# Wormhole Attacks als Sicherheitsrisiko in Wireless Sensor Networks und Gegenmaßnahmen von RPL

Marco Mühl, Universität Passau
Lehrstuhl Intelligente Systeme
MUEHL01@gw.uni-passau.de

***Abstract*—Sicherheit spielt in nahezu jedem Netzwerk eine große Rolle. Angreifer finden in der heutigen Zeit immer fortgeschrittenere und komplexere Methoden, eine ernstzunehmende Bedrohung darzustellen. Insbesondere in Wireless Sensor Networks (WSN) ist es deshalb wichtig, dem entgegenzuwirken. Dies stellt sich aber häufig als große Herausforderung dar, weil meist ein Kompromiss zwischen Sicherheit, Aufwand und somit Performance gemacht werden muss. Im Folgenden wird eine der häufigsten Angriffsmethoden auf WSNs, der Wormhole-Attack, und Lösungsansätze des Routing Protocol for Low Power and Lossy Networks (RPL) dafür beschrieben.**

***Keywords—Wireless sensor networks, Routing Protocol for Low Power and Lossy Networks, Wormhole-Attack, ad-hoc networking, Sicherheit***

## I. Einleitung

Für WSNs ist es aufgrund der oft schwierigen Umgebung der Hardware wichtig, dass die Knoten so energiesparend wie möglich kommunizieren können. Deshalb eignen sich Standard-Protokolle wie AODV oder OLSR [4] nicht als Routing Protokoll für solche Anwendungsfälle. Speziell dafür wurde RPL entwickelt. Es ist ein Distance-Vector Routing Protokoll, das auf die Herausforderungen von WSNs mit kritischen Einflüssen der Umgebung abgestimmt ist. Um noch besser mit korrupten Knoten umzugehen wurde eine Erweiterung für das Standard RPL entwickelt: Trust and Forgiveness. Im Folgenden wird RPL und diese Erweiterung vorgestellt. Außerdem wird einer der häufigsten Angriffe auf WSNs erläutert und anschließend wird gezeigt, wie RPL mit und ohne der Erweiterung mit diesen umgeht. Die Ergebnisse werden bewertet.

Im Rahmen des Internet of Things [11] und ähnlich gelagerter Initiativen werden immer mehr Geräte miteinander verknüpft. Dabei sind vielfach auch nur einfachste Sensoren berücksichtigt. Hier stellen sich Fragestellungen, wie dort eine ressourcenschonende und effiziente Kommunikation gewährleistet werden kann. Ansätze sind bspw. im Bereich von mobilen ad-hoc Netzen zu finden. Dies führt aber gleichzeitig zu neuartigen Herausforderungen: Angriffe und Fehlfunktionen müssen zu einer Isolation der Knoten aus dem Kommunikationsgraph führen. Im Rahmen dieses Papers wird jetzt untersucht, wie sich eine Wormhole-Attack auf ein solches Netzwerk auswirkt und wie RPL und die Erweiterung Trust and Forgiveness damit umgeht.

Dieser Artikel ist wie folgt aufgebaut: In Kapitel 2 werden Standard-Protokolle vorgestellt, die in mobilen ad-hoc Netzwerken Verwendung finden. Dazu wird ein kurzer Überblick zu den verschiedenen Ansätzen solcher Protokolle gegeben. Des Weiteren wird die Funktionsweise von Standard-RPL und der Erweiterung Trust and Forgiveness erläutert. In Kapitel 3 wird die grundlegende Funktionalität sowie der Zusammenhang mit Organic Computing eines mobilen ad-hoc Netzwerks erklärt und anschließend gezeigt, welche Auswirkungen eine Wormhole-Attack auf ein solches Netzwerk hat. Wie RPL mit solchen Angriffen umgeht wird in Kapitel 4, mit Unterscheidung ob mit oder ohne der Erweiterung Trust and Forgiveness, näher betrachtet. In Kapitel 5 werden die Ergebnisse des Vergleiches von RPL mit und ohne Erweiterung bei einer Wormhole-Attack vorgestellt und bewertet. Kapitel 6 ist ein kleiner Ausblick für weitere Arbeiten. Zum Schluss findet sich die Referenzliteratur, die für die Anfertigung dieser Arbeit herangezogen wurde.

## II. Routing In WSNs: Standard-Protokolle und Routing Protocol for Low Power and Lossy Networks (RPL)

### A. Standard-Protokolle

In mobilen ad-hoc Netzwerken können verschiedene Typen von Routing-Verfahren verwendet werden. Sie werden im Folgenden näher betrachtet. Detailliertere Informationen finden sich in [4]. In diesem Paper wird insbesondere auf RPL eingegangen.

#### 1) Proaktive Routing Protokolle

Proaktive Routing Protokolle wie beispielweise OLSR ermitteln Pfade zwischen Knoten schon bevor Nutzdaten gesendet werden. Dies geschieht durch den regelmäßigen Austausch von Kontrollnachrichten. Die Informationen werden von jedem Knoten selbst verwaltet und in Tabellen gespeichert. Der Vorteil dieses Verfahrens ist, dass Pfade bereits vorher bestimmt sind und somit beim Senden von Nutzdaten nicht auf eine Berechnung des Pfades gewartet werden muss. Für Netzwerke, bei denen geringer Stromverbrauch an den Endgeräten und hohe Skalierbarkeit Priorität haben, sind solche Protokolle eher ungeeignet, denn das Updaten der Tabellen braucht viel Netzwerkbandbreite und Strom an den jeweiligen Endgeräten.

#### 2) Reaktive Routing Protokolle

Bei reaktiven Routing-Protokollen wird der Pfad erst bestimmt, wenn Nutzdaten übertragen werden sollen. Die Bestimmung der Route erfolgt meist über ein Routing-Request-Paket, das in das Netzwerk geflooded wird. Wenn dieses Paket einen Knoten mit einer Verbindung zum Zielknoten oder den Zielknoten selbst erreicht, wird ein Route-reply zurückgesendet. Ein Beispiel für ein reaktives Routing Protokoll ist AODV.

Es werden 2 Kategorien von reaktiven Routing-Protokollen unterschieden: Source Routing und Hop by Hop Routing. Beim Source Routing enthält jedes Paket die komplette Route von Source bis Zielknoten. In großen Netzwerken kann dies zu Problem führen. Zum einen steigt die Wahrscheinlichkeit eines Fehlers mit der Anzahl der Knoten, zum anderen wird bei langen Routen der Overhead im Header immer größer. Zur Verringerung des Overheads wurde in [2] mit Secure and Efficient Flooding (SEF) ein Ansatz vorgestellt, der den Overhead verringern kann.

Beim Hop by Hop Routing kennt jeder Knoten nur den Zielknoten und den nächsten Hop. Dadurch wird der Overhead geringgehalten und Routen können in dynamischen Netzwerken performanter berechnet werden. Jedoch muss dadurch jeder Knoten seine aktiven Nachbarn bzw. Routen kennen.

### *B. RPL*

Das Routing Protocol für Low Power and Lossy Networks (RPL) wurde speziell für WSNs entwickelt, die aufgrund ihres Anwendungsgebietes Limitationen aufweisen [1]. Oft ist die Kommunikation der Knoten aufgrund äußerlicher Einflüsse schwierig und auch die Stromversorgung ist meist begrenzt. Deshalb ist es wichtig, dass das Protokoll energiesparend und trotzdem effizient arbeitet.

RPL erstellt als Erstes einen Destination-Oriented Directed Acyclic Graph (DODAG, Abbildung 1). Dieser besteht aus einem Zielknoten (DODAG root) an dem alle Nachrichten ankommen sollen, beliebig vielen Zwischenknoten und beliebig vielen Endknoten. Der Zielknoten ist leistungsstärker und gibt die Nachrichten über einen Uplink zur weiteren Verarbeitung weiter. Um eine Nachricht an den Zielknoten zu senden, wird sie von jedem Knoten an seinen jeweiligen Elternknoten geschickt, bis der Zielknoten erreicht ist. Die Elternknoten werden von jedem Knoten durch die Objective Function (OF) für sich selbst bestimmt. Dabei fließen Parameter wie Energieverbrauch, Latenzzeit und Verbindungsqualität mit ein. An welcher Stelle im DODAG ein Knoten steht, hängt vom Rang eines Knoten ab. So muss ein Elternknoten immer einen kleineren Rang haben als seine Kindsknoten. Der Zielknoten muss immer den Rang 0 haben. Die OF berechnet den Rang eines Knotens. In der Regel gilt: Je höher der Rang eines Knotens ist, desto weiter unten im DODAG befindet er sich, jedoch wird nicht immer der Knoten mit dem niedrigsten Rang als Elternknoten gewählt. Die Knoten kommunizieren in bestimmten Zeitintervallen ihren Rang untereinander über ein DODAG Information Object (DIO). Dies gewährleistet, immer einen dynamischen und anwendungsspezifischen Graphen zu haben.

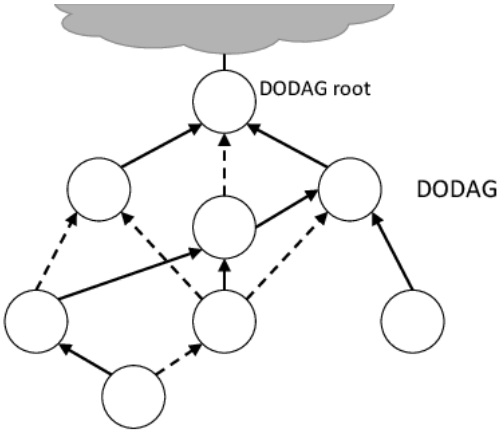


Abbildung 1: Destination-Oriented Directed Acyclic Graph (DODAG): durchgezogene Pfeile für bevorzugte Elternknoten, gestrichelte Pfeile für andere mögliche Routen. Quelle: [3].

### *C. RPL Erweiterung Trust and Forgiveness*

Obwohl das Standard RPL mit fehlerhaften Knoten umgehen kann, stellen Knoten, die nur teilweise ausfallen eine Herausforderung dar. Der Grund dafür ist meist fehlerhafte Software, die auf Knoten installiert ist oder ein Overload auf dem Bus zum Wireless Adapter. Um diese Probleme möglichst effizient zu lösen wurde die Erweiterung Trust and Forgiveness für das Standard RPL eingeführt. Zunächst wird sicherstellt, dass der Zielknoten weiß, ob er alle Pakete erhalten hat. Hierfür wird eine Sequenznummer, die mit jeder Trust Round wieder auf 0 gesetzt wird, an jedes Paket angefügt, das zum Zielknoten gesendet wird. Trust Rounds sind notwendig, da Sensorknoten meist keine oder nur schlecht synchronisierte Uhren haben. Mit jedem neuen DIO vom Zielknoten wird eine neue Trust Round gestartet. Zu Beginn jeder Trust Round prüft jeder Knoten seine Elternknoten und passt diese an, falls es bessere gibt. Durch die Sequenznummern kann nachverfolgt werden, ob der Zielknoten alle Pakete erhalten hat. Um zu gewährleisten, dass auch das letzte Paket angekommen ist und auch der Zielknoten sowie der sendende Knoten diese Information hat, wird das DIO, das periodisch vom Zielknoten gesendet wird, modifiziert. Diese Information wird dem Header des DIOs zusammen mit einem trustinfo-container in einem metric-container hinzugefügt. Im trustinfo-container ist die Anzahl der erhaltenen Pakete eines bestimmten Knotens und dessen ID enthalten.

Durch die Sequenznummer und Trust Rounds kann jeder Knoten die Zustellrate auslesen. Mit einer Trust Metric werden Trust-Werte der Nachbarn ermittelt. Zunächst sind alle Knoten neutral. Der Wert wird beeinflusst durch Zustellrate, vorherigen Trust-Wert und einen Faktor, der die Gewichtung vorheriger Werte bestimmt. Abhängig davon berechnet die OF bei jeder neuen Trust Round den am besten geeigneten Elternknoten.

Dieses Verfahren bringt jedoch mehr Overhead mit sich und kann so die Effizienz des Netzwerkes und den Stromverbrauch der Knoten negativ beeinflussen. Um den Overhead so gering wie möglich zu halten, wurde Secure and Efficient Flooding(SEF) entwickelt [2].

## III. MOBILE AD-HOC NETZWERKE UND WORMHOLE ATTACKS

Wie jedes andere Netzwerk, sind auch WSNs Gefahren und Angriffen von außen ausgesetzt. Daher ist es wichtig, dass Protokolle erkennen können, wenn ein oder mehrere Knoten sich „falsch“ verhalten und entsprechend zeitnah darauf reagieren können.

### A. *Mobiles ad-hoc Netzwerk*

Als mobiles ad-hoc Netzwerk lässt sich prinzipiell jedes Netzwerk einordnen, das aus beliebig vielen Knoten besteht und sich selbst aufbauen und konfigurieren kann. Das heißt, eine vorher existierende Infrastruktur aus Routern oder Access Points ist nicht notwendig, da jeder einzelne Knoten direkt mit anderen Knoten kommuniziert. Dadurch sind ad-hoc Netzwerke sehr dynamisch und Knoten können jederzeit hinzugefügt bzw. entfernt werden. Es gibt diverse Untergruppen von mobilen ad-hoc Netzwerken, die auf das jeweilige Anwendungsgebiet spezialisiert sind. Dazu gehören beispielsweise vehicular ad-hoc Netzwerke oder Wireless Sensor Networks. Genauere Informationen über ad-hoc networking finden sich in [9].

Ein Organic Computing System ist in [12] definiert als technisches System vieler Sensoren und Knoten, die Wissen über sich selbst und ihre Umgebung sammeln und sich zur Laufzeit anpassen können. Das RPL ist grundlegender Bestandteil solcher Sensornetze und sieht self-awareness als Voraussetzung an [13]. Die Knoten sammeln zur Laufzeit Informationen über sich, ihre Umgebung und ihre Nachbarn und tauschen sie untereinander aus.

### B. *Wormhole Attack*

Für eine Wormhole-Attack sind zwei oder mehr angreifende Knoten notwendig, die in das Netzwerk eingebunden werden. Diese sind durch einen sogenannten Wormhole-Link miteinander verbunden. Der Wormhole-Link ist eine schnelle, direkte Verbindung und weist somit geringe Latenz auf [5]. Dadurch werden Routing-Protokolle durcheinandergebracht: Knoten, die eigentlich mehrere Hops voneinander entfernt sind, können nun über die angreifenden Knoten schneller kommunizieren. Dadurch wird den Knoten suggeriert, dass sie nicht weit auseinander sind und Pakete werden vorzugsweise über den schnellen Wormhole-Link geroutet. Genau das ist das Ziel eines solchen Angriffs. Die Angreifer können nun selektiv Pakete nicht übermitteln und die Kommunikation der Knoten, sowie die gesamte Performance des Netzwerks stark beeinflussen. Insbesondere für unverschlüsselte Protokolle stellen Wormhole-Attacks in Kombination mit einem Sniffing-Angriff, bei dem Daten abgehört werden, eine große Gefahr dar. Denn sehr viele Daten werden durch den Wormhole-Link, an dem der Sniffer ansetzt, gerouted. Eine Implementierung mit Beispielen findet sich in [3]. In Abbildung 2 sind Knoten X und Y die angreifenden Knoten. Den Wormhole Link zwischen ihnen stellt die gestrichelte Linie dar. Pakete, die von A nach B gerouted werden sollen, werden nun über den Wormhole-Link gerouted, anstatt über eigentlich mehrere Hops. Die Knoten A und B sehen sich also als Nachbarn an und routen daher die Pakete bevorzugt über den Wormhole Link.

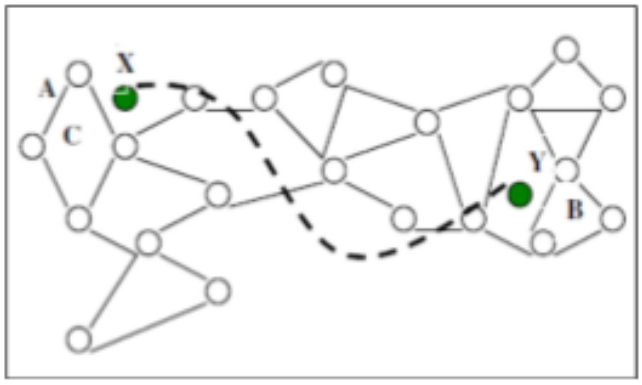


Abbildung 2: Wormhole-Link in Netzwerk. Quelle [8].

## IV. WIE GEHT RPL MIT DIESEN RISIKEN UM

Standard RPL benutzt meist den Expected Transmission Count (ETX) als Metrik zur Bestimmung der besten Route. Dieser wird unter den Knoten in den DODAG Information Objects ausgetauscht. Hierbei wird jedoch nur die Anzahl wirklich gesendeter Pakete berücksichtigt, was zu Folge hat, dass das Routing nicht garantiert werden kann. Angreifende Knoten, die einen ETX Wert fälschen oder durch einen Wormhole-Link den Graphen manipulieren, können daher kaum effizient bekämpft werden. Im Folgenden wird erläutert, wie RPL mit Wormhole-Attacks umgeht. Einmal ohne die Erweiterung „Trust and Forgiveness“ und einmal mit.

### A. *Wormhole*

#### 1) *ohne Trust and Forgivness*

Ein Wormhole-Link kann durch das RPL Protokoll nicht erkannt werden, solange die Pakete normal übertragen werden. Zunächst ist dies vom Angreifer so gewollt (warm-up Phase). Durch die geringe Latenz des Wormhole-Links werden die Routen vorzugsweise über diesen gewählt und so die Struktur des DODAGs verändert. Erst wenn im nächsten Schritt Pakete verzögert oder gar nicht weitergeleitet werden, kann RPL auf die angreifenden Knoten reagieren. Dies wird in [10] genauer beschrieben. Desweiteren wird hier als beste Gegenmaßnahme für Wormhole-Attacks angegeben, dass suksessive Angriffe besser bekämpft werden müssen. Denn meist wird eine Wormhole Attack mit dem Ziel durchgeführt, auf dem Wormhole-Link, auf den viel Traffic gezogen wird, einen weiteren Angriff durchzuführen. Dazu gehören beispielsweise Abhören oder das selektive Nicht-Übertragen von Paketen. Diese Angriffe können weitaus besser erkannt und bekämpft werden.

Dies wird in Standard-RPL ohne der Erweiterung Trust and Forgiveness folgendermaßen umgesetzt: Die Knoten, die den Wormhole Link bereitstellen werden als fehlerhafte Knoten angesehen, wenn sie den suksessiven Angriff ausführen. Wenn also Pakete gedroppt werden, wird dies von RPL erkannt und die Knoten reevaluieren ihre Eltern-Knoten. Jedoch können in dieser Zeit keine Pakete übertragen werden und die vorher gesendeten Pakete kommen folglich auch nicht an. Es werden erst wieder Pakete normal übertragen, wenn neue DIOs an den betroffenen Knoten ankommen und sie neue Elternknoten

wählen. Dieser Vorgang funktioniert, ist aber nicht sehr effizient und bringt viel Paketverlust mit sich.

*2) mit Trust and Forgiveness*

Die Erweiterung Trust and Forgiveness geht das Problem der schnellen Recovery bei Angriffen an. Durch die in Kapitel 2C beschriebene Funktionsweise, kann der Paketverlust während eines Angriffs um einiges geringer gehalten werden. Durch die Implementierung der Second-Chance wird nicht nur ein Elternknoten als einzige Route betrachtet. Gibt es mehrere Kntoen mit einem guten ETX-Wert, werden einige Pakete über diese versendet. Wird dann eine Packet Delivery Rate von 100% erreicht, wird dieser Knoten mit einer guten Bewertung versehen und kann gegebenfalls wieder als Elternknoten verwendet werden. So können Knoten schon andere gute Routen erkennen, auch wenn der aktuell ausgewählte Elternknoten eigentlich noch gut genug ist. Im Falle eines Angriffs können so effizient andere Routen gewählt werden und der Paketverlust klein gehalten werden. Außerdem besteht die Möglichkeit, dass Knoten, die fehlerhaft waren, also beispielsweise Knoten, zwischen denen ein Wormhole-Link war, wieder in das System eingegliedert werden können, nachdem der Wormhole Link eliminiert wurde und die betroffenen Knoten wieder normal senden.

## V. Ergebnis und Bewertung

Die betroffenen Knoten einer Wormhole-Attack können als Knoten mit Fehlfunktion betrachtet werden. Denn in Kombination mit anderen Angriffen werden durch die Knoten Pakete nicht zugestellt, bzw. abgefangen. Daher lässt sich die Funktionsweise wie im in [2] beschriebenen Experiment darstellen. Hierbei wurde eine Simulation durchgeführt. Betrachtet man also ein Szenario wie in Abbildung 3 kommt man zum Ergebnis in Abbildung 4.

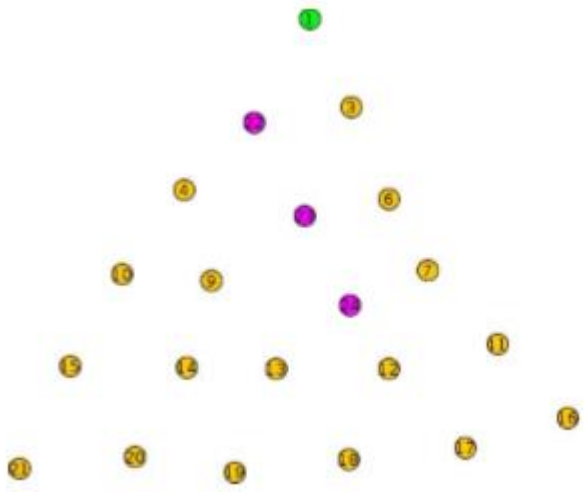

Abbildung 3: Verteilung der Knoten im Experiment: Die angreifenden Knoten sind violett eingefärbt, der Zielknoten grün. Die restlichen, gelben, Knoten senden normal. Quelle: [2].

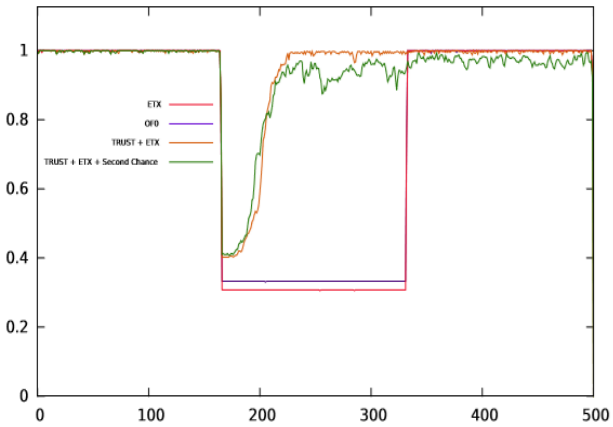


Abbildung 4: Ergebnis der Tests in [2]. Bei Sequenz 160 bis 340 wurde ein Angriff gestartet. Quelle: [2].

Betrachtet man die rote Kurve (Standard-RPL), sieht man, dass zur Zeit des Angriffs die Zustellrate der Pakete auf ca. 30% abfällt. Erst am Ende des Angriffs wird wieder 100% Zustellrate erreicht. Die orange (RPL mit Trust) und grüne Kurve (RPL mit Trust und Second Chance) erholen sich deutlich schneller und erreichen schon während des Angriffs wieder eine Zustellrate >95%. Die Zustellrate von RPL mit Trust und Second Chance bleibt nach Erholung vom Angriff etwas geringer als die ohne Second Chance, weil schlechte Routen periodisch wieder ausgetestet werden und so eine Wiedereingliederung der Knoten in das System zur Laufzeit ermöglicht werden kann. Befindet man sich also in einem System, auf das viele Angriffe ausgeführt werden, ist RPL mit Trust die performantere Wahl, da einer einmal schlechten Route nie wieder vertraut wird. In selten angegriffenen Systemen lohnt sich Second Chance aber schon, da zuvor schlechte Routen wiederverwertet werden können und so ein stabileres System gewährleistet wird.

## VI. Ausblick

Sicherheit spielt in jedem Netzwerk eine wichtige Rolle. In weiteren Arbeiten könnte noch auf andere Angriffsmethoden genauer eingegangen werden und wie RPL mit diesen umgeht. Außerdem ist eine weitere, interessante Fragestellung, ob und wie sich eine Erweiterung wie Trust and Forgiveness auf andere Routing Protokolle auswirkt.

## Referenzen


[1] Jan Kantert, Christian Ringwald, Georg von Zengen, Sven Tomforde, Lars Wolf and Christian Müller-Schloer, Enhancing RPL for Robust and Efficient Routing in Challenging Environments, 2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems Workshops

[2] Jan Kantert, Sven Tomforde, Georg von Zengen, Susanne Weber, Lars Wolf, and Christian Müller-Schloer, Improving Reliability and Reducing Overhead in Low-Power Sensor Networks using Trust and Forgiveness, 2016 IEEE International Conference on Autonomic Computing

[3] Pericle Perazzo, Carlo Vallati, Dario Varano, Giuseppe Anastasi and Gianluca Dini, Implementation of a Wormhole Attack Against a RPL Network: Challenges and Effects, ISBN 978-3-903176-02-7 © 2018 IFIP

[4] Abolhasan, M., Wysocki, T. & Dutkiewicz, E. (2004). A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks, 2 (1), 1-22.

[5] Furrakh Shahzad, Maruf Pasha, Arslan Ahmad, A Survey of Active Attacks on Wireless Sensor Networks and their Countermeasures, International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 12, December 2016

[6] Frank Stajano, Dan Cvrcek, and Matt Lewis, Steel, Cast Iron and Concrete: Security Engineering for Real World Wireless Sensor Networks, S.M. Bellovin et al. (Eds.): ACNS 2008, LNCS 5037, pp. 460–478, 2008. c Springer-Verlag Berlin Heidelberg 2008.

[7] TEODOR-GRIGORE LUPU, Main Types of Attacks in Wireless Sensor Networks, ISSN: 1790-5109, ISBN: 978-960-474-114-4.

[8] RAJ SHREE and R. A. KHAN, Wormhole Attack in Wireless Sensor Network, VOL.2, NO.1, JANUARY 2014, 22–26 Available online at: www.ijcncs.org ISSN 2308-9830

[9] Perkins, Charles E. *Ad hoc networking*. Vol. 1. Reading: Addison-wesley, 2001.

[10] Granjal, Jorge, Edmundo Monteiro, and Jorge Sá Silva. "Security for the internet of things: a survey of existing protocols and open research issues." *IEEE Communications Surveys & Tutorials* 17.3 (2015): 1294-1312.

[11] Gubbi, Jayavardhana, et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." *Future generation computer systems* 29.7 (2013): 1645-1660.

[12] Müller-Schloer, Christian, and Sven Tomforde. *Organic Computing-Technical Systems for Survival in the Real World*. Springer International Publishing, 2017.

[13] Tomforde S. et al. (2011) Observation and Control of Organic Systems. In: Müller-Schloer C., Schmeck H., Ungerer T. (eds) Organic Computing — A Paradigm Shift for Complex Systems. Autonomic Systems, vol 1. Springer, Basel