Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 794326, 11 pages
http://dx.doi.org/10.1155/2013/794326

Research Article

Routing Attacks and Countermeasures in the RPL-Based Internet of Things

Linus Wallgren,[1] Shahid Raza,[1] and Thiemo Voigt[1,2]

[1] SICS Swedish ICT, Isafjordsgatan 22, 16440 Kista, Stockholm, Sweden
[2] Department of Information Technology, Uppsala University, Ångströmlaboratoriet, Lägerhyddsvägen 1, 75237 Uppsala, Sweden

Correspondence should be addressed to Shahid Raza; shahid@sics.se

Received 2 November 2012; Revised 26 May 2013; Accepted 5 June 2013

Academic Editor: Donggang Liu

Copyright © 2013 Linus Wallgren et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a novel routing protocol standardized for constrained environments such as 6LoWPAN networks. Providing security in IPv6/RPL connected 6LoWPANs is challenging because the devices are connected to the untrusted Internet and are resource constrained, the communication links are lossy, and the devices use a set of novel IoT technologies such as RPL, 6LoWPAN, and CoAP/CoAPs. In this paper we provide a comprehensive analysis of IoT technologies and their new security capabilities that can be exploited by attackers or IDSs. One of the major contributions in this paper is our implementation and demonstration of well-known routing attacks against 6LoWPAN networks running RPL as a routing protocol. We implement these attacks in the RPL implementation in the Contiki operating system and demonstrate these attacks in the Cooja simulator. Furthermore, we highlight novel security features in the IPv6 protocol and exemplify the use of these features for intrusion detection in the IoT by implementing a lightweight heartbeat protocol.

1. Introduction

Efforts are underway to connect small and large physical objects with the Internet using IPv6 protocols to form the Internet of Things (IoT). The Routing Protocol for Low-Power and Lossy Networks (RPL) [1] is recently standardized as a routing protocol for the IoT. RPL is primarily designed for low-power and lossy networks (LLNs), also called IPv6 over Low-powered Wireless Personal Area Networks (6LoWPAN) networks. A 6LoWPAN network [2] is a Wireless Sensor Network (WSN) that uses compressed IPv6 protocol for networking and IEEE 802.15.4 as a data-link and physical layer protocol. Unlike in typical stand-alone WSNs, the constrained devices in the IoT are accessible from anywhere. Hence, they are exposed to threats both from the Internet and from within the network.

Potentially any physical object can be connected to the IoT using IPv6. There are a large number of applications for the IoT. The application domains include environmental monitoring, home automation and home security management, industrial automation, smart energy monitoring and management, item and shipment tracking, surveillance and military, smart cities, and health monitoring. Real world deployments of the IoT require secure communication which is a challenge because of the heterogeneity of the IoT devices: some are resource constrained and others can be powerful IP-connected hosts. It is also important that the communication between the IoT devices should be secured end to end (E2E) meaning that the confidentiality and the integrity of messages should be enforced between the source and the destination devices. In order to enforce E2E message security in the IoT using standardized protocols we can use IP security (IPsec) or Datagram TLS (DTLS). Research efforts are underway to securely connect constrained nodes in a 6LoWPAN network with the Internet using lightweight compressed IPsec [3], lightweight DTLS [4, 5], and IEEE 802.15.4 link-layer security [6].

Though message security provides confidentiality and integrity of data packets in transit and authentication between devices, an attacker can still launch a number of attacks against the IoT hosts primarily to interrupt the network. Routing attacks are most common in low-power wireless

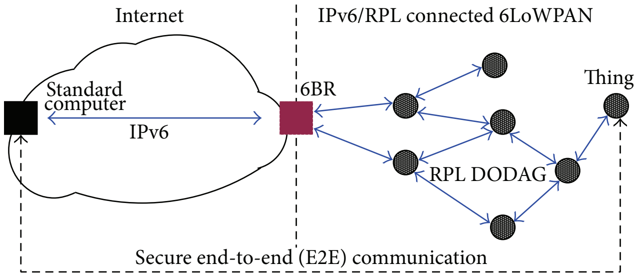

Internet
IPv6/RPL connected 6LoWPAN
Standard computer
IPv6
6BR
Thing
RPL DODAG
Secure end-to-end (E2E) communication

Figure 1: An IoT setup that shows an interconnection of IPv6/RPL connected things in a 6LoWPAN network and the Internet through the 6LoWPAN Border Router (6BR).

networks [7]. In this paper we implement common routing attacks in a 6LoWPAN network where nodes run the Contiki OS [8], the RPL protocol (ContikiRPL [9]) for routing and other novel IoT protocols and show how the RPL protocol behaves in the presence of a particular routing attack.

To counter attacks in a network, Intrusion Detection Systems (IDSs) are used. An IDS analyzes the activities in the network and tries to detect malicious behavior and/or intruders that are trying to disrupt the network. To this end, we investigate the novel IoT protocols/technologies such as CoAP [10], RPL [1], and 6LoWPAN [2] and discuss their strengths and weaknesses which can be exploited by security providers or attackers. Finally, we highlight the new features in the IPv6 protocol that can be used by IDSs or by attackers. To exemplify the use of novel IPv6 security features for intrusion detection we propose and implement a lightweight heartbeat protocol that protects the IoT against selective-forwarding attacks.

The main contributions of this paper are the following.

(i) We investigate how novel features of IoT technologies can be exploited by attackers or IDSs.

(ii) We implement and demonstrate attacks against 6LoWPAN networks running IoT protocols, and we show the effectiveness of well-known routing attacks against RPL and how RPL's self-healing mechanisms protect against some of these attacks.

(iii) We also highlight new security features in the IPv6 protocol and provide a lightweight heartbeat protocol to exemplify that these novel features can be exploited for intrusion detection and mitigation of attacks.

Section 2 discusses IoT technologies with relation to intrusion detection. Section 3 demonstrates attacks against RPL. In Section 4 we discuss IDS in the IoT where we present a heartbeat protocol for the IoT. Finally, Section 5 concludes the paper.

2. IoT Technologies and IDS

In this section we discuss RPL, other IoT technologies, and the novel features in the IoT technologies that can be exploited either by attacks to disrupt networks or by the IDSs to defend against intrusions.

2.1. Internet of Things (IoT). The Internet of Things (IoT) or strictly speaking the IP-connected IoT is a heterogeneous network that consists of the conventional Internet and networks of constrained devices connected together using IP protocol. The networks of constrained devices in the IoT, called 6LoWPAN networks or an IP-connected WSN, are connected to the conventional Internet using 6LoWPAN Border Routers (6BR). Figure 1 shows the interconnection of things in a 6LoWPAN network with the Internet using the 6BR. *Things* in the IoT are uniquely identifiable objects that sense the physical environment and/or the host devices and communicate this data to the Internet. An IoT device (a thing) can be a light bulb, a thermostat, an home appliance, an inventory item, a smartphone, a personal computer, or potentially anything. IPv6 with its potentially unlimited address space can connect billion or even trillion of these devices with the IoT.

The fact that the devices in the IoT are extremely heterogeneous and many of them are resource constrained and are globally connected makes it much more challenging to secure the IoT. The constrained devices in the IoT especially are prone to attacks from the Internet and also from the wireless devices within 6LoWPAN networks. The available IDSs for the Internet and/or for the WSNs may not be suitable to protect IoT devices because they are either too heavyweight for the constrained device or they were not developed in the context of the IoT. Therefore, uncovering the novel requirements of the IoT and providing an IDS for the IoT are worth investigating.

2.2. 6LoWPAN. IPv6 over Low-Power Wireless Personal Area Network [2] (6LoWPAN) is a low cost and low-power communication network which connects resource-constrained wireless devices, typically wireless sensors or actuators, using compressed Internet Protocol version 6 (IPv6). It defines IPv6 header compression [11] and specifies how packets are routed in wireless networks that use the IEEE 802.15.4 protocol at the link and physical layer. It also defines fragmentation [11] of IPv6 datagrams when the size of the datagram is more than the IEEE 802.15.4 Maximum Transmission Unit (MTU) of 127 bytes.

6LoWPAN networks support multihop communication where nodes can forward packets on behalf of other nodes. Energy is one of the scarce resources in 6LoWPAN networks, and usually most of the energy is consumed on idle listening; therefore, 6LoWPAN networks are usually duty cycled meaning that the radio is turned off most of the time and is turned on only for a very short time for listening.

Due to global IP connectivity, 6LoWPAN networks are vulnerable to most of the available attacks against WSNs [12] plus attacks originating from the Internet. Due to the wireless medium and usually unattended deployments, it is easier to compromise 6LoWPAN devices than typical hosts on the Internet. This gives rise to new threats against the core Internet as the compromised 6LoWPAN devices become sources of attacks against conventional Internet hosts. An IDS for 6LoWPAN networks should consider these vulnerabilities. Also, it is important to consider the capabilities of 6LoWPAN devices when designing an IDS.

2.3. CoAP/CoAPs. Due to low-power and lossy links, it is hard to maintain a continuous connection between devices in a 6LoWPAN network. Hence, the connection-less User Datagram Protocol (UDP) is mostly used as the transport layer in 6LoWPAN networks. Further, since connection-oriented web protocols such as HTTP or HTTPs are designed to be used over TCP, a new protocol, the Constrained Application Protocol (CoAP) [10], is being standardized for the IoT. The secure version of CoAP is CoAPs that uses DTLS to protect CoAP messages between two applications in the IoT.

Unlike typical WSNs that have no web protocol, 6LoWPAN networks may use CoAP or CoAPs. Reliability in the CoAP protocol is achieved through the use of *confirmable* messages. An IDS for the IoT can utilize these built-in reliability and security mechanisms in CoAP/CoAPs to protect IoT devices against many known and potential attacks. For example, a well-known attack in the WSN and hence in 6LoWPAN is the HELLO flood [12] that can be detected using reliability mechanisms in the CoAP protocol where devices can check the bidirectionality of paths through CoAP acknowledgments.

2.4. RPL. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [1] is a standardized routing protocol for the IoT. RPL is primarily used in a 6LoWPAN network. RPL creates a destination-oriented directed acyclic graph (DODAG) between the nodes in a 6LoWPAN. It supports unidirectional traffic towards a DODAG root and bidirectional traffic between 6LoWPAN devices and between devices and the DODAG root (typically the 6BR). There may exist multiple *global* RPL instances for a single 6LoWPAN network, and a *local* RPL DODAG can be created among a set of nodes inside a global DODAG. In Figure 2 an RPL DODAG is shown where each node has a node ID (an IPv6 address), a list of neighbors, and a parent node. Each node in a DODAG has a rank that indicates the position of a node relative to other nodes and with respect to the DODAG root. Ranks strictly decrease in the *up* direction towards the DODAG root and strictly increase from the DODAG root towards nodes.

In order to support downward routing either source routing (RPL nonstoring mode) or stateful in-network routing tables (RPL storing mode) are used. Source routing means each packet contains the route the packet is supposed to take through the network. This requires that the DODAG root keeps the information about each node in the network. In a nonstoring mode, all forwarding nodes in an RPL DODAG must maintain in-network routing tables to know where to send packets; in-network routing tables differentiate between the packets heading upwards and the packets traveling downwards in the network. For both modes described previously the RPL DODAG root maintains a complete list of nodes to support downward traffic.

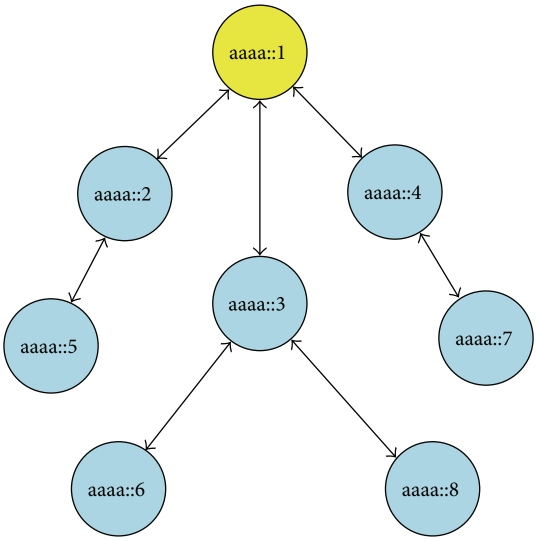


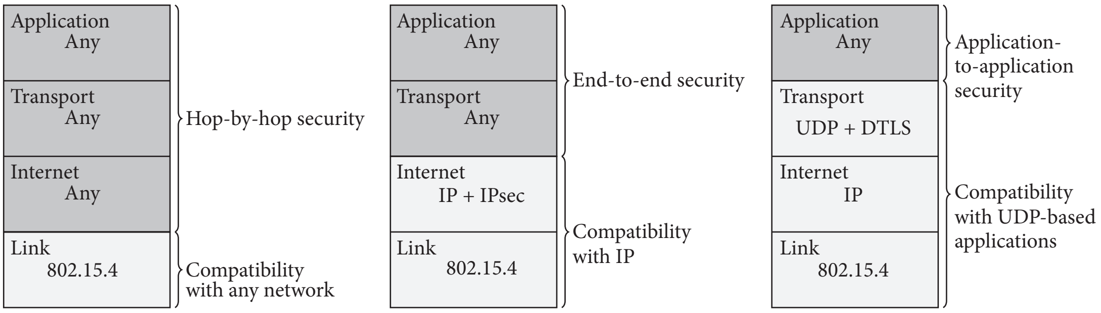
Figure 2: A sample RPL DODAG where each node has a unique IPv6 address.

RPL enables each node in the network to determine whether packets are to be forwarded upwards to their parents or downwards to their children. Typically, as in the case in ContikiRPL [9] that we use to demonstrate attacks in this paper, the simplest way a node can determine the direction of a packet is to know all its descendants which determines the route towards leaf nodes and consider up direction as the default route of a packet. In RPL storing mode, in-network routing tables are used to separate packets heading upwards and the packets heading downwards in the network.

The RPL protocol provides new ICMPv6 control messages to exchange routing graph information. RPL DODAG Information Objects (DIO) are used to advertise information that are used to build the RPL DODAG. Destination Advertisement Object (DAO) messages are used to advertise information required to support downward traffic towards leaf nodes. Each child node upon joining sends a DAO message to its parents; also, parent nodes can explicitly poll the sub-DODAG for DAO messages using DIO messages. Nodes may use DODAG Information Solicitation (DIS) messages to request graph related information from the neighboring nodes.

The RPL protocol could be vulnerable to the routing attacks demonstrated against WSNs [12] and also to the attacks against the IoT [7]; therefore it is worth investigating the routing attacks against RPL, inherent protection mechanisms in RPL, and new intrusion detection mechanisms for RPL-based networks. We discuss attacks in RPL networks in Section 3.

Self-Healing in RPL. RPL has global and local repair mechanisms that can come into action if there is a routing topology failure, a link failure, or a node failure. On a node (parent) or a link failure a local repair mechanism tries to select a new



(a) Link-layer security (b) Internet-layer security (c) Transport-layer security

Figure 3: Communications can be secured at different layers of the protocol stack, and each solution has its own pros and cons and has its own scope and level of interoperability.

parent or path. If there are more local failures, RPL performs a complementary global repair where the whole DODAG is rebuilt. The RPL protocol uses the link-layer metric as a parameter in the calculation of a default route. The path is assumed to be good if link-layer acknowledgements are received on it.

RPL also uses a trickle timer to handle inconsistencies in the RPL DODAG. When an RPL network is stable, the trickle timer interval is large. However, upon detection of inconsistencies, the trickle timer is reset, and more DIO messages are sent (by the nodes) in the vicinity of nodes that are subjected to inconsistencies. The following events are considered as inconsistencies in the RPL:

(i) when routing loops are detected;

(ii) when a node joins a DODAG;

(iii) when a node moves within a network and changes rank.

2.5. Message Security for the IoT. Security is one of the main requirements in real world deployments of the IoT. Security can be provided on per hop basis between two neighboring devices in 6LoWPANs, and/or it can be provided end to end (E2E) between source and destination nodes. Per hop security is important to grant access to the wireless medium and to detect message integrity violations as early as possible to hinder constrained resource depletion. Message security in the IoT can be enabled at different layers in the stack using standardized mechanisms; security at the data-link layer using standardized IEEE 802.15.4 security protects messages on a per hop basis but works with any networking and communication protocol at the upper layer. In addition to the actual messages it can protect the data-link layer and upper layer headers as well. Previously, we have implemented and evaluated IEEE 802.15.4 data-link layer security in the 6LoWPAN [6]. RPL also provides per hop security between two neighboring nodes which protects the RPL messages. Security at the routing layer (i.e., at the RPL layer) is not needed if link layer security (i.e., per hop security) is enabled. Also, it is more secure to provide security at the link layer because it can protect the integrity of the link-layer and upper layers (that include RPL) as well and can even encrypt the 6LoWPAN layer and upper layer headers and payloads including the RPL messages.

Security at the IP layer using standardized IPsec is E2E between two hosts on the Internet and works with both TCP and UDP protocols. We have previously provided lightweight 6LoWPAN compressed IPsec for the IoT [3]. Transport/session layer security protects messages between two applications on an E2E basis but only works with one of the transport protocol such as TCP or UDP. In the IoT, UDP is mostly used, and hence standardized Datagram TLS (DTLS) can be used. Earlier, we have provided 6LoWPAN header compression for DTLS [5] to make it lightweight for the constrained devices in the IoT.

Despite message security with any of the previous mechanisms, IoT devices are still vulnerable to network disruptions, such as DoS attacks. An IDS for the IoT should consider and/or utilize the standardized message security technologies discussed previously. Figure 3 summarizes the pros and cons of providing security at different layers.

2.6. Intrusion Detection Systems. An Intrusion Detection System (IDS) analyzes activities or processes in a network or in a device and detects attacks, reports them, and/or mitigates the harmful effect of the detected attacks. Due to the diversity of attacks and the unpredictable behavior of novel attacks, IDSs are subjected to false positives (to raise an alarm when there is no attack) and false negatives (not raising an alarm when there is an attack). Generally, there are two categories of IDSs: signature based and anomaly based. Signature based detections compare the current activities in a network or in a device against predefined and stored attack patterns called signatures. This approach cannot detect new attacks, needs specific knowledge of each attack, has a significant storage cost that grows with the number of attacks, and has a high false negative but low false positive rate. Anomaly based detections determine the ordinary behavior of a network or a device, use it as a baseline, and detect anomalies when there are deviations from the baseline. This approach can detect new attacks but has comparatively high false positive and false negative rates because it may raise false alarms and/or cannot detect attack when attacks only show small deviations from the baseline.

Table 1: The IoT technologies at different layers with example open source implementations in the Contiki operating system.

OSI layer	IoT technology	Contiki impl.
Application	CoAP, CoAPs	Erbium [16]
Session	DTLS	TinyDTLS [http://tinydtls.sourceforge.net]
Transport	UDP	μIP [17]
Network	IPv6, RPL, IPsec	μIP [17], ContikiRPL [9], and IPsec in Contiki [6]
	6LoWPAN	SICSLoWPAN [8]
Data link	802.15.4 MAC	ContikiMAC [18]
Physical	802.15.4 PHY	Contiki 802.15.4 [8]

An IDS for 6LoWPAN networks requires a trade-off between the storage cost of the signature based detection and the computing cost of the anomaly based techniques, should counter attackers from the conventional Internet, and should consider that the attackers in a 6LoWPAN can harm both the 6LoWPAN network and the Internet. We propose to complement an IDS for the IoT with a firewall that can be typically placed in the 6BR. Unlike a typical one-way firewall, a firewall for the IoT should block malicious activities and allow benign activities from the Internet to 6LoWPAN networks, and vice versa. In our previous work we have developed a real-time intrusion detection system in the context of the IoT [13].

3. Attacks against RPL

In this section we investigate the protection capabilities of the RPL protocol against the well-known security attacks presented for WSNs. We experimentally study if the RPL protocol can counter these attacks and/or mitigate their impact.

Attack Implementation. We implement well-known routing attacks in a 6LoWPAN network where nodes run the Contiki OS [8], a well-known operating system for the IoT. Contiki has an implementation of RPL, ContikiRPL [9]. We make use of the RPL implementation in the Contiki OS to implement attacks. ContikiRPL storing mode uses in-network routing where nodes keep track of all descendants. To provide IP communication in 6LoWPAN we utilize μIP, an IP stack in the Contiki OS. We demonstrate attacks against a simulated RPL network using the Cooja simulator [14]. In our simulations we use emulated Tmote Sky nodes [15] running ContikiRPL.

In Table 1 we highlight the standardized IoT technologies at different layers of the protocol stack that are expected to be used in most of the IoT deployments that rely on interoperability among different vendors. We also mentioned the corresponding open source implementations (ContikiMAC is not a true implementation of the 802.15.4 MAC) of these IoT technologies in the Contiki OS which we use in this paper to demonstrate attacks against the RPL protocol.

3.1. Selective-Forwarding Attacks. With selective-forwarding attacks [12] it is possible to launch DoS attacks where malicious nodes selectively forward packets. This attack is primarily targeted to disrupt routing paths; however, it can be use to filter any protocol. For example, an attacker could forward all RPL control messages and drop the rest of the traffic. This attack has severer consequences when coupled with other attacks, for example, sinkhole attacks.

One of the solutions to guard against selective-forwarding attacks is to create disjoint paths between the source and the destination nodes. However, it is quite hard to create network-wide completely disjoint paths. To counter selective-forwarding attacks, nodes in the RPL may *dynamically* select the paths to parents/children; as there may be multiple parent or child nodes in the RPL DODAG with almost the same link quality. Also, RPL supports source routing, though not widely implemented, that can be used by an IDS for the IoT to verify path availability in the DODAG.

It is generally very difficult to defend against all selective-forwarding attacks. One can, however, defend against many with the use of encryption and analysis of application level traffic. That is to detect if any application traffic is lost and report such losses to the underlying RPL system in order to improve path quality. Another effective countermeasure against selective-forwarding attacks is to make sure the attacker cannot distinguish between different types of traffic, thus forcing the attacker to either forward all traffic or none. In IPv6, ICMPv6 messages are protected by IPsec; hence IPsec can be used to secure the RPL control messages DIO, DAO, and DIS.

3.1.1. Implementing Selective-Forwarding Attacks against RPL. In our implementation of the selective-forwarding attacks we let the malicious node drop all packets except RPL packets. As specified in Algorithm 1, we check in the malicious node running Contiki OS and ContikiRPL that if the received packet is not destined to the malicious node and is not an RPL packet, it is dropped. Our selective-forwarding attack allows for RPL to function normally, but any application data is lost. We simulate this attack in Cooja, and through serial output from the nodes we can verify that the application data is in fact lost from children to the attacker. We run the simulation for 24 hours to allow RPL self-healing and self-management mechanisms to correct this malicious behavior; however, we could see through the output of the malicious node and its parent node that the attack is still active. This means the malicious node still drops all packets except of RPL messages, which shows that, even after running simulation for 24 hours, the RPL self-healing mechanisms

```
Require: Packet—The IPv6 packet received
Require: OwnIP—The IPv6 address of this node
  if $Packet.protocol \neq$ RPL and $Packet.destination \neq OwnIP$ then
    Drop packet
  end if
```

Algorithm 1: Selective-forwarding attacks in RPL.

cannot self-correct the network. Therefore, an IDS for the IoT running RPL in 6LoWPAN networks should actively provide countermeasures to detect selective-forwarding attacks.

3.2. Sinkhole Attacks. In *sinkhole attacks* [12] a malicious node advertises an artificial beneficial routing path and attracts many nearby nodes to route traffic through it. This attack in itself does not necessarily disrupt the network operation; however when coupled with another attack, it can become very powerful. Ngai et al. present an IDS [19] against sinkhole attacks. Their approach requires two-way communication with the nodes and encryption of the messages. RPL already uses IP and has standardized ways to provide bidirectional communication; E2E message security is enforced using IPsec which is mandatory in IPv6.

Routing protocols that do not use metrics provided by neighboring nodes are immune to sinkhole attacks, as there is nothing for the attacker to spoof. For example, this is the case with preprogrammed routes. The RPL protocol provides several mechanisms to nodes in the DODAG to determine which node to use as its default route. One of them is rank, which is calculated and transmitted by the neighboring nodes, though based on the relative position of nodes from the DODAG root. An attacker can launch a sinkhole by advertising a better *rank* thus attracting nodes down in the DODAG to select it as parent. RPL, however, uses the link-layer quality to calculate routes which makes sinkhole attack less effective in RPL-based networks.

If the geographical locations of the nodes in the RPL DODAG are known, the effect of sinkhole attacks can be mitigated by using flow control and making sure that the messages are traveling towards the actual destination. RPL also supports multiple DODAG *instances* which provides alternative routes to the DODAG root. A potential IDS for the IoT could be hosted in the 6BR and can utilize information from multiple DODAGs to detect sinkhole attacks.

3.2.1. Implementing Sinkhole Attacks against RPL. We implement a sinkhole attack in a Cooja simulated RPL network by simply changing the advertised rank when sending RPL control messages, specifically the DIO messages. Any delay normally used to reduce network congestion is also removed in order to allow our malicious node to be the first node to advertise such a beneficial route. Figure 4 shows that the sinkhole attack is very effective against an ordinary RPL network and causes a lot of traffic to get routed through the attacker. Figure 4 shows node number 26 performing a sinkhole attack. Most of the nodes down in the DODAG select it as their parent. We run this simulation for 24 hours to let RPL DODAG correct itself against the malicious behavior; however, we see no noticeable changes in the network state, and the sinkhole attack is still effective, except that the nodes with bad links to node 26 choose different parents.

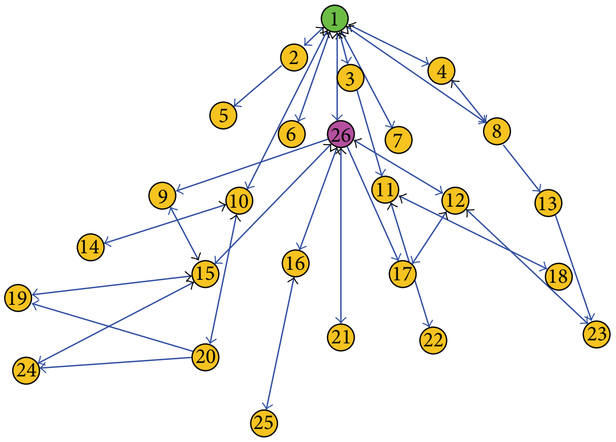
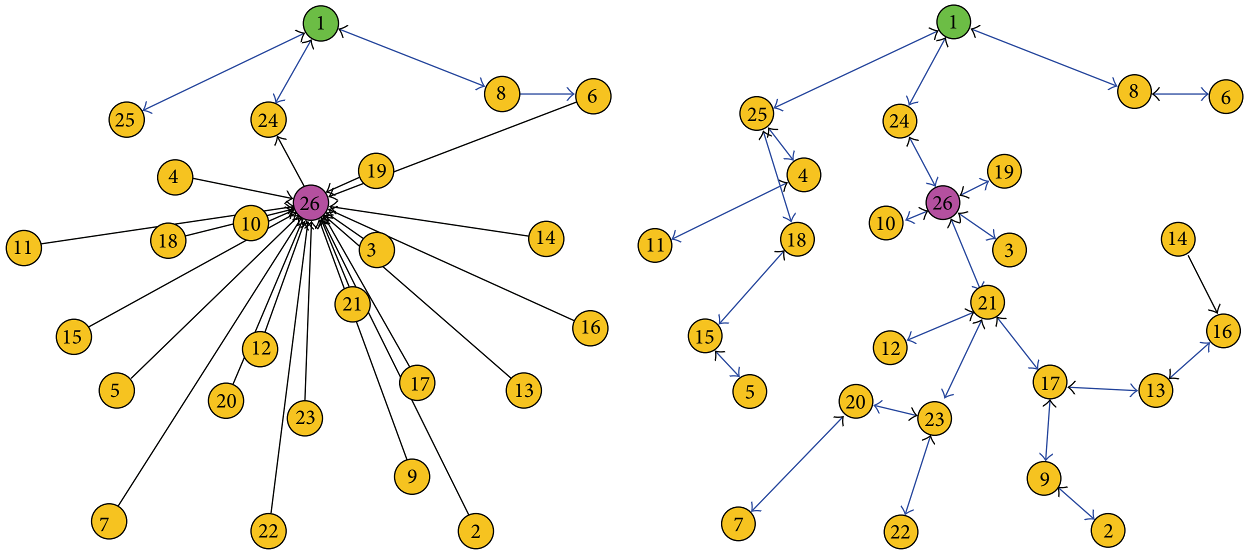


Figure 4: Screenshot of a simulated RPL network with sinkhole attack, running actually implemented IoT technologies, shows that RPL is effected by sinkhole attacks.

3.3. HELLO Flood Attacks. The HELLO message refers to the initial message a node sends when joining a network. By broadcasting a “HELLO” message with strong signal power and a favorable routing metric, an attacker can introduce himself as a neighbor to many nodes, possibly the entire network; however, in some of the nodes in the attacker’s vicinity, when trying to join the attacker, their messages may get lost because the attacker might be out of range.

In RPL, DIO messages that are used to advertise information about DODAGs to new nodes can potentially be used to launch a HELLO flood attack. If *secure* DIO messages are used for advertisements or link-layer security is enabled, the attacker has to compromise a node in order to perform this attack.

Karlof and Wagner suggest a simple solution to this attack where for each HELLO message the link is checked to be bidirectional [12]. This solution is similar to what is already available in the RPL protocol where it uses the link-layer metric as a parameter in the calculation of the default route. If no link-layer acknowledgements are received, the path is assumed to be bad, and a different route is chosen.



(a) RPL network with HELLO flood attack at the start of simulation

(b) RPL network with HELLO flood attack after 10 minutes of simulation

Figure 5: Cooja screenshot of an RPL network running actually implemented IoT technologies shows that RPL self-healing mechanisms overcome HELLO flood attack without any designated IDS.

If geographical locations of the nodes in the RPL DODAG are known, all packets received from a node that is far beyond the transmission capabilities of ordinary network nodes could be discarded to mitigate HELLO flood attacks. The self-healing mechanisms in the RPL, discussed in Section 2.4, may overcome this attack by trying another parent.

3.3.1. Implementing HELLO Flood Attacks against RPL. We implement a HELLO flood attack against an RPL network and let the RPL self-healing mechanism counter the attack. Using the Cooja simulator we alter the connectivity between the simulated nodes in the RPL network. We thus simulate a HELLO flood by letting a malicious node have the ability to send data to all other nodes in the network; however only nodes physically close to the attacker have the ability to respond. In order to increase the efficiency of the HELLO flood attack we combine it with a sinkhole attack, described in Section 3.2.

At first the HELLO flood attack interrupts the network as almost all nodes in the network choose the attacker (node 26) as its default route, as shown in Figure 5(a). However, nodes soon realize that the attacker is in fact not a valid route, and choose a different default route. We show in Figure 5(b) that the state of the network changes using RPL inherent mechanisms, and the HELLO flood attack is automatically mitigated within 10 minutes of its launch. However, nodes 3, 10, 19, and 21 are still connected through the malicious node 26 which shows that the *sinkhole* attack is not fully eliminated.

3.4. Wormhole Attacks. A wormhole is an out of band connection between two nodes using wired or wireless links. Wormholes can be used to forward packets faster than via normal paths. A wormhole in itself is not necessarily a breach security; for example, a wormhole can be used to forward mission critical messages where high throughput is important, and the rest of the traffic follows the normal path. However, a wormhole created by an attacker and combined with another attacks, such as sinkhole, is a serious security threat.

As we discussed in Section 4.1, an IDS for the IoT could place processing intensive modules and a firewall in the 6BR. An attacker can create a wormhole between a compromised constrained node in a 6LoWPAN network and a typical device on the Internet and can bypass the 6BR. Such a wormhole can become a very serious security breach and is very hard to detect especially when the wormhole is systematically switched on and off. Ways to prevent or at least detect such a wormhole in the IoT are a research challenge that needs to be addressed.

It is comparatively easy to detect wormholes created within an RPL DODAG. One approach is to use separate link-layer keys for different segments of the network. This can counteract the wormhole attack as no communication will be possible between nodes in two separate segments. Also, by binding geographic information to the neighborhoods it is possible to overcome a wormhole [20]. As wormholes are usually coupled with other attacks, detecting the other attack and removing/avoiding the malicious node will ultimately overcome wormhole attacks.

3.4.1. Implementing Wormhole Attacks against RPL. We simulate a wormhole attack by using the network simulator Cooja and set up a physical medium where two nodes on opposite sides of the network have a very good connection. As nodes 2 and 25, shown in Figure 6, are subjected to a wormhole attack, they form a high quality route, and the neighboring nodes connect through the malicious nodes 2 and 5. We run this

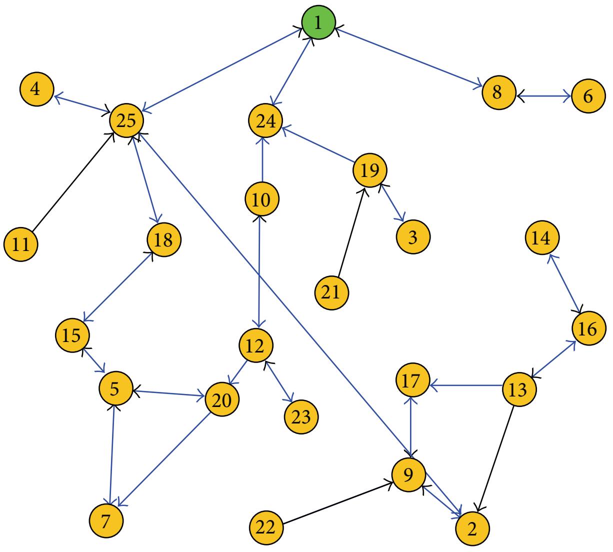


Figure 6: Screenshot of the simulated RPL network, running actually implemented IoT technologies, shows that RPL is effected by wormhole attacks.

simulation for 24 hours to allow RPL inherent mechanisms to self-heal the RPL DODAG. However, the network state has shown that the attack is still there after 24 hours which means that RPL does not provide any specific mechanisms to counter wormhole attacks.

3.5. Clone ID and Sybil Attacks. In a clone ID attack, an attacker copies the identities of a valid node onto another physical node. This can, for example, be used in order to gain access to a larger part of the network or in order to overcome voting schemes. In a sybil attack, which is similar to a clone ID attack, an attacker uses several logical entities on the same physical node. Sybil attacks can be used to take control over large parts of a network without deploying physical nodes.

By keeping track of the number of instances of each identity it is possible to detect cloned identities. It would also be possible to detect cloned identities by knowing the geographical location of the nodes, as no identity should be able to be at several places at the same time. The location of nodes or similar information could be stored either centralized in the 6BR or distributed throughout the network in a distributed hash table (DHT) [21].

In an IP/RPL network cloned identities will cause trouble when packets are heading to one of the cloned identities. Packets will be forwarded to *one* of the cloned identities based on the routing metrics in the network, and the rest of the cloned identities will be unreachable from certain nodes in the network. This however does not affect the network otherwise, and therefore cloned identities on their own cause no harm on a 6LoWPAN network.

3.5.1. Implementing Clone ID Attacks against RPL. Using the Cooja network simulator we simulate cloned identities by disabling the multiple-id check in the simulator and simply add several nodes with the same ID. In our Cooja simulated IPv6 network running RPL the cloned identities have the same IP address.

Simulations show that there are no inherent mechanisms in RPL to counter cloned identities. In Figure 7 the cloned identities are indicated in purple and have ID 26. The paths shown with blue arrows represent the downward path. The downward paths from the cloned identities are visualized correct as Cooja nodes in such cases are aware of the source node. However, all nodes which have chosen one of the cloned nodes as their parent will all have their upwards route, the black arrows, pointing towards the leftmost cloned node.

RPL is also subject to alteration and spoofing routing attacks. In RPL a malicious node can send modified rank information to the neighboring nodes. It can also send modified or spoofed DIS, DIO, and DAO messages if RPL security is disabled which is the typical case. The 6LoWPAN networks can also suffer from traffic analysis that in itself is not disrupting, but the information obtained from analyzing the traffic could be used to launch other sophisticated attacks. Typically, IPsec in tunnel mode or traffic randomization with extra generated traffic is used to counter these attacks. However, the constrained nature of the IoT devices precludes the applicability of these countermeasures. Usually attacks are not performed in isolation and are combined to get more gains. An IDS for the IoT should consider different possible combinations of these attacks and device solutions to protect network against multiple attacks.

4. IDS and the IoT

In this section we present a placement of an IDS in a novel IoT setup and propose a mechanism to eliminate malicious nodes in the RPL network. We also discuss intrusion detection capabilities of IPv6 through the heartbeat protocol.

4.1. Placement of an IDS in the IoT. Unlike typical WSN that assume no constant connectivity with the sink node, in the IoT the sink node (the 6BR) is assumed to be always available and is not the end point of communication, but rather *things* are globally recognizable. This novel architecture, as shown in Figure 1, based on standardized protocol such as RPL and 6LoWPAN, gives us more flexibility in the placement of IDSs.

An IDS for the IoT can better utilize this architecture and place processing intensive IDS modules, such as anomaly based detections, in the 6BR, and the corresponding lightweight modules, such as rule or signature based detections, in the constrained sensor nodes. As already discussed, the 6BR has more capacities than a typical resource-constrained sensor node. Any such distributed architecture, however, requires a trade-off between the local storage/processing and network communication. Placement of IDS modules in constrained devices will require more storage and processing capabilities; however, these devices have limited resources. On the other hand, a placement of an IDS in the 6BR requires a fresh state of the network, which ultimately incurs more communication overhead between sensors and the 6BR. In LLNs, sending and receiving bits

```
Require: Hosts—A list of hosts in the RPL DODAG
Require: Responses—A list of ICMPv6 Echo
  Replies from the previous iteration of this algorithm
  for Host in Hosts do
    ICMP.sendEchoRequest(Host)
  end for
  for Respons in Responses do
    Hosts.remove(Respons.source)
  end for
  for Host in Hosts do
    Alarm.raise("Host is offline or filetered", Host)
  end for
```

Algorithm 2: Lightweight heartbeat.

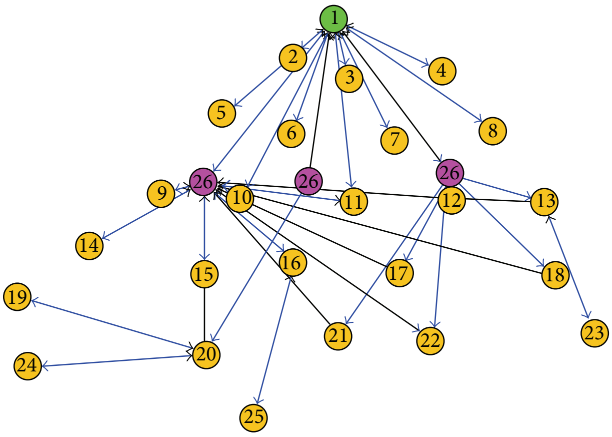


Figure 7: Screenshot of the simulated RPL network, running actually implemented IoT technologies, visualizes the cloned identities and shows that RPL is effected by clone ID attacks.

are more power consuming than local processing. Hence it is worth evaluating an IDS approach in both the centralized and distributed placements to better understand its applicability in the IP-connected LLNs. IDS modules in the 6BR have the additional advantage that they can stop intrusion attempts from the Internet. Also, they can block intrusion attempts from inside LLNs against critical infrastructure on the Internet. This is useful since it is easier to physically access and compromise wireless nodes than typical Internet hosts.

4.2. Eliminating Malicious Nodes from RPL. Once nodes are detected as malicious it is important to eliminate these nodes from the network. The simplest approach to avoid a fake node is to ignore it which requires identification. In the IoT, both IP addresses and MAC addresses are vulnerable and can be easily spoofed. One possible way to ignore malicious nodes is to use either a whitelist or a blacklist. A whitelist contains all legitimate nodes, whereas a blacklist would include all malicious nodes. On one hand maintaining a whitelist is easier; on the other hand it is not very scalable. Considering that there will be limited devices under one 6BR or in a single RPL DODAG we propose to use a whitelist as it is easy to be managed in the presence of many attackers. However, there can be potentially thousands of devices in an RPL network. In such large networks blacklists are easier to manage. In either way it is important that an attacker should not be able to obtain another valid identity since that would enable sybil or clone ID attacks [12].

4.3. Intrusion Detection and IPv6. Compared to IPv4 that is mostly used in the Internet today and is well tested, IPv6 is a new protocol and is not yet widely deployed. IPv6 also provides some novel features that can be exploited by both the security provider and the attacker. For example, the flow label field in IPv6 is not protected by IPsec E2E security. Unlike in IPv4, IPsec is mandatory in IPv6. Further, in IPv6 ICMPv6 is protected by IPsec. In this section we use IPsec protected ICMPv6 echo messages and provide a lightweight solution to defend against selective-forwarding attacks in the IoT that are otherwise difficult to detect.

Lightweight Heartbeat. For a 6LoWPAN network, for running RPL or any other IPv6 based routing scheme, we can use a simple heartbeat. Our heartbeat protocol is described in Algorithm 2. In this algorithm, we simply send an ICMPv6 echo request from the 6BR to each node and expect a response. We will notice if traffic is being filtered to and/or from that node if we do not receive an ICMPv6 echo reply. We do this with regular intervals, called heartbeats, to have an up-to-date picture of the state of the network. ICMPv6 echo/reply mechanisms are widely available in IPv6 networks; hence it is not required that the nodes should be reprogrammed to support ICMPv6. For example, in many Contiki OS configurations it is enabled by default.

The heartbeat protocol will work with its full potential if IPsec with ESP [22] is used. Without IPsec, this method will only be able to detect the simplest attacks if there are faults in the networks for other reasons, for example, a broken node. This is because without IPsec, it is possible for an attacker to simply choose not filter ICMPv6 packets and therefore avoid being detected by this technique. The lightweight heartbeat in an IPsec enabled network would be able to detect selective-forwarding attacks as there is no way to distinguish between ICMPv6 traffic and normal traffic as everything after the IPv6

Table 2: Energy and power usage of one node in an RPL network for one heartbeat compared with RPL only (no heartbeat).

Overhead	Energy (mJ)	Power (mW)
RPL only	202.6	1.702
RPL and heartbeat	225.3	1.893
Heartbeat only	22.7	0.191

ESP extension header is encrypted, including the ICMPv6 extension header [23]. The heartbeat concept can be extended to potentially detect many attacks, for example, jamming or physically damaging nodes, since the nodes would stop responding to ICMPv6 requests.

As a proof-of-concept we implement the heartbeat protocol in a 6LoWPAN network running ContikiRPL and other IoT technologies shown in Table 1. We measured the ROM/RAM and energy overhead of the heartbeat protocol. No additional ROM or RAM is used in the constrained nodes as ICMPv6 is already available in most of the IPv6 implementations including the μIP in the Contiki OS. However, each constrained node in the 6LoWPAN network consumes 0.1158 mJ of additional energy to process a single ICMPv6 message. The heartbeat protocol has a little ROM/RAM overhead in the 6BR that sends ICMPv6 messages to the nodes in the 6LoWPAN; however, in the IoT the 6BR is not assumed to be a constrained device.

We also evaluate the network-wide energy overhead of our lightweight heartbeat where the 6BR sends ICMPv6 echo requests to all nodes, and each node handles its ICMPv6 reply and routes replies on behalf of other nodes. In this experiment the RPL DODAG consists of 16 emulated Tmote Sky nodes. Total energy and power usage by a single node (on average) for one ICMPv6 message from the 6BR to all nodes are shown in Table 2.

An IDS for the IoT should take into account the other unexplored IPv6 features to protect the IoT devices against potential malicious activities. We plan to explore this in the future.

5. Conclusion

In this paper we have reviewed novel IoT protocols and highlighted their strengths and weaknesses that can be exploited by the IDSs. We have shown that, while the RPL protocol is vulnerable to different routing attacks, it has inherent mechanisms to counter HELLO flood attacks and mitigate the effects of sinkhole attacks. An IDS for the IoT can be complemented with the novel security mechanisms in the IPv6 protocol; for example, our heartbeat protocol can defend against selective-forwarding attacks.

The aim of this paper is to highlight the importance of security in the RPL-based IoT and to provide grounds to the future researchers who plan to design and implement IDSs for the IoT.

Acknowledgments

This work was financed by the SICS Center for Networked Systems (CNS), SSF through the Promos Project, and Connect All IP-based Smart Objects (CALIPSO), funded by the European Commission under FP7 with Contract no. FP7-ICT-2011.1.3-288879.

References

[1] T. Winter, P. Thubert, A. Brandt et al., "RPL: IPv6 routing protocol for low-power and lossynetworks," RFC 6550, March 2012.

[2] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personalArea networks (6LoWPANs): overview, assumptions, problem statement, and goals," RFC 4919, 2007.

[3] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *Proceeding of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, Barcelona, Spain, June 2011.

[4] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, *DTLS Based Security and Twowayauthentication for the Internet of Things*, Ad Hoc Networks, 2013.

[5] S. Raza, D. Trabalza, and T. Voigt, "6low-pan compressed dtls for coap," in *Proceeding IEEE 8th International Conference of Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 287–289, IEEE, 2012.

[6] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, *Secure Communication for the Internet of Things—A Comparison of Link- Layer Security and IPsec for 6LoWPAN*, Security and Communication Networks, John Wiley & Sons, 2012.

[7] O. Garcia-Morchon, R. Hummen, S. S. Kumar, R. Struik, and S. L. Keoh, "Security Considerations in the IP-based Internet of Things," March 2012.

[8] A. Dunkels et al., "The contiki operatingsystem," 2012, http://www.sics.se/contiki/.

[9] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL," in *Proceeding of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, pp. 406–407, ACM, April 2010.

[10] Z. Shelby, K. Kartke, C. Bormann, and B. Frank, "Constrained application protocol(CoAP)," draft-ietf-core-coap-12, October 2012.

[11] J. Hui and P. Thubert, "Compression format for IPv6 datagrams over IEEE 802.15.4-basednetworks," RFC 6282, 2011.

[12] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2, pp. 293–315, 2003.

[13] S. Raza, L. Wallgren, and T. Voigt, *SVELTE: Real-Time Intrusion Detection in the Internetof Things*, Ad Hoc Networks, Elsevier, 2013.

[14] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceeding of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06)*, pp. 641–648, IEEE, November 2006.

[15] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceeding of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, April 2005.

[16] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power CoAP for Contiki," in *Proceeding of 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '11)*, pp. 855–860, IEEE, October 2011.

[17] A. Dunkels, "Full tcp/ip for 8-bit architectures," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pp. 85–98, ACM, 2003.

[18] A. Dunkels, "The ContikiMAC radio duty cycling protocol," SICS Technical Report T2011:13, 2011.

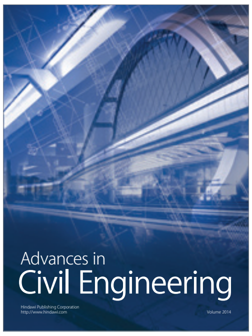
[19] E. C. H. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *Proceeding of the IEEE International Conference on Communications (ICC '06)*, vol. 8, pp. 3383–3389, IEEE, 2006.

[20] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang, "Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach," in *Proceeding of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 2, pp. 1193–1199, March 2005.

[21] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," in *Proceeding of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 259–268, ACM, April 2004.

[22] S. Kent, "Ip encapsulating security payload(esp)," RFC 4303, 2005.

[23] A. Conta, S. Deering, and M. Gupta, "Internet control message protocol (ICMPv6) forthe internet protocol version 6 (IPv6) specification," RFC 4443 (Draft Standard), 2006.



Hindawi

Submit your manuscripts at
http://www.hindawi.com