Adaptive Multipath Key Reinforcement for Energy Harvesting Wireless Sensor Networks

Di Mauro, Alessio; Dragoni, Nicola

Published in:
Procedia Computer Science

Link to article, DOI:
10.1016/j.procs.2015.08.311

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 63 (2015) 48 – 55

Procedia Computer Science

6th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, EUSPN-2015

Adaptive Multipath Key Reinforcement for Energy Harvesting Wireless Sensor Networks

Alessio Di Mauro[a], Nicola Dragoni[b,*]

[a]*DTU Compute, Technical University of Denmark, Denmark*

[b]*DTU Compute, Technical University of Denmark, Denmark, and Centre for Applied Autonomous Sensor Systems, Örebro University, Sweden*

Abstract

Energy Harvesting - Wireless Sensor Networks (EH-WSNs) constitute systems of networked sensing nodes that are capable of extracting energy from the environment and that use the harvested energy to operate in a sustainable state. Sustainability, seen as design goal, has a significant impact on the design of the security protocols for such networks, as the nodes have to adapt and optimize their behaviour according to the available energy. Traditional key management schemes do not take energy into account, making them not suitable for EH-WSNs. In this paper we propose a new multipath key reinforcement scheme specifically designed for EH-WSNs. The proposed scheme allows each node to take into consideration and adapt to the amount of energy available in the system. In particular, we present two approaches, one static and one fully dynamic, and we discuss some experimental results.

Peer-review under responsibility of the Program Chairs

Keywords: Energy Harvesting Wireless Sensor Networks; Key Management; Multipath Key Reinforcement

1. Introduction

Confidentiality and authentication in sensor networks strongly rely upon cryptographic algorithms to encrypt data and compute cryptographic message authentication codes (CMACs). The encryption algorithms themselves require a sound design and a strong key to work properly. Assuming that the soundness of communication protocols and encryption algorithms holds, in this paper we focus on how to securely generate and distribute keys in regular and energy harvesting (EH) wireless sensor networks (WSNs).

The Importance of Key Management in Sensor Networks. Generally speaking, with the term *key management* we identify a series of processes and techniques connected with handling cryptographic keys in a distributed sensor network. Key generation is the first step. In order to securely communicate, two entities (node-node or node-sink) require a so called *shared key*. These keys should be generated so that only the intended recipients have access to them. Furthermore, depending on the protocol in use, having a single shared key may not be enough. If a security scheme for

* Corresponding author. Address: DTU Compute, Richard Petersens Plads, 2800 Kongens Lyngby - Denmark. Tel.: +4545253731.
E-mail address: ndra@dtu.dk

Peer-review under responsibility of the Program Chairs
doi:10.1016/j.procs.2015.08.311

a sensor network (see for instance[1]) provides both confidentiality and authenticity through a single encryption algorithm, different keys should be used for each security service. This can be achieved in different ways, by generating and sharing additional keys, or by deriving *sub-keys* from a *master key*. If it is allowed for nodes to dynamically join and part the network, these procedures should be repeated to accommodate for the new users. Moreover, if forward and backward security are required, re-keying techniques are needed in order to prevent old nodes to access new messages and new nodes to decrypt previously recorded packets. In addition to this, cryptographic keys have a fixed lifespan, they should not be used to encrypt or authenticate more then a given number of messages. This is usually not a limitation of the key itself, but rather is due to the fact that, depending on the specific scheme, encrypting the same values more than once with the same key could potentially leak unwanted information. In order to prevent this and make each packet unique, additional values are added. However, these values have a fixed length and even by using all the possible combinations there are only so many of them. When the combinations are exhausted, values will repeat. To avoid that, keys should be renewed. Last but not least, if attacks are detected or nodes are compromised, new keys should be distributed once the attack has been dealt with.

Key Management in Energy-Constrained Sensor Networks. While key management is important for the security of any sensor network, it becomes crucial in energy-constrained sensor networks, such as emerging Energy Harvesting - Wireless Sensor Networks (EH-WSNs). EH-WSNs constitute systems of networked sensing nodes that are capable of extracting energy from the environment, such as electromagnetic or piezoelectric energy. Each node is usually equipped with an energy storage unit that acts as an energy buffer. Contrary to traditional WSNs, where the target is to maximize the lifespan of the networks, the foundational design goal of EH-WSNs is *sustainability*, based on the so called *Energy Neutral Operation (ENO)* state[2]. The idea is that if the energy that is harvested is more than the energy that is consumed - over a period of time that can be supported by the energy buffers - then the node operates at a sustainable state and effectively has a continuous lifetime. Sustainability, seen as design goal, has a great impact on the design of the protocols for the sensor network (including key management protocols) as the nodes have to adapt and optimize their behaviour according to the available energy. Since traditional key management schemes do not take energy into account, designing specific key management schemes able to adapt to the energy constraints and requirements of each node becomes crucial to secure energy-constrained networks such as EH-WSNs.

Contribution of the Paper. In this paper we propose a new multipath key reinforcement scheme specifically designed for EH-WSNs. In our scheme, sustainability can be achieved by balancing the number of reinforcement links used by the two nodes willing to establish a new key and the availability of reinforcement neighbors. Both parameters can be adaptively chosen according to the amount of energy available to each node. In particular, we present two different approaches, one static (thresholds for the hysteresis cycle statically defined) and one fully dynamic (sliding threshold window that adapts to the current harvesting rate of a sensor).

Outline of the Paper. The paper follows an incremental structure. Section 2 introduces canonical approaches for distributing and managing cryptographic keys in a sensor network. Section 3 focuses on a specific scheme named multipath key reinforcement. In particular, we discuss how this scheme can provide better security but why it is not a good match for EH-WSNs. Section 4 proposes a new version of this scheme specific for EH-WSNs, taking full advantage of the different energy levels of an EH-WSN by means of an adaptive approach. We conclude the paper by describing some experimental work (Section 4) and summing up the main contribution of the paper (Section 5).

2. Basic Keying Schemes for Sensor Networks

We will now introduce typical keying schemes for sensor networks and highlight which are their main advantages and disadvantages for each of them.

2.1. Single Key

The most simple approach that can be adopted is to use a single key for the system. This has numerous advantages in terms of ease of use. First of all, it is possible to hard-code the key inside a node at the time of creation. In this way each node has the possibility to interact with every other node of the network without having to carry out any

procedure. This scheme requires an almost negligible amount of memory since only a single value must be stored. Furthermore, it is possible for new nodes to join at any time and start communicating with preexisting ones.

Despite these advantages, the single key scheme falls short in terms of security guarantees. First and foremost, it provides a single point of failure. Whenever a node is compromised so it is the security of the entire system. With a minimal effort, an attacker is able to effectively become a fully fledged member of the network, able to send authenticated messages, receive messages addressed to other nodes and decrypt all past and future messages.

As a result, this scheme is usually adopted only for demonstration purposes in security protocols due to its ease of implementation, but is should never be used in real deployment.

2.2. Pairwise Keys

The opposite approach to the single key is to use a different key for each pair of nodes plus the sink. From a security standpoint this scheme offers the best possible security. If an attacker is able to compromise a node and obtain all its keys, only the communications which directly involve this node are compromised. Any other message is secured using a different key to which the attacker has no access. Additionally, it is easy to recover from the loss of a single node, all that is required is to distribute the identity of such node so that every other member of the network can invalidate the specific key used to communicate with it.

Unfortunately, this scheme is extremely costly and it does not meet the scalability requirements of a typical WSN. Given a network with n nodes, the number of necessary keys for the whole system is $n(n-1)/2$. Considering that each node has to maintain a number of keys that increases linearly with the number of nodes $(n-1)$ and that the overall number of keys is quadratic in the number of nodes, this translates to a unsustainable memory consumption for an average node. Assuming a node with 32 KiB of available memory and a cryptographic key of 128 b, the whole memory would be completely filled with keys after only 23 nodes. In addition to that, for each node added to the network new keys must be generated and distributed.

2.3. Random Pre-Distribution

Besides being unsustainable from a memory point of view, the pairwise scheme is also an overkill. Assuming that we want to achieve link based security, a message is encrypted/authenticated and sent to the next hop where it is decrypted and checked. The procedure is then repeated for each hop until the final destination is reached. This implies that not every pair of nodes has to share a key, but that one key is needed only for links through where messages are actually being transmitted. The idea presented in[3] takes this into account and proposes a randomized scheme where a pre-distribution phase assigns a small set of keys to each node in a way that, with high probability, two nodes connected by a link will share a key. Furthermore, a key generation procedure is used to obtain a key for links that do not have one.

In the *pre-distribution phase* a large number of keys P (approximately $2^{17} - 2^{20}$) is generated. Each node then draws k values from P and uses them as its key-ring. Trusted controller nodes are then used to store a mapping between the identity of a node and the identifier of the keys in its key-ring. Finally, each controller node is given the keys shared with each node.

Subsequently, the *shared-key discovery phase* takes place. Here each node discovers which key, if any, the node shares with its neighbours. This is done by having each node broadcast the identity of the keys in the key-ring or through a challenge-response scheme, depending on whether or not the discovery phase should be public or private. Each pair of nodes that are physically in range one another and share a common key define a link.

The third phase, called *path-key establishment*, allows nodes physically in range but not sharing a key to obtain one. To do that, pre-existing keys left unused after the shared-key discovery phase are transmitted to the nodes participating in the path-key establishment.

Once the three phases are terminated, each node in range shares a pair-wise key with its neighbours.

The main property of this random scheme is that the probability that the connectivity graph induced by the network is connected can be made arbitrarily large. The authors use a formula derived by Erdős and Réni[4] in the study of random graphs to show how this can be achieved. Let P_c be the desired probability of the connectivity graph being

connected and p the probability that there exists a link between two nodes. Then, given a number n of nodes, $G(n, p)$ is a random graph whose probability of being connected is

$$P_{\mathrm{C}} = \lim_{n\to\infty} \Pr\left[G(n,p) \text{ is connected}\right] = \mathrm{e}^{\mathrm{e}^{-c}} \tag{1}$$

$$p = \frac{\ln(n)}{n} + \frac{c}{n} \text{ where c is any real constant.} \tag{2}$$

This allows to compute the degree of a node as $d = p(n-1)$ which is the required number of neighbors needed by a node. Furthermore, it is possible to impose connectivity constraints upon the network (required number of neighbors with a shared key) and the key-ring size, and consequently derive the size of the key pool P given a desired probability p' that two nodes share a key. This is obtained from Equation (3)

$$p' = 1 - \frac{\left(1 - \frac{k}{P}\right)^{2(P-k+1/2)}}{\left(1 - \frac{2k}{P}\right)^{(P-2k+1/2)}}. \tag{3}$$

We omit the details of how this is derived and redirect the interested reader to the original paper[3]. Instead we present a short numeric example to help clarifying the concept.

Assume a network with $n = 10,000$ nodes and a desired connectivity probability $P_{\mathrm{C}} = 0.99999$. By inverting Equation (1) we obtain $c = 11.51$ and Equation (2) yields $p = 0.002$. From this we can compute the required degree $d = 20.71$. Hence, if each node has on average d neighbors the network is connected with probability P_{C}. Furthermore, if we fix the key-ring size to $k = 80$ and a probability $p' = 0.5$, we can derive the size of the pool from Equation (3) to be $P \approx 10,000$.

Finally, the increase of the key-ring size is sub-linear in the size of the key pool. For example, if we increase P by a factor of ten, thus making it $P = 100,000$, we have an increase of k of a factor of 3.3 yielding a value of $k = 260$.

3. Multipath Key Reinforcement for Sensor Networks

The random pre-distribution scheme described in Section 2.3 tackles the problem of distributing shared keys within a WSN. However, the keys used therein are simply the keys obtained after the shared-key discovery or the path-key establishment phases. Anyway, all the keys are drawn from a fixed pool and, in order to achieve greater probability of two nodes to share a key while maintaining the size of the key-ring manageable for the memory size of a node, the pool should be kept as small as possible. In contrast, with a small pool there is the concrete possibility that the same key is used on more than one link, therefore if an attacker compromises a node not only all the links that directly involve the node will be compromised, but also any other link that uses one of the keys found in that node. To address this,[5] present a multipath reinforcement scheme whose goal is to strengthen the security by allowing each pair of nodes to use a unique random key. This task can not be solved trivially by generating a sub-key from the already shared key because an attacker that has been recording the key setup messages prior to capturing a node could now decrypt those messages and obtain the new key and therefore access all the messages encrypted with it.

The proposed scheme takes advantage of disjoint paths. Assuming that two nodes u and v want to generate a new key from the existing key k_{shared}, then u chooses j different disjoint paths connecting u to v that were setup during the key distribution phase. For each one of these j paths, u generates a random value x with the same length of the key and sends it to v through the different j paths. After receiving j many values, v computes the new key $k_{\mathrm{reinforced}}$ as

$$k_{\mathrm{reinforced}} = k_{\mathrm{shared}} \oplus x_1 \oplus x_2 \oplus \ldots \oplus x_j. \tag{4}$$

In order to compromise $k_{\mathrm{reinforced}}$, an adversary has to compromise at least one link on all of the j paths. While increasing the number j of paths decreases the probability of the attacker to succeed, the non immediate trade-off is that the longer the path, the higher the probability of an attacker to compromise at least one link. Moreover, computing the disjoint paths is computationally intensive. To solve this, the scheme uses paths of two hops (three nodes), this makes the discovery procedure less intensive and ensures that they are disjoint by construction. A quick way to find such paths is for u and v to exchange their neighbors table and identify the nodes in common.

Assuming ideal communications, i.e., circular communication range with radius r for both transmission and reception, two nodes separated by a given distance have an expected area of overlap of $0.5865\pi r^2$. Hence, the expected number of reinforcing neighbors (the neighbors common to two nodes trying to run the reinforcement scheme) is given by $0.5865p^2n'$, where p is the probability of two nodes to share a key and n' is the number of neighbors of a node. This can also be expressed in terms of the degree of a node as $0.5865\,d^2/n'$.

We now derive the increase of security achieved by the scheme. Let t be the number of links used to reinforce the key and q_{link} the probability that an adversary will compromise a single node. Then the probability that the adversary will compromise the new key is equal to the probability of compromising either one of the hops in the path, minus the probability of compromising both. By applying this to all the t neighbors and including the original link we have

$$q_{\text{reinforced}} = q_{\text{link}}(2q_{\text{link}} - q_{\text{link}}^2)^t. \tag{5}$$

The protocol average overhead can be approximated to 10, while the effort required by the attacker to break a reinforced link for a probability $q_{\text{link}} = 0.1$ translates to an increase of 146 times. The scheme experiences diminishing returns, the higher the probability of a node to be compromised, the lower the effort required by the adversary.

4. Adaptive Multipath Key Reinforcement for Energy Harvesting - Wireless Sensor Networks

We now take a closer look at how the multipath reinforcement scheme can be applied to energy-harvesting wireless sensor networks (EH-WSNs). While the scheme can be run unmodified in this kind of sensor networks, it will not take advantage of the core properties of EH. To address that we present a new adaptive scheme that takes into account the available energy of the reinforcement neighbors.

As we have stressed in Section 1, contrary to WSNs where the target is to maximize the lifespan of the networks, one of the foundational goals of EH-WSNs is sustainability, reaching the energy neutral operation (ENO) state. In our scheme this can be achieved by balancing the number of reinforcement links used by the two nodes willing to establish a new key and the availability of reinforcement neighbors. Both parameters can be adaptively chosen according to the amount of energy available to each node.

Depending on the particular network and to some extent also on the nature of the energy being harvested, different energy situation are likely to be present. We will now describe how the protocol adapts.

4.1. Scheme Description

Let us assume that nodes u and v want to run the reinforcement scheme in order to obtain a new key. We define s_u as the required number of reinforcing neighbors for node u and $k_{u,v}$ as the maximum number of neighbors connecting both u and v, that is the size of the intersection of the key-ring of u and v. The value s_u can be chosen in different ways, for example on a message-per-message basis according to the content of the packet or as a global parameter depending on the size of the network and the required maximum probability that an adversary will compromise a link.

4.1.1. $s_u > k_{u,v}$

If $s_u > k_{u,v}$, then there are not enough available neighbors to run the protocol. One option in this case is to wait for enough nodes to come online. However, given the unpredictability of EH-WSNs this may never realize, and by the time that new nodes have become available, older ones might have run out of power. In this case if a key must be established in a short period of time we fall back to a centralized scheme where both u and v are assisted by the sink node. This protocol in inspired by the well-known Needham-Schroeder protocol[6] and by[7] where the sink plays the role of the trusted third party. Each node is equipped with a unique key, shared with the base station (BS). The protocol will start with u communicating its intention to establish a key with v to the sink node. The sink will generate a new key $e_{u,v}$ for u and v and a token t_u. These values together with the nodes identities are sent to both u and v, each encrypted with their own BS shared key. After that u will send the token and its own identity to v, encrypting them with $e_{u,v}$. The node v will then decrypt this value and compare the identity of u with the one received from the sink. If the two match, v will encrypt the token under $e_{u,v}$ and send it back to u, thus completing the protocol.

This protocol shifts most of the computational burden towards the BS, however it still requires a significant number of messages to be completed and, most importantly, it is not distributed. Each node relies on the BS and a considerable

amount of energy will be spent by the nodes close to it since they will be involved in the majority of the traffic. For this reason this protocol is suitable only when not enough neighbors are available.

4.1.2. $s_u \leq k_{u,v}$

In case $s_u \leq k_{u,v}$, then the protocol can be run if the reinforcing neighbors have enough energy available to participate. This value can be advertised by the nodes themselves. An ideal way to do that is by including this information together with the amount of energy available to the node as part of the MAC[1] protocol, such as in a beacon of a receiver-initiated (RI) MAC protocol[8]. In this way, u will receive regular updates with a minimal overhead. When enough neighbors are available u can greedily choose the d_u ones with the highest amount of energy and start the protocol with them.

A node will choose autonomously whether or not it is able to participate to a run of the protocol and advertise that to its neighborhood. One way to achieve that is by setting a threshold on the energy reservoir above which a node is considered to have stored enough energy to participate. The disadvantage of this approach is that if a node is hovering around the threshold value it might be asked to take place in a run only to find itself without enough energy when the actual ensuing transmission should be performed. To avoid that we define a threshold window $(t_{\text{low}}, t_{\text{high}})$ effectively setting up a comparator with hysteresis. Whenever the energy available to a node is less than t_{low} the node will not participate in the protocol, whereas if the value is above t_{high} the node will consider itself able to participate. If the current available energy falls within the range $(t_{\text{low}}, t_{\text{high}})$ the node will maintain its previous status until one of the other conditions is met. This provides a configurable energy buffer that can be varied according to many parameters such as the typical load of the node, the size of the energy reservoir, the length of a packet, etc.

The $(t_{\text{low}}, t_{\text{high}})$ window can be controlled by varying three parameters. The value of t_{high} will determine how quickly the node will start participating, that is the amount of energy required in order to be considered eligible for the protocol. The value of t_{low} will determine how aggressively the node will participate or how quickly it will transition from the eligible to the ineligible state. The difference $t_{\text{high}} - t_{\text{low}}$ will determine the size of the buffer or how resilient the node is to change its status after a change in its available energy.

4.2. Scheme Evaluation

In order to evaluate different approaches we ran a series of simulations. For this purpose we built a custom tool using the Java programming language. The tool uses the common technique of generating discrete events that are consumed based on a priority function that represents the time elapsed. Each node is assigned a random harvesting rate, the value is sampled from a normal distribution with a mean of 0.015 and a standard deviation of 0.005. To extract pseudo-random samples, the default LCM generation built into Java has been used. The different simulations have been run for a period of 180,000 time-units.

Three arrangements of thresholds have been used, in each of them the threshold is expressed as a portion of a unity energy reservoir. The first arrangement is $t_1 = (0.8, 0.9)$, the window is small in size and located towards the maximum value. This produces a very conservative behavior where in order to become eligible, a node has to be almost fully charged. The second arrangement is $t_2 = (0.1, 0.2)$ in which the window has the same size but its position is significantly lower. In this case a small amount of energy is required for the node to participate in the scheme and it will keep participating until its reservoir is almost completely depleted. The third arrangement is $t_3 = (0.15, 0.85)$ where the two threshold are set at the midway point of the previous arrangements, thus making the window size considerably wider. As a result, this will provide a more stable behavior where the node will maintain its status for longer and transition from one mode to the other when is either considerably charged or almost out of power.

We evaluated these thresholds in three different network scenarios, by varying s and k of $(2, 3)$, $(2, 4)$ and $(3, 4)$, respectively. The simulation continuously runs the reinforcement scheme as the only main task within a node. We let the simulation run for an allotted amount of time while randomly changing the amount of energy harvested by each reinforcing neighbor and keeping track of how many times the protocol was successfully run.

[1] Medium Access Control

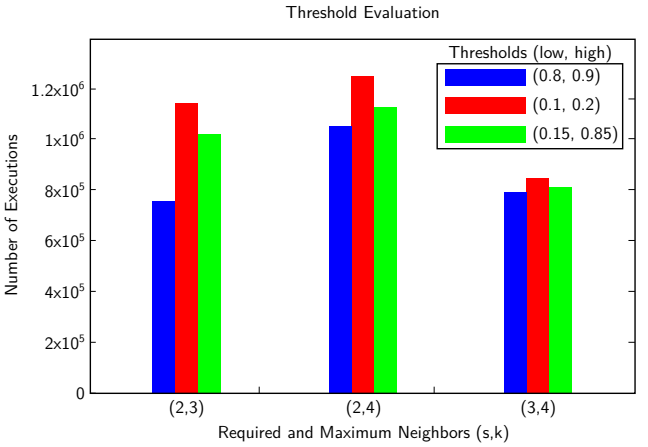

Fig. 1: Multipath reinforcement for EH-WSNs. Here the thresholds for the hysteresis cycle are statically defined.

As shown in Figure 1, the common trend is that the second configuration, which was the most permissive one, always achieves the highest number of runs, whereas configuration number one, being the most conservative, achieves the lowest amount of protocol executions. Finally, the third configuration presents a good compromise on the number of runs by keeping the node active most of the time while not letting it run as low in power as the first scheme. This was indeed the expected behavior.

The three configurations are a good display of how different parameters can be accommodated by the system. Depending on what is the desired energy status of the nodes, the thresholds can be set accordingly. If the network prioritizes security and therefore wants the scheme to be able to run whenever is required, without worrying of the fact that this could cause some of the nodes involved in the protocol to consume all of their stored energy, then a configuration similar to the first arrangement can be used. On the other hand, if we need a network where energy should be mainly used to exchange messages and perform other tasks, while the key reinforcement should be run only when there is some disposable energy, then the second configuration is the most suited. The third configuration is a high resiliency one and can be used for example when there are relatively short and frequent fluctuation in the availability of the scavenged energy source.

If the main energy source were to disappear for a long enough period of time, the first configuration would prevent a node to partake in the protocol almost immediately (as soon as the stored energy started decreasing). The second configuration would instead allow the node to almost run out of energy while still running the protocol. If the third configuration was used in a scenario like this, the node would instead join the protocol only when almost fully charged and therefore be able to sustain a considerable number of executions, but it would not stop right away or exhaust its energy storage when energy becomes unavailable. Once the main source would reestablish itself, the node would simply start charging again without having modified its behavior in the meanwhile.

4.3. Adaptive Sliding Window

Another configuration that we took into consideration has been designed to increase adaptability, thus better suiting EH-WSNs. Here we use a sliding threshold window that adapts to the current harvesting rate of a sensor. The window is allowed to shift up and down, and to shrink or expand depending on the current harvesting rate. The idea is to start in a configuration similar to the first scenario presented before with a small window positioned at the top. As we know, this is a conservative approach and a good starting point for when the node first comes online and has not much energy available. As the harvesting rate and the available energy increase, the window will start moving down and increase in size, reaching a configuration that is halfway between arrangements number two and three. If the harvesting rate becomes negative, the window will revert to its previous state by floating back up and reducing its size.

As shown in Figure 2 the adaptive configuration performs similarly to the second configurations, however, it will prevent a node to run too low on energy by increasing the minimum threshold when there is not enough energy.

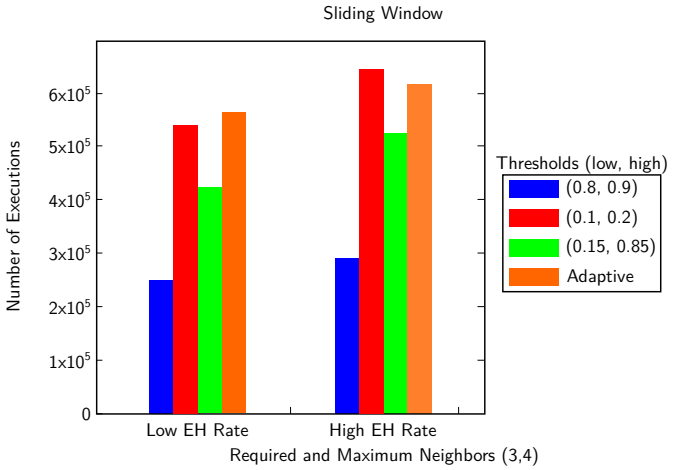

Fig. 2: Multipath reinforcement for EH-WSNs. Both thresholds for the participation of a node move according to the amount of energy available, making a node more likely to participate if it has abundant energy.

Another advantage of this approach is that the nodes that are harvesting more energy will be the ones that will take part in the protocol more often. Furthermore, this concept remains true in an adaptive way, meaning that if the energy source will change in such a way that some nodes will not harvest as much energy, but others will start harvesting more, then the second group will take over the duties of the first one.

As with the other scheme, the initial level of the thresholds and the expansion-contraction rate are system parameters that can be tuned according to the application and the energy source in use.

5. Conclusion and Acknowledgement

In this paper we have proposed a new multipath key reinforcement scheme specifically designed for EH-WSNs. In this scheme, sustainability can be achieved by balancing the number of reinforcement links used by the two nodes willing to establish a new key and the availability of reinforcement neighbors. Both parameters can be adaptively chosen according to the amount of energy available to each node. In particular, we have presented two different approaches, one static (thresholds for the hysteresis cycle statically defined) and one fully dynamic (sliding threshold window that adapts to the current harvesting rate of a sensor). Experimental results by means of simulations have shown the validity of the proposed protocol. The research described in this paper has partially been funded by the IDEA4CPS project.

References

1. Di Mauro, A., Fafoutis, X., Dragoni, N.. Adaptive Security in ODMAC for Multihop Energy Harvesting Wireless Sensor Networks. *International Journal of Distributed Sensor Networks* 2015;**2015**:1–10.
2. Kansal, A., Hsu, J., Zahedi, S., Srivastava, M.B.. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* 2007;**6**.
3. Eschenauer, L., Gligor, V.D.. A Key-Management Scheme for Distributed Sensor Networks. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*. ACM; 2002, p. 41–47.
4. Erdős, P., Rényi, A.. On the Evolution of Random Graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences* 1960;**5**:17–61.
5. Chan, H., Perrig, A., Song, D.. Random Key Predistribution Schemes for Sensor Networks. In: *Proceedings of the 22nd IEEE Symposium on Security and Privacy (SP)*. IEEE; 2003, p. 197–213.
6. Needham, R.M., Schroeder, M.D.. Authentication Revisited. *ACM SIGOPS Operating Systems Review* 1987;**21**(1):7.
7. Zhang, X., Heys, H.M., Li, C.. Energy Cost of Cryptographic Session Key Establishment in a Wireless Sensor Network. In: *Proceedings of the 6th International Conference on Communications and Networking in China (CHINACOM)*. IEEE; 2011, p. 335–339.
8. Fafoutis, X., Di Mauro, A., Vithanage, M.D., Dragoni, N.. Receiver-Initiated Medium Access Control Protocols for Wireless Sensor Networks. *Computer Networks* 2015;**76**(0):55 – 74.