since the number of independent nodes increases after a while as the nodes change positions. TBC does not consider key issues so it is enough to be in range to be connected in TBC. However, SUTSEC does not only consider coverage, but also key share between nodes. Thus, the results show that connectivity performance of SUTSEC is not as good as TBC. It is an expected result since, TBC did not consider common key share for connecting the nodes. SUTSEC's connectivity decreases faster since after mobility it is more

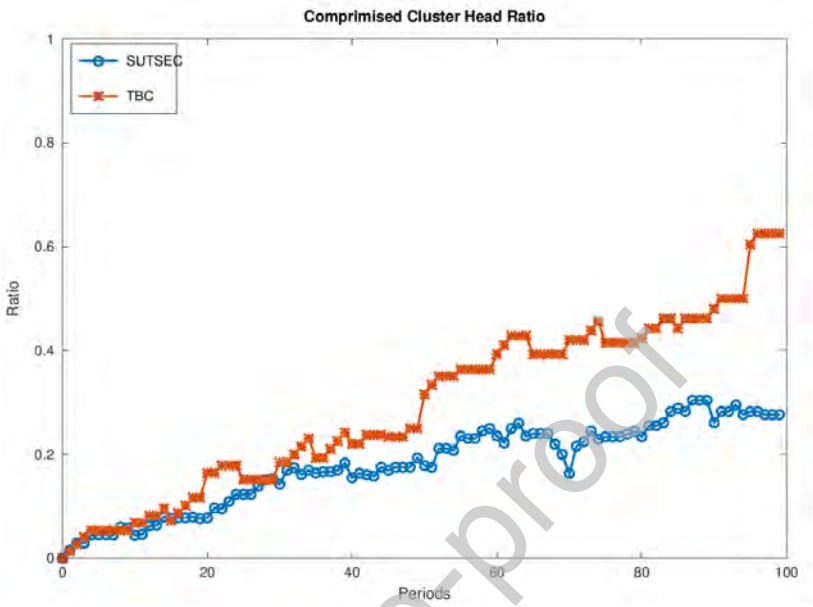


Figure 6: Compromised Cluster Heads

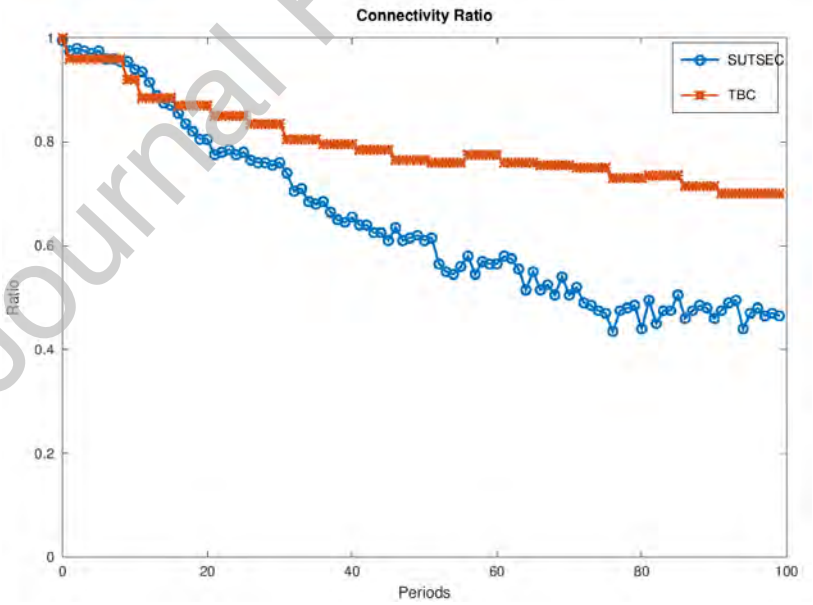


Figure 7: Connectivity

difficult to find common keys between nodes. Additionally, when the number of captured nodes increases, our model finds cluster heads more difficultly. Because our model goes over the trust issue with a fine-toothed comb and it prefers to make a group independent until it finds a trusted cluster head node. Thus, there is a trade off between a more secure network and connectivity.

Figure 8 shows the compromised links ratio for SUTSEC during different capture rates. The results are illustrated for capture ratio of the adversary for one node per period and two nodes per period. It is obvious that compromised links ratio increases when the number of captured nodes increases. Also Figure 9, shows the comparison of additionally compromised links and compromised links ratio. This figure shows the importance of additionally compromised links metric, since this show the actual effect of an adversary on the network. The results suggest that additionally compromised links are more than two fold of the captured links. Thus, the real effect of node compromise needs to be considered as double. The system managers can be more optimistic since just a few of nodes are captured. But it is obvious that the trouble is much more that it is expected. This problem can be handled if the key pool is larger and the same key is not utilized in different places of the network, but this time connectivity will be lower.

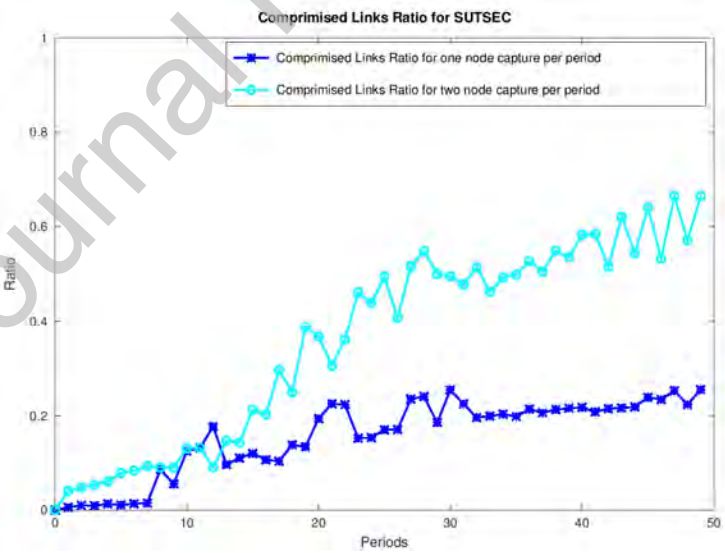


Figure 8: Compromised Links

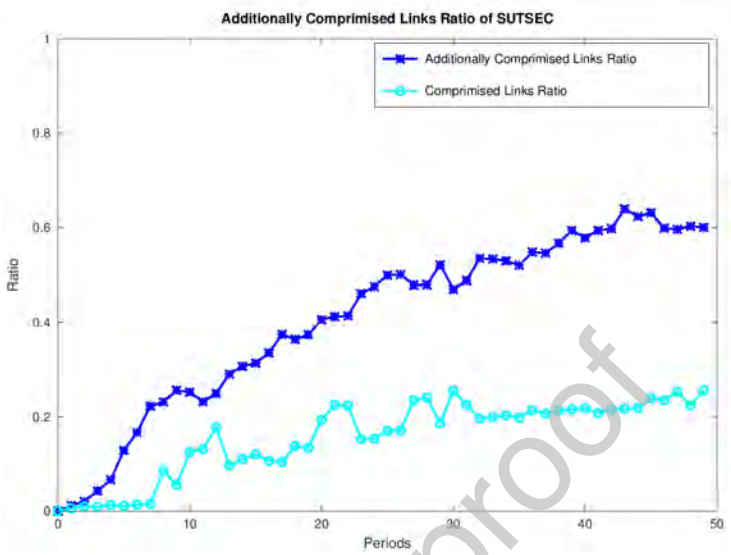


Figure 9: Additionally Compromised Links

6. Discussion

In our proposal, one of the the most important role is provided by the trust calculation mechanism. Each node calculates the trust value according to the network events. In order to improve accuracy, these calculations can be provided in auto regression style. A node will give its reputations according to not only the current but also previous observes.

In our simulations, we utilized random key predistribution scheme. However, our system can also utilize hierarchical key distribution schemes. For instance, Blom's key distribution scheme [39] can be appropriate. Additionally, our results suggest that additionally compromised links ratio is very high since same keys are utilized in different places of the network. Thus, zone based key distribution schemes [40] can also be utilized and analyzed if they are appropriate in case of mobility. More detailed key distribution analysis with different key distribution models can be considered as a future work.

In [1], the authors suggest that when clustering in IoT, from the perspective of reducing redundant data, devices with similar usage should be grouped together. However, we suggest that this argument is not valid in all conditions. Grouping the prior type devices (which holds critical data or

needs lower latency) together is not preferable since their priorities can be in conflict and it will be more difficult for a cluster head to serve for several low latency required devices. Thus, we suggest that this strategy depends on device types, applications and preferences. Since we utilized SDN in our network, this strategy can be adjusted easily in CH election module of the controller according to the preferences. This property is favorable in terms of QoE in 5G platforms. More detailed priority analysis with having varied number of priority devices distributed in different clusters is left as a future work.

7. Conclusion

In this work we propose a trust based secure clustering mechanism for IoT environment. Clustering for IoT networks needs to consider QoS and QoE which are including energy efficiency, reliable communication, lower latency and user preferences awareness. Thus, we inspire and improve an existing clustering work TBC [12]. SUTSEC considers power, trust, secure centrality, mobility, priority and heterogeneity during clustering the nodes. It also utilizes SDN which provides dynamism. It also considers secure communication among nodes by utilizing key distribution. These properties are provided by the controller that consists of registration, key distribution, path, mobility, trust and CH election modules. We compare our model with TBC according to compromised cluster head ratio and connectivity metrics. The results suggest that it gives highly favorable performance on trusted cluster head election. Since the priority of the mechanism is to protect the system, in the worst case it may choose to make the nodes independent instead of binding it to a compromised node. Besides, since SUTSEC provides secure communication, it needs to consider key share during communication. For this reason, it is expected to have lower connectivity as it needs to find a common key in order to be connected. Also, SUTSEC is analyzed in terms of compromised links and additionally compromised links. Our results suggest that additionally compromised links ratio needs to be considered since just compromised links results can mislead the experts.

As a future work, we plan to make experiments in a real testbed. This will make our results more realistic. To the best of our knowledge, in the literature, the trust parameters are not defined for SDN environment. It can be defined and more scrutinizing and improved trust calculations can be provided. Also, more detailed priority analysis can be provided with having

varied number of priority devices distributed in different clusters in a real testbed. Additionally, hierarchical key distribution schemes can be applied for SUTSEC.

References

References

[1] L. Xu, R. Collier, G. M. OHare, A Survey of Clustering Techniques in WSNs and Consideration of the Challenges of Applying such to 5G IoT Scenarios, IEEE Internet of Things Journal 4 (5) (2017) 1229–1249.

[2] G. Santucci, The Internet of Things: Between the Revolution of the Internet and the Metamorphosis of Objects, Vision and Challenges for Realising the Internet of Things (2010) 11–24.

[3] K. Kalkan, S. Zeadally, Securing Internet of Things with Software Defined Networking, IEEE Communications Magazine 56 (9) (2017) 186–192.

[4] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An Information Framework for Creating a Smart City through Internet of Things, IEEE Internet of Things journal 1 (2) (2014) 112–121.

[5] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, A. Kayssi, Identity-based Authentication Scheme for the Internet of Things, in: Computers and Communication (ISCC), 2016 IEEE Symposium on, IEEE, 2016, pp. 1109–1111.

[6] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, P. Siano, IoT-Based Smart Cities: a Survey, in: 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), IEEE, 2016, pp. 1–6.

[7] A. Whitmore, A. Agarwal, L. Da Xu, The Internet of ThingsA Survey of Topics and Trends, Information Systems Frontiers 17 (2) (2015) 261–274.

[8] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, S. Shieh, IoT Security: Ongoing Challenges and Research Opportunities,

in: Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on, IEEE, 2014, pp. 230–234.

[9] B. Shen, S.-Y. Zhang, Y.-P. Zhong, Cluster-based Routing Protocols for Wireless Sensor Networks., Ruan Jian Xue Bao(Journal of Software) 17 (7) (2006) 1588–1600.

[10] L. Yan, Y. Pan, J. Zhang, Trust Cluster Head Election Algorithm Based on Ant Colony Systems, in: Computational Science and Optimization (CSO), 2010 Third International Joint Conference on, Vol. 2, IEEE, 2010, pp. 419–422.

[11] S. A. Camtepe, B. Yener, Key Distribution Mechanisms for Wireless Sensor Networks: a Survey, Rensselaer Polytechnic Institute, Troy, New York, Technical Report (2005) 05–07.

[12] G. V. Crosby, N. Pissinou, J. Gadze, A Framework for Trust-Based Cluster Head Election in Wireless Sensor Networks, in: Dependability and Security in Sensor Networks and Systems, 2006. DSSNS 2006. Second IEEE Workshop on, IEEE, 2006, pp. 10–pp.

[13] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, M. Imran, Security in Software-Defined Networking: Threats and Countermeasures, Mobile Networks and Applications 21 (5) (2016) 764–776. doi:10.1007/s11036-016-0676-x.

[14] M. Dabbagh, B. Hamdaoui, M. Guizani, A. Rayes, Software-Defined Networking Security: Pros and Cons, IEEE Communications Magazine 53 (6) (2015) 73–79.

[15] S. Scott-Hayward, G. O'Callaghan, S. Sezer, SDN Security: A Survey, in: 2013 IEEE SDN For Future Networks and Services (SDN4FNS), IEEE, 2013, pp. 1–7.

[16] M. Nobakht, V. Sivaraman, R. Boreli, A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using Openflow, in: 2016 11th International conference on availability, reliability and security (ARES), IEEE, 2016, pp. 147–156.

[17] S. Chakrabarty, D. W. Engels, S. Thathapudi, Black SDN for the Internet of Things, in: 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, IEEE, 2015, pp. 190–198.

[18] P. Bull, R. Austin, E. Popov, M. Sharma, R. Watson, Flow Based Security for IoT Devices Using an SDN Gateway, in: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, 2016, pp. 157–163.

[19] K. Kalkan, L. Altay, G. Gür, F. Alagöz, JESS: Joint Entropy-Based DDoS Defense Scheme in SDN, IEEE Journal on Selected Areas in Communications 36 (10) (2018) 2358–2372.

[20] V. R. Tadinada, Software Defined Networking: Redefining the Future of Internet in IoT and Cloud Era, in: Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, IEEE, 2014, pp. 296–301.

[21] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, M. Conti, A Survey on the Security of Stateful SDN Data Planes, IEEE Communications Surveys & Tutorials 19 (3) (2017) 1701–1725.

[22] Q. Yan, F. R. Yu, Q. Gong, J. Li, Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges, IEEE Communications Surveys & Tutorials 18 (1) (2015) 602–622.

[23] R. Amin, M. Reisslein, N. Shah, Hybrid SDN Networks: A Survey of Existing Approaches, IEEE Communications Surveys & Tutorials 20 (4) (2018) 3259–3306.

[24] N. Bizanis, F. A. Kuipers, SDN and Virtualization Solutions for the Internet of Things: A Survey, IEEE Access 4 (2016) 5591–5606.

[25] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An Application-Specific Protocol Srchitecture for Wireless Microsensor Networks, IEEE Transactions on wireless communications 1 (4) (2002) 660–670.

[26] K. Zhang, C. Wang, C. Wang, A Secure Routing Protocol for Cluster-Based Wireless Sensor Networks Using Group Key Management, in: 2008 4th International Conference on Wireless Com-

munications, Networking and Mobile Computing, 2008, pp. 1–5. doi:10.1109/WiCom.2008.889.

[27] H. Chan, A. Perrig, D. Song, Random Key Predistribution Schemes for Sensor Networks, in: Security and Privacy, 2003. Proceedings. 2003 Symposium on, IEEE, 2003, pp. 197–213.

[28] G. Han, J. Jiang, L. Shu, J. Niu, H.-C. Chao, Management and Applications of Trust in Wireless Sensor Networks: A Survey, Journal of Computer and System Sciences 80 (3) (2014) 602–617.

[29] T. A. Al-Janabi, H. S. Al-Raweshidy, Efficient whale optimisation algorithm-based sdn clustering for iot focused on node density, in: 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2017, pp. 1–6.

[30] C. Gonzalez, S. M. Charfadine, O. Flauzac, F. Nolot, Sdn-based security framework for the iot in distributed grid, in: 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), IEEE, 2016, pp. 1–5.

[31] C. Gonzalez, O. Flauzac, F. Nolot, A. Jara, A novel distributed sdn-secured architecture for the iot, in: 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS), 2016, pp. 244–249.

[32] O. Flauzac, C. Gonzalez, F. Nolot, Developing a distributed software defined networking testbed for iot, Procedia Computer Science 83 (2016) 680–684.

[33] D. Wu, D. I. Arkhipov, E. Asmare, Z. Qin, J. A. McCann, Ubiflow: Mobility management in urban-scale software defined iot, in: 2015 IEEE conference on computer communications (INFOCOM), IEEE, 2015, pp. 208–216.

[34] M. B. Yassein, S. Aljawarneh, M. Al-Rousan, W. Mardini, W. Al-Rashdan, Combined software-defined network (sdn) and internet of things (iot), in: 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), IEEE, 2017, pp. 1–6.

[35] A. C. Baktir, A. Ozgovde, C. Ersoy, How can edge computing benefit from software-defined networking: A survey, use cases, and future directions, IEEE Communications Surveys Tutorials 19 (4) (2017) 2359–2391.

[36] J. Zhao, C. Qiao, R. S. Sudhaakar, S. Yoon, Improve Efficiency and Reliability in Single-Hop WSNs with Transmit-Only Nodes, IEEE Transactions on Parallel and Distributed Systems 24 (3) (2012) 520–534.

[37] L. Xu, G. M. O'Hare, R. Collier, A Balanced Energy-Efficient Multihop Clustering Scheme for Wireless Sensor Networks, in: 2014 7th IFIP Wireless and Mobile Networking Conference (WMNC), IEEE, 2014, pp. 1–8.

[38] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, C. E. Rothenberg, Mininet-wifi: Emulating Software-Defined Wireless Networks, in: Network and Service Management (CNSM), 2015 11th International Conference on, IEEE, 2015, pp. 384–389.

[39] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, A. Khalili, A pairwise key predistribution scheme for wireless sensor networks, ACM Transactions on Information and System Security (TISSEC) 8 (2) (2005) 228–258.

[40] W. Du, J. Deng, Y. S. Han, S. Chen, P. K. Varshney, A Key Management Scheme for Wireless Sensor Networks using Knowledge, in: INFOCOM 2004. Twenty-third AnnualJoint conference of the IEEE computer and communications societies, Vol. 1, IEEE, 2004.