An Implementation and Evaluation of the Security Features of RPL

Pericle Perazzo, Carlo Vallati, Antonio Arena, Giuseppe Anastasi, and Gianluca Dini

University of Pisa, Department of Information Engineering, Pisa 56122, Italy,
[name.surname]@iet.unipi.it,
WWW home page: http://www.dii.unipi.it

Abstract. Wireless Sensor and Actuator Networks (WSANs) will represent a key building block for the future Internet of Things, as a cheap and easily-deployable technology to connect smart devices on a large scale. In WSAN implementation, the Routing Protocol for Low-Power and Lossy Networks (RPL) has a crucial role as the standard IPv6-based routing protocol. The RPL specifications define a basic set of security features based on cryptography. Without these features, RPL would be vulnerable to simple yet disruptive routing attacks based on forgery of routing control messages. However, the impact of these features on the performances of the WSAN has not been investigated yet. The contribution of this paper is twofold: an implementation of the RPL security features for the Contiki operating system, which is, at the best of authors' knowledge, the first available, and an evaluation of their impact on the WSAN performances by means of simulations. We show that the protection against eavesdropping and forgery attacks has a modest impact on the performances, whereas the protection against replay attacks has a more considerable impact, especially on the network formation time which increases noticeably.

Keywords: Internet of Things, embedded systems, secure routing, RPL

1 Introduction

Recent technology advancements are rapidly making real the Internet of Things (IoT), a future in which objects will be empowered with communication capabilities to enable seamless integration with information systems. In this future, such smart objects will penetrate the physical world around us, in some cases implementing remote monitoring and control capabilities, in others, offering enhanced features that exploit automation and self-coordination. IoT applications are expected to cover a wide range of domains such as smart home, smart city, e-health, and so on.

Wireless Sensor and Actuator Networks (WSANs) will be a key enabler for all these IoT applications, because they allow for rapid and cost-effective installation of smart objects over large areas. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [16], standardized in 2012 by the IETF ROLL

working group, is considered at the present the most mature option to connect IPv6-enabled devices and form WSANs over lossy links with minimal overhead [5]. Considering the importance of the services delivered, protecting the routing functionalities from attacks will be a major challenge to prevent malicious attempts to disrupt IoT network operations [10]. A basic set of cryptographic security mechanisms to guarantee routing resilience to external attackers has been introduced by design in RPL specifications [16]. However, the impact of such mechanisms on the WSAN performances has not been investigated yet.

In this paper, we give a first evaluation of the impact of the RPL security mechanisms on the WSAN performances. We develop a standard-compliant implementation of the RPL security mechanisms for the Contiki operating system [3], and we evaluate it by means of simulations. We show that the RPL security mechanisms have a negligible impact on the performances in terms of network formation time and power consumption, if they do not have to defend against replay-based attacks. Otherwise, if also replay protection is needed, the impact on performances is more pronounced, especially on the network formation time which increases noticeably. To the best of our knowledge, this is the first implementation of the standard security mechanisms of RPL.

The remainder of the paper is organized as follows. In Section 2 we review related work. In Section 3 we offer a short introduction to the RPL specifications, including its security mechanisms. In Section 4 we describe our implementation of the security mechanisms of RPL. In Section 5 we evaluate the impact of security in RPL performances. Finally, Section 6 concludes the paper.

2 Related Work

Many research papers [1, 4, 6, 8, 12, 14] studied possible attacks against RPL, and proposed countermeasures. Dvir et al. [4] took into consideration Rank attack and DODAG Version attack. Both these attacks can be considered as RPL-specific instances of the more general sinkhole attack [7], in which a malicious node attracts a large amount of traffic from surrounding nodes in order to eavesdrop or interrupt it. The authors presented a countermeasure to both attacks based on asymmetric cryptography. Perrey et al. [12] presented an improvement of such countermeasure which corrects some of its vulnerabilities, but requires round-trip protocols for path validation. Weekly and Pister [14] presented and evaluated the synergy between two countermeasures against sinkhole attacks in RPL: parent fail-over and rank authentication. Iuchi et al. [6] presented a countermeasure against Rank attack based on a particular next-hop selection policy. This countermeasure requires the nodes to choose sub-optimal routes. Le et al. [8] studied the impact of an attack in which a malicious node deviates from the normal behavior by selecting as next hop the worst neighbor instead of the best one. Airehrour et al. [1] proposed a countermeasure against blackhole attack, in which a malicious node drops all the traffic forwarded to it, and breaks the availability of large parts of the network. Their countermeasure requires every

node to operate in promiscuous mode, and to receive and process also packets not destined to it.

All these countermeasures provide for partial security, since they defend against specific attacks only, namely Rank attacks [4, 6, 12, 14], DODAG Version attacks [4, 12], blackhole attacks [1], and attacks involving next-hop selection [8]. All these attacks can be avoided, at least in case of external adversaries, by simply using the standard RPL security mechanisms. Indeed, they impede a malicious entity to become part of the network and transmit routing control messages. In this paper, we evaluate the impact of RPL security mechanisms on the WSAN performances.

3 IPv6 Routing Protocol for Low-Power and Lossy Networks

The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [5, 16] is an IPv6 distance-vector routing protocol focused on resource-constrained devices and lossy wireless environments. RPL assumes that the majority of the traffic is upstream, i.e., directed towards a single node acting as a border router. Downstream traffic, i.e., generated by the border router and directed towards other nodes, is considered to be sporadic, and node-to-node traffic to be rare. For this reason, RPL builds and maintains a logical topology for upstream data delivery, and downstream routes are established only when required. Specifically, the built topology is a *Destination Oriented Directed Acyclic Graph* (DODAG), in which every node has a set of neighbors (*parent set*), which are candidates for upstream data delivery. Among the nodes in the parent set, one node is selected as the *preferred parent*. The preferred parent is the node exploited for upstream data forwarding, whereas the other parents are kept as failover.

The DODAG is rooted in a single node, called *DODAG root*, to which all upstream data is directed. The DODAG root is also a border router for the other nodes. It is responsible for triggering the network formation through the emission of *DODAG Information Object* (DIO) messages. Initially, every non-root node listens for DIO messages. When a DIO is received, the node joins the network using the information included in the message. Right after having joined the network, the node starts emitting DIOs to advertise its presence and its distance to the root. During regular operations, the emission of DIO messages is regulated by the *Trickle* algorithm [9], which aims at reducing the power consumption of the nodes by minimizing the redundant messages and by adapting dynamically the transmission rate over time. The asynchronous emission of DIOs can be requested through *DODAG Information Solicitation* (DIS) messages, e.g., to accelerate the join process of a node during network formation or to recover from errors during regular network operations. All the RPL messages (DIO, DIS, etc.) are ICMPv6 messages of Type 155.

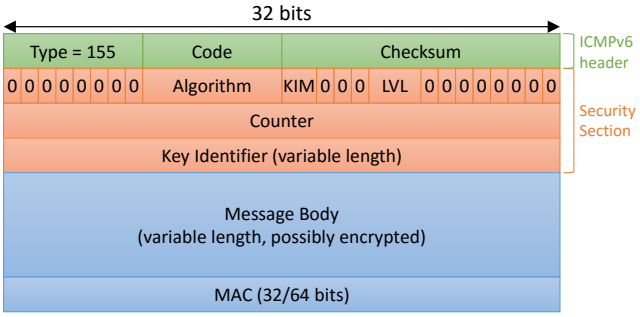

Fig. 1. Secured RPL message format.

3.1 Security Mechanisms

A DODAG can operate in one of the following security modes: unsecured mode, preinstalled mode, or authenticated mode. In the unsecured mode, the RPL messages are sent in the clear and without any security protection. In the preinstalled mode, the RPL messages are protected by cryptography-based security mechanisms using keys assumed to be already present in each node at boot time. In the authenticated mode, the RPL messages are protected in the same way, but the nodes receive keys from some key authority after having underwent an authentication process. The preinstalled and the authenticated modes differ only in the way the keys are deployed on the nodes, and they have in common all the other security mechanisms. In the scope of the present paper, we will refer to both preinstalled and authenticated modes with the general term "secured modes".

The RPL specifications define the following security services: (a) data confidentiality, (b) data authenticity, (c) replay protection. Data confidentiality assures that routing information is not disclosed to unauthorized entities. Data authenticity assures that routing control messages are not modified in transit. Replay protection assures that malicious duplicates of routing control messages are discarded.

If the DODAG operates in a secured mode, all the RPL messages are *secured*. A secured RPL message follow the general format shown in Fig. 1, in which the message body is preceded by a *Security Section*. The Code field of the ICMPv6 header determines the type of the RPL message: secured DIO, secured DIS, etc. The Algorithm field of the Security Section specifies the algorithm suite employed to authenticate and encrypt the message. With the current version of the specifications, only CCM (CTR with CBC-MAC) with AES-128 [15] is supported. CCM is a mode of operation for 128-bit block ciphers which can provide for both confidentiality and authenticity by combining the CTR (Counter) encryption mode of operation and CBC-MAC (Cipher Block Chaining Message Authentication Code). The LVL bits (Security Level) specify whether the message is only authenticated or both authenticated and encrypted, and the length

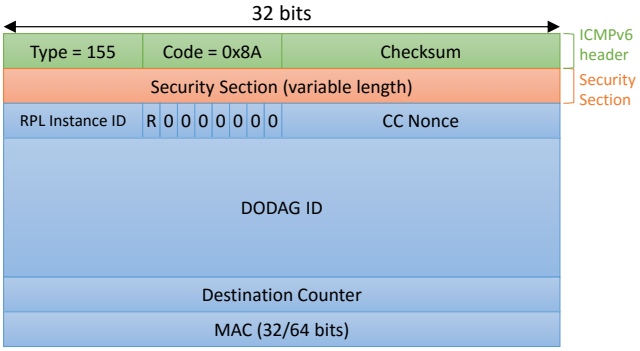

Fig. 2. Consistency Check (CC) message format.

of the MAC field. The Key Identifier field, whose format is specified by the KIM bits (Key Identifier Mode), identifies the employed cryptographic key. The Counter field is a value incremented at each sent RPL message. It is used both as input of CCM and to implement replay protection mechanisms.

To recover from situations in which a node loses its current Counter value, for example after a reboot, the RPL specifications foresee a *Consistency Check* (CC) message. The format of such a message is shown in Fig. 2. The CC message is used to inform a destination about the last valid value of its Counter field, and to issue generic challenge-response handshakes. The Destination Counter field contains the last valid value of the Counter field of the target node. The CC Nonce field is used as a proof of freshness within challenge-response handshakes, and the R bit specifies whether the message is a challenge (CC request) or a response (CC response). The RPL Instance ID and the DODAG ID fields identify the DODAG to which the sender node belongs to, in the case that multiple RPL instances or multiple DODAGs in the same RPL instance are present.

4 Our Implementation

Our standard-compliant Contiki implementation of the RPL security mechanisms is available from a public repository[1]. It extends the standard module ContikiRPL [13] and it has two possible configurations: a *light-security configuration* which implements only data confidentiality and data authenticity services; and a *full-security configuration* which also implements replay protection service.

Both configurations use a network-wide cryptographic key assumed to be already present in each node at boot time. This realizes the preinstalled security mode foreseen by the RPL specifications [16], which is of course vulnerable to key stealing through node compromise. However, the presented implementation is adaptable to the authenticated security mode, in which the nodes receive keys from some key authority. The authenticated mode will lose some performance

[1] https://github.com/arenantonio92/contiki.

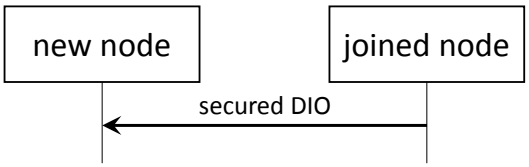

Fig. 3. Join procedure in light-security configuration. The solid arrow represents a multicast message.

with respect to the preinstalled one, since nodes have to undergo also an authentication process when they join the network. Therefore, the performances presented in this paper represent also the best-case performances for the authenticated mode.

4.1 Threat Model

The light-security configuration defends against an adversary which tries to eavesdrop legitimate RPL messages to infer the topology, or forge malicious RPL messages to become part of the network and act as an internal adversary. This models a wide range of simple yet disruptive routing attacks [1, 4, 6, 8, 12, 14].

The full-security configuration defends against an adversary which also tries to replay legitimate RPL messages to modify the topology. For example she can replay a legitimate DIO message originally sent by the root in a zone of the network where the root is not directly reachable. The victim nodes receiving such replayed message could think that they have a direct link with the root and could forward their upstream data along such link. Since the link does not actually exist, all the upstream communication of the victim nodes will be broken.

4.2 Light-Security Configuration

The light-security configuration simply includes a security section on each RPL message, which provides for integrity with a Message Authentication Code (MAC), and confidentiality with encryption. With such a configuration, the procedure by which a node acquires the first preferred parent (*join procedure*) is similar to that of the unsecured mode, except that DIO messages are secured. The join operation follows the sequence diagram shown in Fig. 3. The joined node sends a multicast secured DIO, which is received by the new node. The new node checks the validity of the MAC of the secured DIO, and possibly decrypts it. With the information carried by the DIO, the new node can choose the joined node as parent. This concludes the join procedure. The emission of secured DIO messages by the joined node is recurrent and regulated by the Trickle algorithm [9]. The asynchronous emission of secured DIOs can be requested through secured DIS messages.

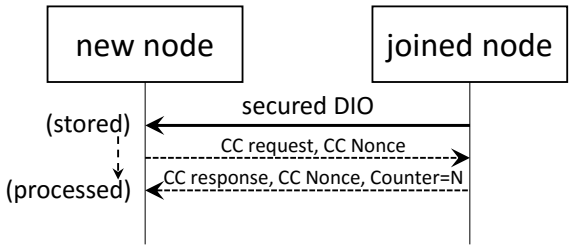

Fig. 4. Join procedure in full-security configuration. The dotted arrows represent unicast messages.

4.3 Full-Security Configuration

The full-security configuration provides for integrity and confidentiality with encryption, in the same way as the light-security configuration. In addition, it provides for replay protection by checking the Counter field of the Security Section of the incoming messages. Every node maintains a *counter watermark* for each neighbor, containing the highest Counter field received from that neighbor. Upon receiving a new message from that neighbor, if its Counter field is less than or equal to the counter watermark, then the message will be discarded. Otherwise, the message will be accepted and the counter watermark updated.

When a new node wants to join the DODAG, it does not have a counter watermark for the other nodes. Therefore, it cannot assess whether a DIO message received from them is a replay of an old message. In order to initialize the counter watermarks on the new node for the nodes already joined, we use CC requests and CC responses. The join operation follows the sequence diagram shown in Fig. 4. The joined node sends a multicast secured DIO, which is received by the new node. The new node checks the validity of the MAC of the secured DIO, and possibly decrypts it. Then, it stores the DIO into memory without processing it, waiting to assess whether it is a replay of an old message or not. The new node then initiates a CC challenge-response handshake with the joined node. In such a handshake, the new node sends a CC request carrying a fresh CC Nonce to the joined node, which answers with a CC response carrying back the same CC Nonce. The CC response also carries the current Counter N of the joined node. The fresh CC Nonce, together with the MAC protecting the whole CC response, assures the new node of the freshness of N and, retroactively, that the stored DIO message was not a replay. Upon having received the CC response, the new node checks the validity of its MAC, and possibly decrypts it. Then, the new node checks that the CC Nonce is the same of that transmitted in the CC request. Now the new node can initialize a counter watermark for the joined node to N. Finally, the new node checks if the stored DIO had a Counter field of $N - 1$, i.e., immediately precedent of the Counter field of the CC response. If it has, the DIO was not a replay, so it can be processed and the new node can choose the joined node as its parent. This concludes the join procedure.

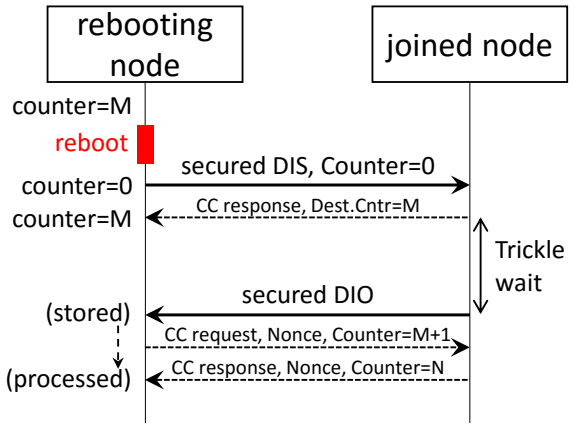

Fig. 5. Rejoin procedure in the full-security configuration.

Note that, from now on, the new node has a counter watermark for its parent. This means that it can process future secured DIOs from such parent without repeating CC challenge-response handshakes.

In a low-power wireless network it can happen that a node reboots, as result of hardware failure, software bug or simply battery shortage. In this case, the node loses its current Counter value, and its transmitted messages will start again from a zero Counter field. As a consequence, the messages will be discarded by the neighbors as possible replays. To recover from this situation, we use a *rejoin procedure*, which follows the sequence diagram shown in Fig. 5. After rebooting, the node starts sending multicast secured DIS messages with a special Counter field of zero. If some neighbor is maintaining the value M of its past Counter and receives a secured DIS, then it sends a CC response to the rebooted node carrying a Destination Counter field of M. Then, the rest of the procedure is similar to the join procedure (see Fig. 4). After a wait imposed by the Trickle algorithm, the joined node sends a multicast secured DIO message. The rebooted node stores the DIO into memory without processing it, and initiates a CC challenge-response handshake with the joined node. After that, the rebooted node can initialize a counter watermark for the joined node to the Counter value N carried by the CC response. Finally, the new node checks if the stored DIO had a Counter field of $N - 1$. If it has, the DIO was not a replay, so it can be processed and the rebooted node can choose the joined node as its parent. This concludes the rejoin procedure.

5 Impact of Security on Performances

In order to evaluate the overhead introduced by the RPL security mechanisms, a performance evaluation based on simulations has been run. To this aim, three

different RPL configurations are considered: the unsecured mode, which corresponds to the "vanilla" Contiki RPL implementation [13]; the preinstalled mode with light-security configuration; and the preinstalled mode with full-security configuration.

Simulations have been run exploiting COOJA [11], a network emulator which is available as part of the Contiki distribution. COOJA emulator provides a realistic simulation environment in which wireless nodes are emulated allowing to run the same binary image that would be executed on real nodes. In our simulations, COOJA has been configured to emulate the Zolertia-Z1 sensor mote[2], an MSP430-based board with an IEEE 802.15.4-compatible CC2420 radio chip. Wireless channel is simulated using the Unit Disk Graph Medium model, which implements a disk reception model. The reception/transmission range is set to 50m and the interference range is set to 100m.

In order to assess the performance of the RPL protocol in networks of different sizes, a regular grid topology with an increasing number of nodes is considered. Specifically, a 2x2 grid of 4 nodes, a 3x3 grid of 9 nodes, a 4x4 grid of 16 nodes, and a 5x5 grid of 25 nodes are considered. The distance between the nodes is fixed to 30m in all the topologies. The node at the top-right corner is configured to behave as DODAG root. All the RPL settings are configured according to the Contiki default parameters. The radio duty cycling algorithm adopted by each node is the ContikiMAC one.

5.1 Impact on Network Formation

The additional complexity introduced by the security on the join operation influences mainly the initial network formation. The impact of the RPL security mechanisms on the network formation operation is assessed through the following metrics:

- *Network formation time*, defined as the time between the beginning of the simulation and the moment in which the last node joins the DODAG. This metric measures the time required by the network to become fully operational.
- *Power consumption*, defined as the average value of the power consumption of each node in the network. This metric includes both communication costs (for transmission and reception of messages) and computation costs (for cryptographic operations and message processing). The power consumption is evaluated using the Powertrace tool [2] and the nominal power consumption values reported in the Z1 datasheet.
- *RPL message overhead*, defined as the overall message size in bytes of all the RPL messages sent over the simulation by all the nodes in the network. This metric is adopted to assess the additional overhead in message size introduced by the security mechanisms.

[2] Zolertia Z1 website: http://zolertia.io/z1

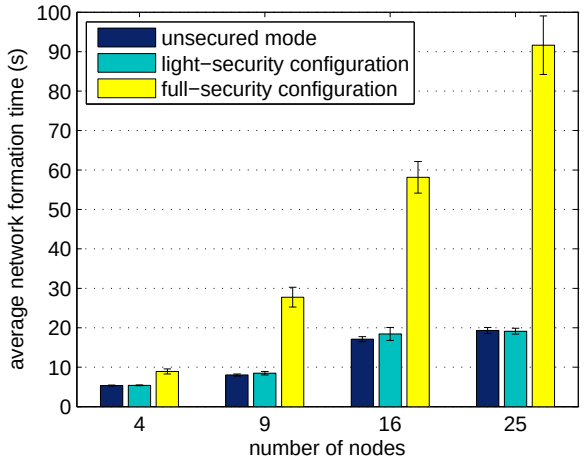

Fig. 6. Average network formation time. 95%-confidence intervals are displayed in error bars.

Simulations are run for 60 minutes. In order to obtain statistically sound results, 32 independent replications with different seeds are run for each scenario. The average value of each metric with its 95%-confidence interval is reported.

In Fig. 6 the average network formation time is reported. As expected, the network formation time increases with the network size. Regardless of the network size, just the usage of encrypted and authenticated RPL messages (light-security configuration) does not influence the formation time of the network. Instead, the introduction of the replay protection mechanism (full-security configuration) increases noticeably the network formation time. The overhead introduced by the replay protection increases with the network size. This can be explained considering the additional exchange of messages required before each node can join the DODAG. The network formation time with the full-security configuration can be probably reduced by employing algorithms to avoid collisions between CC challenge-response handshakes of joining nodes, e.g., with random waits. We plan to investigate this possibility in future work.

In Fig. 7 the average power consumption is reported. As expected the lowest power consumption is obtained with the unsecured mode, whereas the light-security configuration causes a slight increase in the average power consumption. This can be explained considering the computation overhead of the cryptographic operations and the increased size of the transmitted RPL messages due to the Security Section. With the full-security configuration, the power consumption increases noticeably, e.g., around 18% in the 5x5 scenario. This can be explained considering the additional number of messages exchanged during the join procedure executed by each node. As expected, results obtained with different network sizes show an increase of the power consumption when the network size grows. This is due to the larger number of messages exchanged among

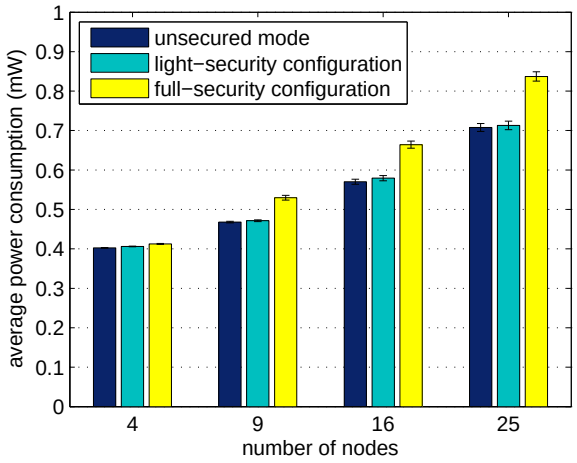

Fig. 7. Average power consumption.

the nodes, which increases the overall time spent by each node in receiving and processing messages.

Fig. 8 illustrates the average overhead due to RPL messages. The reported value includes the DIO and DIS messages, and also the CC messages exchanged in the full-security configuration. As expected, the lowest overhead is obtained with the unsecured mode. With the light-security configuration, instead, the overhead increases, as the presence of the Security Section increases the size of the RPL messages. With the full-security configuration, the overhead shows an additional slight increase, as additional messages are exchanged for the CC challenge-response handshakes. Coherently with the trend shown in Fig. 7, also in this case the overall overhead increases when larger networks are considered.

5.2 Impact on Reboot Recovery

The additional complexity introduced by the security on the rejoin operation influences mainly the recovery from a reboot of a node. The impact of the RPL security mechanisms on the node reboot recovery operation is assessed through the following metrics:

- *Reboot recovery time*, defined as the time between the reboot event and the moment in which the rebooted node joins again the DODAG. This metric measures the time required by the rebooted node to become fully operational again.
- *Power consumption*, defined as the power consumption of the rebooted node. This metric includes both communication costs (for transmission and reception of messages) and computation costs (for cryptographic operations and message processing).

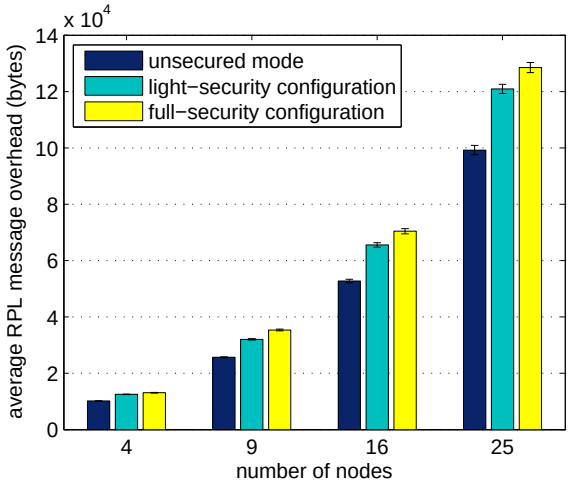

Fig. 8. Average RPL message overhead.

For each simulation run, a non-root node is randomly selected to reboot after 10 minutes of simulation. In order to obtain statistically sound results, 32 independent replications with different seeds are run for each scenario.

In Fig. 9 the average reboot recovery time is reported. As expected, the RPL unsecured mode is the one that results in a shorter delay, while the light-security configuration results in a delay which is slightly higher, due to the cryptographic operations that are required. The full-security configuration, instead, results in a delay which is higher only when large topologies are considered. This can be explained by considering that after rebooting the node receives a CC response for each neighbor, in order to synchronize again its counter watermark.

Finally, in Fig. 10 the average power consumption is reported. As expected, the full-security configuration increases significantly the energy consumption in large topologies, as it requires additional CC messages to be exchanged with each neighbor.

6 Conclusion

In this paper, we gave a first evaluation of the impact of the RPL security mechanisms on the WSAN performances. We developed a standard-compliant implementation of the RPL security mechanisms for the Contiki operating system [3], and we evaluated it by means of simulations. We showed that the RPL security mechanisms have a negligible impact on the performances in terms of network formation time and power consumption, if they do not have to defend against replay-based attacks. Otherwise, if also replay protection is needed, the impact on performances is more pronounced, especially on the network forma-

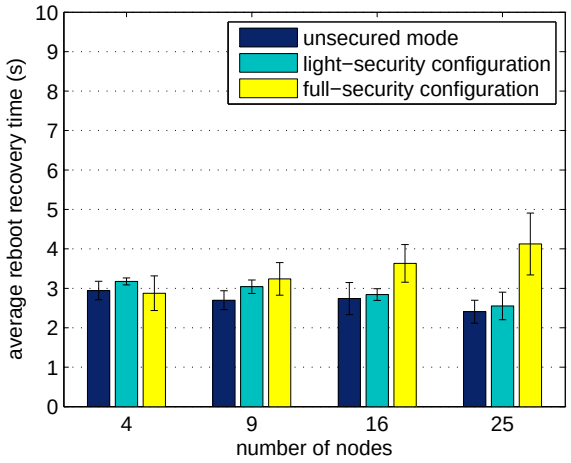

Fig. 9. Average reboot recovery time.

tion time which increases noticeably. To the best of our knowledge, this is the first implementation of the standard security mechanisms of RPL.

References

1. Airehrour, D., Gutierrez, J., Ray, S.K.: Securing RPL routing protocol from black-hole attacks using a trust-based mechanism. In: 26th International Telecommunication Networks and Applications Conference (ITNAC). pp. 115–120 (2016)
2. Dunkels, A., Eriksson, J., Finne, N., Tsiftes, N.: Powertrace: Network-level power profiling for low-power wireless networks. Tech. rep., Swedish Institute of Computer Science (March 2011)
3. Dunkels, A., Gronvall, B., Voigt, T.: Contiki – a lightweight and flexible operating system for tiny networked sensors. In: 29th Annual IEEE International Conference on Local Computer Networks (LNC). pp. 455–462 (2004)
4. Dvir, A., Holczer, T., Buttyan, L.: VeRA – version number and rank authentication in RPL. In: IEEE 8th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS). pp. 709–714 (2011)
5. Gaddour, O., Koubâa, A.: RPL in a nutshell: A survey. Computer Networks 56(14), 3163–3178 (2012)
6. Iuchi, K., Matsunaga, T., Toyoda, K., Sasase, I.: Secure parent node selection scheme in route construction to exclude attacking nodes from RPL network. In: 21st Asia-Pacific Conference on Communications (APCC). pp. 299–303 (2015)
7. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Networks 1(2), 293–315 (2003)
8. Le, A., Loo, J., Lasebae, A., Vinel, A., Chen, Y., Chai, M.: The impact of rank attack on network topology of routing protocol for low-power and lossy networks. IEEE Sensors Journal 13(10), 3685–3692 (2013)
9. Levis, P., Clausen, T., Hui, J., Gnawali, O., Ko, J.: The Trickle algorithm. RFC 6206, RFC Editor (2011)

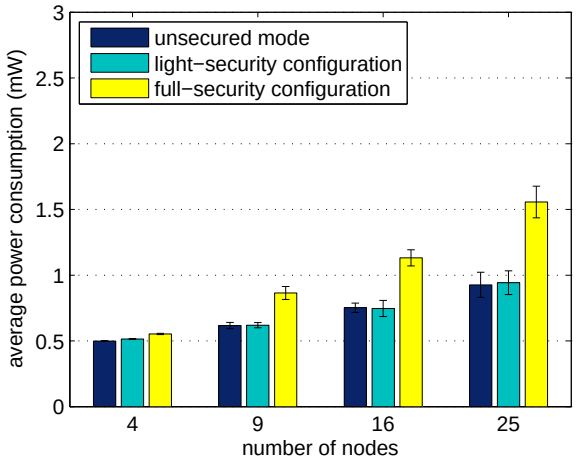

Fig. 10. Average power consumption.

10. Mayzaud, A., Badonnel, R., Chrisment, I.: A taxonomy of attacks in RPL-based Internet of Things. International Journal of Network Security 18(3), 459–473 (2016)
11. Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T.: Cross-level sensor network simulation with COOJA. In: 31st IEEE Conference on Local Computer Networks (LCN). pp. 641–648 (Nov 2006)
12. Perrey, H., Landsmann, M., Ugus, O., Wählisch, M., Schmidt, T.: TRAIL: Topology authentication in RPL. In: International Conference on Embedded Wireless Systems and Networks (EWSN). pp. 59–64 (2016)
13. Tsiftes, N., Eriksson, J., Dunkels, A.: Low-power wireless IPv6 routing with ContikiRPL. In: 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 406–407 (2010)
14. Weekly, K., Pister, K.: Evaluating sinkhole defense techniques in RPL networks. In: IEEE 20th International Conference on Network Protocols (ICNP). pp. 1–6 (2012)
15. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM). RFC 3610, RFC Editor (2003)
16. Winter, T.: RPL: IPv6 routing protocol for low-power and lossy networks. RFC 6550, RFC Editor (2012)