Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 760302, 10 pages
http://dx.doi.org/10.1155/2015/760302

Research Article

Adaptive Security in ODMAC for Multihop Energy Harvesting Wireless Sensor Networks

Alessio Di Mauro,[1] Xenofon Fafoutis,[2] and Nicola Dragoni[1,3]

[1]*DTU Compute, Technical University of Denmark, Richard Petersens Plads, 2800 Kongens Lyngby, Denmark*
[2]*Department of Electrical and Electronic Engineering, University of Bristol, Woodland Road, Bristol BS8 1UB, UK*
[3]*Centre for Applied Autonomous Sensor Systems, Örebro University, 701 82 Örebro, Sweden*

Correspondence should be addressed to Nicola Dragoni; ndra@dtu.dk

Received 11 December 2014; Accepted 26 March 2015

Academic Editor: Deyun Gao

Energy Harvesting Wireless Sensor Networks (EH-WSNs) represent an interesting new paradigm where individual nodes forming a network are powered by energy sources scavenged from the surrounding environment. This technique provides numerous advantages, but also new design challenges. Securing the communications under energy constraints represents one of these key challenges. The amount of energy available is theoretically infinite in the long run but highly variable over short periods of time, and managing it is a crucial aspect. In this paper we present an adaptive approach for security in multihop EH-WSNs which allows different nodes to dynamically choose the most appropriate energy-affecting parameters such as encryption algorithm and key size, providing in this way energy savings. In order to provide evidence of the approach's feasibility in a real-world network, we have designed and implemented it as extension of on-demand medium access control (ODMAC), a receiver-initiated (RI) MAC protocol specifically designed and developed to address the foundational energy-related needs of Energy Harvesting Wireless Sensor Networks.

1. Introduction

Wireless sensor networks (WSNs) are more and more pervasive, consistently used to perform many different kinds of monitoring tasks, ranging all the way from outdoor surveillance to body area networks. The classic means of operation for sensor nodes have been batteries; nevertheless thanks to technological advancements Energy Harvesting Wireless Sensor Networks (EH-WSNs) are quickly becoming a reality. In this scenario, nodes are able to obtain the energy required to operate from the surrounding environment. Different energy sources can be used, with some being more efficient and prominent than others, but the key idea is that the theoretical infinite lifespan of EH-WSN poses unique challenges as nodes can run for a much longer period of time, allowing for more permanent instalment and new use scenarios. Furthermore, EH-WSNs are characterized by spatial inconsistencies: depending on the particular source of energy being scavenged it is not uncommon to find completely different energy situation between different portions of the network, independent of their physical correlation. For example, imagine a solar-powered EH-WSN with nodes lying on either side of a wall. The nodes might be well in range and able to communicate, but depending on the time of day one side of the network could be in the shadows and unable to harvest energy. Depending on the topology and the application being run, this might impact availability or even disconnect the network.

Given their use in critical situations, WSNs require solid and reliable security capabilities. For this reason, security research on wireless sensor networks (WSNs) has flourished over the past years. However, the introduction of energy harvesting capabilities represents a game changing scenario which has yet to be fully researched. Previous WSNs technologies may not work for the EH-WSNs paradigm and new specifically tailored solutions need to be designed in order to

take full advantage of what energy harvesting can offer, for instance, concerning security [1].

The Need for Adaptive Security. The building block of security mechanisms for WSNs is encryption schemes. Independently from the specific application, what normally happens is that the data channel is made confidential and/or authentic through the use of encryption schemes and related modes of operation. The typical family of algorithms used with sensor nodes are symmetric encryption algorithms since they are considerably less expensive in terms of energy requirements when compared to public key encryption schemes [2]. Different algorithms have different energy requirements and while some of these are connected to how good and optimized the actual implementation is, a considerable portion is intrinsic to the specific algorithm. It is logical to expect that a block cipher with a block size of 128 bits will require more CPU cycles than an algorithm with a block size of 64 bits in order to perform similar operations. A similar point can be made for the key size of an algorithm; a longer key is bound to produce higher energy requirements, despite the fact that it should also increase the complexity of the cryptanalysis and the robustness of the cipher-text. For this reason, when energy is a big concern, having to commit to a specific algorithm is going to be a suboptimal decision. In an energy harvesting (EH) scenario, a specific scheme can be inadequate in different ways: for example, it could be too expensive in terms of energy and cause the whole system to delay sending new messages until enough energy has been gathered. Within a network with heterogeneous messages, a given scheme could not meet the security requirements for a particular type of message, while it could be more than enough for a different type.

Contribution and Outline of the Paper. In this paper we discuss and propose an adaptive security scheme that allows each node to autonomously and independently choose the most suitable security algorithm to use for a given link of the network and for a given energy configuration. The idea is that each node can advertise all the different supported schemes and dynamically adjust them, depending on the current energy situation. To make the approach concrete, we have built our solution on on-demand medium access control (ODMAC) [3], a receiver-initiated (RI) medium access control (MAC) protocol specifically designed and developed to address the foundational needs of EH-WSN. Indeed, ODMAC messages are sent in clear and there is no control over their authenticity, allowing an attacker to eavesdrop the communication, intercept messages, and forge new ones. The goal has been to extend ODMAC with adaptive security while still maintaining a low-resource profile, keeping the protocol suitable for EH-WSNs.

The paper is organised as follows. In Section 2 we briefly introduce ODMAC. Then in Section 3 we extend ODMAC with an adaptive security suite. We discuss implementation and experiments in Section 4 and we focus on some energy and security considerations in Section 5. Sections 6 and 7 close the paper with related works and conclusions, respectively.

2. ODMAC Protocol

The key feature ODMAC provides in order to support EH-WSN is to allow each node to independently choose its own duty-cycle (DC) and adjust it according to different parameters like the harvesting rate of a node or the requirements of the application [3, 4]. Thanks to that, a trade-off between power and performance can be obtained. For example, in case of scarce energy the DC can be decreased to give time to the node to harvest more and gather enough to survive a communication. On the other hand, when energy is abundant the DC can be increased and as a result also the performance of the network will increase: more packets exchanged will translate to increased throughput and decreased delay.

To achieve this, the protocol relies on the RI paradigm [5], which not only has proven to be more energy efficient than its counterpart sender-initiated (SI) [6], but also is a good match for DC adaptation. A node running ODMAC has two different DCs, one for exchanging messages and one for sensing purposes. These two DCs are, respectively, called the *beaconing* and the *sensing* duty-cycle.

As the name of the paradigm suggests, the message transmission starts from the receiver (Figure 1). Whenever a beaconing period elapses and a node r ready to receive data enters the active state, it will perform a clear channel assessment (CCA) to determine whether or not there is an ongoing transmission already happening. If the channel is available, r will transmit a beacon b manifesting its intentions to receive a packet and it will then start to listen to the channel for incoming packets for a fixed amount of time.

Similarly, when a sender s enters the active state because a sensing event has occurred, it will start listening to the channel for an appropriate beacon. This is a beacon that satisfies specific predefined conditions such as moving the packet closer to its destination. Should such a beacon be received s will immediately transmit its packet and go into sleep mode. In the upcoming wake-ups, s waits for a new beacon from r which will work as an acknowledgment (ACK) for the packet sent previously.

Whenever data is received by a node, if the node itself is not the final recipient (as it will happen most of the times in a multihop network), a forwarding procedure will begin. This is identical to the sensing and transmission operation described above, with the exception that, instead of being generated locally, data is obtained from another node.

Since ODMAC has specifically been designed for EH-WSNs, its guiding design principle is sustainability. More specifically, a node adapts its DCs to remain operational with the current energy situation, aiming to achieve an energy neutral operation (ENO) state [7]. The concept of ENO refers to a state where the energy consumed by a node is always less than or equal to the energy harvested from the environment. This state guarantees infinite lifetime as soon as there are not any hardware failures. The way this is obtained is through a feedback loop where a node is able to monitor its current energy level and initiate a communication when the value is high enough.

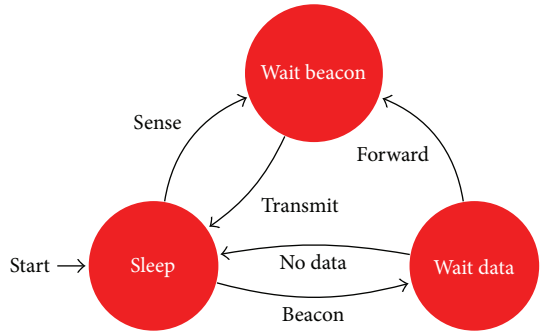

Figure 1: ODMAC described as a finite state machine.

ODMAC provides many advanced features that makes the protocol specifically suitable for EH-WSNs, such as *opportunistic forwarding*, *altruistic backoff*, and *layer-based anycast routing*. Introducing all these features is outside the scope of the paper. Interested readers can refer to the ODMAC literature [3, 4, 8, 9] for all the design and implementation details.

3. Extending ODMAC with Adaptive Security

The key limitation of ODMAC and, as a consequence, of the resulting ODMAC-based network is that it does not provide any security suite. Even the key security services like confidentiality, integrity, and availability are not provided by ODMAC. This means that messages are sent in clear and there is no control over their authenticity, allowing an attacker to eavesdrop the communication, intercept messages, and forge new ones. Thus, the goal is to extend ODMAC in order to achieve much better security properties while still maintaining a low-resource profile and remaining suitable for EH-WSNs. The key approach to achieve this goal is *adaptive security*.

3.1. Modes of Operation. We have extended ODMAC by developing a security suite inspired by TinySec [10]. The security suite has four modes of operation: *no security, authentication, encryption, and authentication + encryption* which can be chosen on a per-message basis, allowing for full customization from the user. When both authentication and encryption are chosen, they are composed using the secure *encrypt-then-MAC* paradigm [11].

The scheme supports encryption algorithms with 64-bit blocks and 80-bit or 128-bit keys. In our proof of concept implementation we have used Skipjack [12] which requires 20 B of RAM and around 6.5 KiB of ROM. Despite its age, Skipjack continues to prove secure for an 80-bit key algorithm [13]; the most successful attack so far is an impossible differential attack on 31 of the 32 rounds yielding a result marginally faster than exhaustive search [14]. Considering that the National Institute for Standards and Technology (NIST) has proposed to phase out the use of 80-bit keys by 2015, it is a good idea to turn the attention to algorithms supporting 128-bit keys. Depending on the application and the implementation, different ciphers can be used [15]. Piccolo [16] and TWINE [17] are good candidates for software implementations, respectively, requiring 91 B and 23 B of RAM and 2.5 KiB and 2.2 KiB of ROM. For hardware implementation the best candidate is PRESENT [18] which uses 1886-gate equivalent (GE) and has also been included in the standard for lightweight cryptographic methods by the International Organization for Standardization (ISO) [19].

Encryption is carried out using cipher block chaining (CBC) mode with cipher-text stealing to avoid last block message expansion, while authentication is done with cipher block chaining message authentication code (CBC-MAC). This allows using one encryption algorithm to perform both operations. It is important to highlight that a key for the system in *authentication + encryption* mode is actually a pair of keys, one for each operation. Authentication codes and encryption are checked and recomputed at each hop. This has the advantage of intercepting maliciously or fortuitously malformed packets as early as possible, avoiding the waste of energy to route them to their final destination only to discard them there.

A possible extension to the scheme is to use an authenticated encryption (AE) mode. The advantage of this technique is that it is possible to obtain both authentication and encryption at the same time, without having to run the algorithm twice. The highest performance algorithms are offset codebook (OCB) (used in MiniSec [20]) and Galois counter mode (GCM), with the former having the better performance [21]. Unfortunately OCB is patented and could not be used freely until recently (9 January 2013), when a free license has been issued for open-source noncommercial application. The algorithm is still not free for commercial applications. The other mode, GCM, is notoriously cumbersome to implement correctly. A comparison between different implementations in TinySec can be found in [22] where CBC and GCM are analyzed in conjunction with both the advanced encryption standard (AES) and Skipjack. The results show that GCM in combination with AES obtains a 12% increase in energy consumption, a 28% increase in RAM usage, and a 35% decrease in throughput compared to the original implementation of TinySec. While constantly outperforming CBC in combination with AES, it is still a considerable decrease in performance, justifiable only if the application requires both authentication and encryption without any differentiation.

3.2. Adaptive Security Scheme. The key feature the scheme provides is to allow each node of the network to independently choose the best compromise between security and energy consumption according to different metrics. This scheme is inspired by [23]. However, differently from that, we allow dynamic multihop communication, making the approach more suitable to concrete, real-world scenarios.

A WSN is characterized by nodes producing and exchanging packets. Upon creation, each packet p_i is assigned a security value $h_{p_i} := H(p_i)$, where $H : P \rightarrow E \times A$ is a function mapping elements from the set of possible packets P to tuples representing security configurations. This function assesses the criticality of a specific packet. We will abstract from its implementation, but it could be thought as

a direct connection between specific parameters of a packet and importance values. For example, packets representing aggregate values could be considered more important than single measurements, or potentially harmful control packets (e.g., a message asking to reduce the transmission power) would be rated higher than regular messages.

As described before, the h values are tuples $(e, a) \in E \times A$, where each component directly translates into a specific security configuration of encryption and authorization, respectively. Different values are mapped to different algorithms and parameters. This mapping can be decided at design-time of the specific network application. An example can be seen in Table 1, where we describe only encryption modes assuming tuples of the form $(e, 0)$. Another possible demonstration can be the default security protocol list of 802.15.4 [24, Table 75], which comprises encryption, authentication, and authenticated encryption.

The protocol relies heavily on the RI paradigm; whenever a receiver node r transmits a beacon, it will include its security capabilities $c^t_{r,\max}$ and $c^t_{r,\min}$; these are, respectively, the highest h tuple that r can satisfy and the lowest h tuple that r will accept, at time t. A sender node s can then analyze beacons to check if both the destination and the security capabilities of its owner are satisfactory. Assuming a total ordering on the security capabilities, let $\tilde{r}$ be the final recipient for node s (e.g., the base station) and $\Delta(u, v)$ the function that measures the distance in number of hops between two nodes u and v; then a beacon b from node r is considered adequate for packet p_i if and only if $\Delta(r, \tilde{r}) < \Delta(s, \tilde{r}) \wedge c^t_{r,\min} \leq h_{p_i} \leq c^t_{r,\max}$, that is, if the distance between p_i and its final destination $\tilde{r}$ decreases by sending p_i to r and r can satisfy the security requirements of p_i. Note that we use a strict inequality for the distance to account for ODMAC opportunistic forwarding [4], where beacons moving a message closer to the final destination of the packet are still considered adequate even if they could be suboptimal from a routing standpoint.

The pseudocode for data transmission and reception can be seen in Algorithms 1 and 2. The key point of these algorithms is the generation of the c values (lines 1.5, 1.6, 2.6, and 2.7). These values are tightly connected to the amount of energy available in a node and to the security policies of the system. The notion of available energy is something constantly varying, especially in EH-WSNs. It may be the case that a node has only enough energy to run in *(No Security, No Security)* mode at the current time. However, that situation might improve after it has been able to scavenge some more energy. On the other hand security policies can impose both static and dynamic values according to the specific type of application. In the rest of the section, we will show through some examples how different scenarios can be accommodated by adapting how security values are generated. Before that, it is important to stress another point concerning the sensitivity of the data to be transmitted. Indeed, the level of sensitivity of the information belongs to the application logic of the node. In other words, our framework provides the possibility for the node to choose among different security levels. It is then up to the application logic to decide which security level is appropriate for a specific message, according to the data to be sent and the energy available. For instance, if the information waiting to be transmitted is not sensitive, then the node can adopt the *(No Security, No Security)* mode for that transmission, independently of the energy available (which could allow for a better security level).

Table 1: An example of the H-security mapping.

$H(x)$	Encryption	Authentication
$(0, 0)$	No	No
$(1, 0)$	Skipjack (key size 80 bits, block size 64 bits)	No
$(2, 0)$	HIGHT (key size 128 bits, block size 64 bits)	No
$(3, 0)$	AES128 (key size 128 bits, block size 128 bits)	No

```
(1)  function SEND_DATA(data, dest)
(2)      Packet p
(3)      Beacon b
(4)      p.id ← self.id
(5)      p.e ← SET_ENC_CAPABILITIES()
(6)      p.a ← SET_AUTH_CAPABILITIES()
(7)      p.data ← PACK_DATA(data)
(8)      repeat
(9)          b ← WAIT_FOR_BEACON()
(10)     until Δ(b.id, dest) < Δ(p.id, dest) AND
(11)         b.e_min ≤ p.e ≤ b.e_max AND
(12)         b.a_min ≤ p.a ≤ b.a_max
(13)     TRANSMIT(p, b.id)
(14) end function
```

Algorithm 1: Adaptive security data transmission.

```
(1)  function RECEIVE_DATA()
(2)      Packet p
(3)      Beacon b
(4)      b.id ← self.id
(5)      repeat
(6)          (b.e_min, b.e_max) ← SET_ENC_CAPABILITIES()
(7)          (b.a_min, b.a_max) ← SET_AUTH_CAPABILITIES()
(8)          TRANSMIT(b)
(9)          p ← WAIT_FOR_PACKET()
(10)     until p ≠ nil
(11)     data ← UNPACK_DATA(p.data)
(12)     return data
(13) end function
```

Algorithm 2: Adaptive security data reception.

Static Configuration. The first scenario can be used to understand how our scheme works at its core. Senders generate c values for outgoing packets according to the amount of energy available to the node at time of creation, using a lookup-table to match security configurations and energy requirements. A simple definition of this routine can be seen in Algorithm 3; note that here the sender is not taking into

```
(1) function SET_ENC_CAPABILITIES()
(2)   E ← GET_CURRENT_ENERGY()
(3)   return encryption_scheme[E]
(4) end function
```

Algorithm 3: Encryption capabilities generation in static configuration.

account the criticality of the data to choose the security configuration but rather is using a "best effort" kind of strategy. At the same time, the receiving side of each node is statically assigned security ranges. As a result the system can be seen as a weighted directed graph where an edge from u to v of cost c means that u can communicate to v (is physically in range), but only provided that it uses the security features represented by c. The way to obtain this behavior is to set $c_{\min}$ and $c_{\max}$ to the same value. By doing this a receiver can decide the security class of the packets to accept.

Dynamic Configuration. The static configuration is good for describing how the model works; however its utility is limited. An extension is presented in the *dynamic configuration*. WSNs often cover large geographical areas such as forests or fields. It could be the case that treating the whole area as a single zone with some fixed properties is not the best approximation. Imagine an example where a network is deployed in an area covering two different buildings connected by an open space. It is sensible to believe that the nodes inside the buildings will be susceptible to fewer risks compared to the nodes out in the open. For this reason, in dynamic configuration, we allow the node at each hop to reconfigure the h value of a packet, increasing it when moving towards a less secure zone or decreasing it when moving away from such a zone. Here the sender has also to address the importance of the packet as part of the process, making sure that important packets are not underprotected which would result in a security issue or that less important packets are not overprotected leading to a waste of energy. Similarly to senders, receivers can adapt their advertised c values depending on the specific area they are in and the amount of energy currently available. The pseudocode for the c values generation in this case can be seen in Algorithm 4.

As a result, a more fine-grained approximation of the area can be achieved, allowing a more conscious use of the available energy.

Path Configuration. A third way for tailoring our scheme to a specific application is by using *path configuration*. Here the idea is to force packets through specific paths by carefully choosing the $c_{\min}$ values advertised by the receivers. Assuming that multiple paths are available to one destination and that the h value of a packet is related only to its importance, a receiver can dynamically choose to accept different types of packet by adjusting the value of $c_{\min}$. For example, imagine that we would like the network in Figure 2 to route all the packets containing aggregate measurements through nodes a, b, and c, whereas we do not care where

```
(1) function SET_ENC_CAPABILITIES()
(2)   E ← GET_CURRENT_ENERGY()
(3)   Z ← GET_ZONE_ID()
(4)   c ← encryption_scheme[E]
(5)   if is_low_security(Z) then
(6)     c ← c + x
(7)   end if
(8)   return c
(9) end function
```

Algorithm 4: Encryption capabilities generation in dynamic configuration.

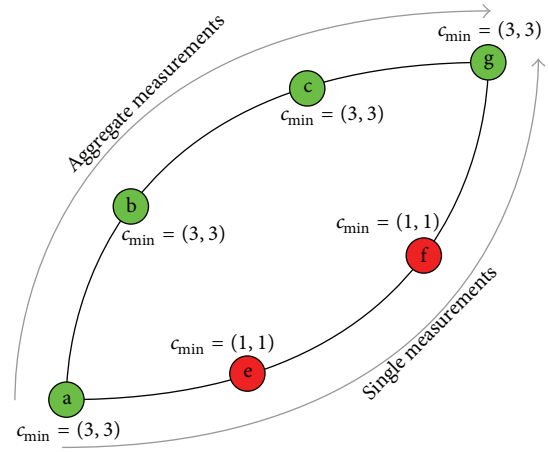

Figure 2: Example of how routing can be affected in path configuration. The high security packets will be sent through nodes a, b, and c, while the low security ones will travel through e and f.

single measurements packets are routed. This can be achieved by setting $c_{\min}$ to $(3, 3)$ in a, b, and c and to $(1, 1)$ in the remaining nodes. We also have to make sure that aggregate packets are assigned h values of at least $(3, 3)$ and they will be picked up only by nodes a, b, and c as wanted. Furthermore, the values advertised by receivers can be again dynamically varied according to the situation of the network. For example, if nodes a, b, and c become unavailable for a period of time, other nodes can take over their duties by increasing their own $c_{\min}$ to accept aggregated packets. Another possibility is to dynamically react to a localized attack (e.g., jamming) by redirecting traffic to a safe area of the network.

Discussion. The configurations provided above are meant to be examples and guidelines on how the scheme itself can be adapted to different scenarios and application requirements and are by no means meant to be exhaustive. Having the possibility to modify the behaviour of both senders and receivers allows for considerable flexibility, enabling the design of solutions that are tailored to the problem at hand and hence can guarantee good performances. For instance, imagine an application where delay is a main concern; that is, packets should arrive from the nodes to the base station as quickly as possible. In this scenario, waiting for a beacon advertising

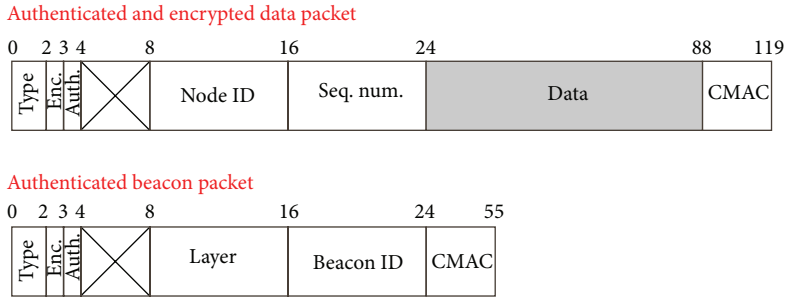

Figure 3: Fully secured packet format for ODMAC. An authentication code is added at the end of both types of message, and the data field is also encrypted.

the best energy to security ratio costs precious time. It is instead possible to have nodes try to send packets using the first useful beacon, regardless of its security parameters.

At the other side of the spectrum we could have an application where security is the main focus. Here each node could keep track of the security configuration advertised through different beacons and only use the best seen so far to relay messages, possibly using a weight function that provides diminishing returns according to how old a packet is.

4. Implementation and Experimental Results

We implemented the ODMAC security suite, as an extension of the ODMAC implementation presented in [4], on the eZ430 platform by Texas Instruments. For space limitations, we will focus only on the security aspects of the implementation and interested readers can refer to [4] for more details on the ODMAC implementation. The ODMAC security suite incorporated some modification to the transmission routines, but it is generally transparent to the user which only has to set the desired mode. When a packet is created and about to be sent, it is first encrypted (if required) and then authenticated (again, if required). The well-known advantage of applying those transformations in this order is that integrity of the cipher-text is provided and, as a consequence, also integrity of the plain-text. Furthermore, it is not possible to maliciously modify the cipher-text so that it will be decrypted to some other (meaningful) plain-text, and finally assuming that the output of the encryption will appear to be random, so will the result of the cryptographic message authentication code (CMAC), preventing structural information from leaking through.

When a message is received the opposite procedure is performed. First of all, the CMAC is verified; if there is a mismatch the message will be discarded and the node informed with a corresponding error. This prevents malformed messages from having an impact on the whole network by being forwarded to other nodes, causing them to spend unnecessary energy in the process. Once the message is authenticated it will be decrypted and the normal behaviour of the node will continue. The packet format for the highest security mode (authentication and encryption) is shown in Figure 3.

Several experiments on ODMAC can be found in literature [3, 4], covering different EH-WSN scenarios. Our experiments are meant to complement those experiments by focusing on ODMAC security. From this perspective, we are particularly interested in showing the impact of the security suite. To this aim, let us consider a scenario where two kinds of packets are available to a sender node: low security and high security ones. Low security packets will be sent unencrypted, whereas high security ones must be encrypted and authenticated. Whenever possible we give precedence to high security packets, meaning that if enough energy is available an encrypted and authenticated message will be sent. An unsecured message will be sent otherwise.

For the experiment we used a single link with a transmitter that is powered through photovoltaic cells using the CBC-EVAL-09 platform. In a controlled environment, we supply the system with a constant level of illumination. In parallel, we feed the voltage of the output capacitor of CBC-EVAL-09 to the analog-to-digital converter of eZ430, effectively making energy-aware. In this setting, we define two voltage levels. The normal behavior of the sender node would then be the following: upon waking up it would check the capacitor value; if the value is above the higher threshold the node would send a high security packet, if the value is between the high and the low thresholds the node would send a low security packet, and if the value is below the low threshold the node would go back to sleep. A wake-up event, that is, a packet transmission attempt, is scheduled at a period of 10 s.

We first run a control experiment only sending low security packets, thus disabling the security overhead. Afterwards, we have run the full experiment with both thresholds. Both experiments lasted for 90 minutes and are summarized in Figure 4. In the control case we were able to send 525 packets, averaging 0.097 packets/s (0.97 packets/10 s). This value is very close to the theoretical limit of 1.00 packet/10 s. For the full-blown experiment we managed to transmit 486 packets over the entire duration, 397 of which were high security (encrypted and authenticated), while the remaining 89 were low security. As a result we achieved a cumulative average of 0.090 packets/s (0.90 packets/10 s) and an average rate of 0.073 packets/s (0.73 packets/10 s) for high security packets only. Despite having added more functionalities and tightened the energy requirement by increasing the transmission threshold, we were able to maintain similar performances.

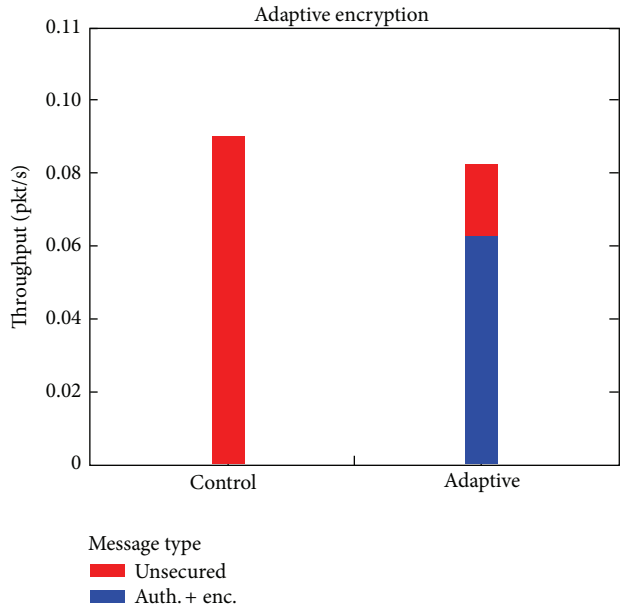

Figure 4: The adaptive, security-enabled scheme has tighter energy requirements but shows a decrease in packets sent of only 7.42%.

Observe that since the two scenarios used the exact same energy budgets, we can infer the actual power overhead of the security system; transmitting a high security packet consumes on average 9.8% more energy than a low security packet. This demonstrates the significance of an adaptive security system from a different perspective. Based on this estimation, if we blindly use the highest security, also for packets that do not require it, we would be able to transmit only 478 packets, that is, 8 packets less.

5. Energy and Security Considerations

In this section we will provide some considerations on the scheme in general, including how it is possible to obtain the different information required by the protocol and some considerations on the security guarantees of our scheme.

5.1. Energy Management. Each node has to be aware of the current state of charge (SOC) of its main battery to decide which security values to advertise through beacons. This can be realized with different methods, the most common being Voltage Based Estimation and Coulomb Counting. In the first method it is possible to directly measure the voltage across the battery and relate this to the actual SOC by means of the discharge characteristics relative to the specific chemistry process used within the cell. For this method to achieve reasonable accuracy, compensation factors for temperature, cell age, and discharge rate should be factored in, making it slightly less practical for WSNs. With Coulomb Counting the idea is to consider the battery as a closed system containing a given amount of charge, when full, and subtracting from this value as the battery depletes. In order to measure the actual current drawn from the battery different sensing techniques such as shunt resistors or hall effect sensors can be used.

To correctly assess the cost of a specific configuration it is required to measure the amount of energy needed in order to use it. This can be done empirically, by measuring the state of charge of the battery before and after a large enough number of transmissions and then computing the average in offline experiments.

Once these quantities are known, it is possible for a node to correctly advertise the currently supported configurations. As the state of charge varies over time, less or more configurations will become available and the newly created beacons will reflect the situation adaptively changing the supported features. This is where RI protocols shine: thanks to their core mechanic it is extremely easy to convey information from the sender to the receiver before the actual packet is sent, without having to perform unnecessary communications. The sender can then use this information to decide whether or not the receiver is appropriate.

5.2. Security Considerations. Possible attacks to this protocol are closely related to the underlying MAC protocol and the encryption algorithms used. Without touching on the security of the individual algorithms, which is out of the scope of this work, we now analyze what a potential adversary might be able to achieve by manipulating messages within the network and taking advantage of the protocol inner workings. This analysis ties into the correct design of the system and can help define security properties according to the required features. For the adversary model in this section we will consider a Dolev-Yao attacker [25] which is aware of the protocol and is able to eavesdrop, intercept, and create new messages using the knowledge accumulated over time.

A first concern is about forging beacons. This would allow the adversary to advertise incorrect security capabilities or malicious routing information in the form or wrong identities. This problem is avoided by ensuring that either an encryption or an authentication layer is always present. The attacker would have to share a key with other nodes in order to be able to communicate fresh messages with them. This argument relies on the secrecy and the strength of the key, which is in line with the attacker model. If, for example, poor key exchange mechanisms are used within the application and the adversary can get hold of the key, the security of the scheme is obviously defeated.

It is worth pointing out that while using either authentication or encryption provides the same upshot (packets cannot be forged), the way this is obtained is slightly different. In case of encryption we can say that, in order to create a counterfeit beacon, the adversary should produce a key that would allow him to create packets that would be correctly and meaningfully decrypted by recipients nodes; in other words he would have to have a shared key with all the target nodes. The number of such keys depends on the keying scheme used: single-key, probabilistic, group-based, or pairwise, just to name a few.

On the other hand, in the authentication case, the adversary would have to produce a key that can validate the content of the message against a tag appended at the end of the message itself, proving that the identity of the owner of the message is legitimate. This is a separate key that can be

managed in a completely different way from the other one; for example, it could be a single key which once compromised gives the possibility to exchange authentic messages with every other node within the network.

Furthermore, if the system allows nodes to dynamically join the network, it becomes much harder to discover an attacker that tries to disguise himself as a regular node, complies to the protocol long enough to establish a genuine identity, and then goes rogue. The result of this is that while both systems guarantee the freshness of a message, the decision of which is better suited to deal with the problem, as often happens, lies in the details like the key management scheme used.

A second possible course of action for the adversary is to try and spoof or modify received beacons in order to retransmit them at a later time. This is avoided by using authentication schemes which, by definition, prevent messages from being modified. In other words, by using security suites like our scheme presented in Section 3 both integrity and confidentiality are achieved.

Since the adaptive scheme is based upon EH, energy exploitation must be carefully taken into account. While an attacker with physical access to the node could in theory prevent it from recharging and keeping it in a low security state, we believe this is not an effective attack. First of all if an attacker has physical access to a node, energy exploitation is not the main concern, but rather the node could be cloned or reprogrammed or have secret keys extracted from it, all kinds of attacks that would cause much greater harm to the whole network. Secondly if the attacker wants to have some kind of distributed effect on the network by changing the current energy parameters, he must do so for a considerable number of nodes, and, depending on the actual network size and the kind of energy used to power the nodes, this could be unfeasible.

Finally, one more concern is about replayed beacons. As we have discussed previously, this technique can be used to impersonate another entity, carrying out communications on behalf of the entity and trying to gain some advantage from it. Depending on how this is done, it is possible to force senders to use lower than necessary security settings in order to obtain cryptographic advantage or to force higher than necessary security settings, thus making nodes use more energy for each message exchange and shortening their active time, possibly causing a denial of service. Other ways of performing this kind of attack are similar in principle but are a bit more subtle; for example, an adversary could monitor the traffic looking for nodes important to the specific application, like nodes forwarding traffic in a high security path, nodes with a high incoming degree (topology bottlenecks), or nodes performing critical measurements. Once such potential targets have been identified, the attacker can then use a series of replayed messages in order to selectively disrupt the victims. A solution to this issue can be found in receiver authentication protocol (RAP) [26], the scheme which specifically targets the beacon replay attack. RAP can be used on top of any security mode and can be factored into the design of the system.

6. Related Work

Adaptive security is not a brand new concept. The work in [23] uses a similar environment and a similar approach, additionally focusing on priority, but limited to single-hop networks with carrier sense multiple access (CSMA). The authors say that, rather than achieving an absolute decrease in energy consumption, they manage to obtain a trade-off among consumed energy, importance of the packets sent, and their security.

Another example of adaptive security can be found in [27]. Here optical wireless communications are taken into account. The authors propose to subdivide an encryption system S into n subsystems $S_1, S_2, S_3, \ldots, S_n$ each one representing one encryption parameter such as key size, number of rounds, or operation mode. The idea then is to vary those parameters according to the security requirements or the amount of available energy in the case of battery powered devices.

The work in [28] is closer to static analysis. Three main parameters are used to define the security level of a protocol: the protection level, the probability of an attack, and the impact of a successful attack. Concerning the protection level, parameters such as the efficacy of an attack (provided it is successful), the knowledge required to mount it, its cost, the communication overhead, and the complexity of the implementation are considered. Similarly, the impact of a successful attack is calculated according to the financial losses during the attack, the cost for recovering from the attack, and the losses in reputation suffered by the owners of the system. Finally, the probability of an attack is assumed to be given. These values are composed to obtain a single *security level*. Individual security mechanism is then analyzed and defined in terms of complexity and power consumption. Ultimately, according to the system specifications, the required security capabilities, and the provided cost functions, specific security parameters are chosen and the system is run accordingly.

7. Conclusion and Future Work

In this paper we have proposed an adaptive approach for security in multihop Energy Harvesting Wireless Sensor Networks (EH-WSNs). The rationale behind the approach is to allow each node to autonomously and independently choose the most suitable encryption/authentication algorithm to use for a given link of the network and for a given energy configuration. The idea is that each node can advertise all the different supported schemes and dynamically adjust them, depending on the current energy situation. In order to provide evidence of the scheme's feasibility in a real-world network, we have designed and implemented it as extension of ODMAC [3, 4], a RI MAC protocol specifically designed and developed to address the foundational energy-related needs of EH-WSN. Our experimental analysis suggests that the energy consumption overhead of the system is 9.8%, while demonstrating that an adaptive solution is performing better than blindly providing the highest security.

Although the proposed scheme has been implemented as extension of ODMAC, from a design perspective the security

scheme is independent of the specific class/family of MAC protocols adopted (in the case of this paper, RI MAC protocols [5]). This means that our scheme can be adapted to work also with the other important class of MAC protocols, namely, sender-initiated (SI) MAC protocols. This represents our key future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was partially supported by the IDEA4CPS project granted by the Danish National Research Foundation.

References

[1] A. Di Mauro, D. Papini, and N. Dragoni, "Security challenges for energy-harvesting wireless sensor networks," in *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems (PECCS '12)*, pp. 422–425, SciTePress, 2012.

[2] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, 2005.

[3] X. Fafoutis and N. Dragoni, "ODMAC: an on-demand mac protocol for energy harvesting—wireless sensor networks," in *Proceedings of the 8th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '11)*, pp. 49–56, ACM, November 2011.

[4] X. Fafoutis, A. Di Mauro, and N. Dragoni, "Sustainable medium access control: implementation and evaluation of ODMAC," in *Proceedings of the 4th Workshop on Energy Efficiency in Wireless Networks and Wireless Networks for Energy Efficiency (E2Nets '13)*, pp. 407–412, 2013.

[5] X. Fafoutis, A. Di Mauro, M. D. Vithanage, and N. Dragoni, "Receiver-initiated medium access control protocols for wireless sensor networks," *Computer Networks*, vol. 76, pp. 55–74, 2015.

[6] E.-Y. A. Lin, J. M. Rabaey, and A. Wolisz, "Power-efficient Rendez-vous schemes for dense wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, vol. 7, pp. 3769–3776, IEEE, June 2004.

[7] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, article 32, 2007.

[8] X. Fafoutis, A. di Mauro, and N. Dragoni, "Sustainable performance in energy harvesting: wireless sensor networks," in *Proceedings of the 4th International Conference on Future Energy Systems (e-Energy '13)*, pp. 267–268, ACM, May 2013.

[9] X. Fafoutis, C. Orfanidis, and N. Dragoni, "Altruistic backoff: collision avoidance for receiver-initiated mac protocols for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 576401, 11 pages, 2014.

[10] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor network," in *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, ACM, 2004.

[11] T. Kohno, A. Palacio, and J. Black, "Building secure cryptographic transforms, or how to encrypt and MAC," *IACR Cryptology ePrint Archive*, vol. 177, 2003.

[12] National Institute of Standards and Technology (NIST), SKIPJACK and KEA Algorithm Specifications, 1998.

[13] J. Kim and R. C.-W. Phan, "A cryptanalytic view of the NSA's Skipjack block cipher design," in *Advances in Information Security and Assurance: Third International Conference and Workshops, ISA 2009, Seoul, Korea, June 25-27, 2009. Proceedings*, vol. 5576 of *Lecture Notes in Computer Science*, pp. 368–381, Springer, Berlin, Germany, 2009.

[14] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials," *Journal of Cryptology*, vol. 18, no. 4, pp. 291–311, 2005.

[15] Y. W. Law, J. Doumen, and P. Hartel, "Survey and benchmark of block ciphers for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 65–93, 2006.

[16] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "*Piccolo*: an ultra-lightweight blockcipher," in *Cryptographic Hardware and Embedded Systems—CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1, 2011. Proceedings*, vol. 6917 of *Lecture Notes in Computer Science*, pp. 342–357, Springer, Berlin, Germany, 2011.

[17] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE: a lightweight block cipher for multiple platforms," in *Proceedings of the 19th International Conference on Selected Areas in Cryptography (SAC '13)*, pp. 339–354, Springer, 2013.

[18] A. Bogdanov, L. R. Knudsen, G. Leander et al., "PRESENT: an ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems—CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings*, vol. 4727 of *Lecture Notes in Computer Science*, pp. 450–466, Springer, Berlin, Germany, 2007.

[19] International Organization for Standardization (ISO), *Information Technology—Security Techniques—Lightweight Cryptography—Part 2: Block Ciphers*, International Organization for Standardization, London, UK, 2012.

[20] M. Luk, G. Mezzour, A. Perrig, and V. D. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, pp. 479–488, ACM, April 2007.

[21] T. Krovetz and P. Rogaway, "The software performance of authenticated-encryption modes," in *Fast Software Encryption: 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13–16, 2011, Revised Selected Papers*, Lecture Notes in Computer Science, pp. 306–327, Springer, Berlin, Germany, 2011.

[22] V. Jariwala and D. C. Jinwala, "Evaluating Galois Counter mode in link layer security architecture for wireless sensor networks," *International Journal of Network Security & Its Applications*, vol. 2, no. 4, pp. 55–65, 2010.

[23] A. V. Taddeo, M. Mura, and A. Ferrante, "QOS and security in energy-harvesting wireless sensor networks," in *Proceedings of the 7th International Conference on Security and Cryptography (SECRYPT '10)*, pp. 1–10, IEEE, July 2010.

[24] IEEE, *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks Specific Requirements Part*

15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE, 2003.

[25] D. Dolev and A. C.-C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[26] A. Di Mauro, X. Fafoutis, S. Mödersheim, and N. Dragoni, "Detecting and preventing beacon replay attacks in receiver-initiated MAC protocols for energy efficient WSNs," in *Secure IT Systems: 18th Nordic Conference, NordSec 2013, Ilulissat, Greenland, October 18–21, 2013, Proceedings*, vol. 8208 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, Berlin, Germany, 2013.

[27] C. Taramonli, R. J. Green, and M. S. Leeson, "Energy conscious adaptive security scheme for optical wireless," in *Proceedings of the 14th International Conference on Transparent Optical Networks (ICTON '12)*, pp. 1–4, July 2012.

[28] N. Fotiou, G. F. Marias, G. C. Polyzos et al., "Towards adaptable security for energy efficiency in wireless sensor networks," in *Proceedings of the 28th Meeting of the Wireless World Research Forum (WWRF '12)*, pp. 1–6, Wireless World Research Forum, 2012.