Energy-free Security in Wireless Sensor Networks

Adel Elgaber
Institut FEMTO-ST UMR CNRS 6174
Université de Franche-Comté, France
aelgaber@femto-st.fr

Julien Bernard
Institut FEMTO-ST UMR CNRS 6174
Université de Franche-Comté, France
julien.bernard@femto-st.fr

Yacouba Ouattara
Institut FEMTO-ST UMR CNRS 6174
Université de Franche-Comté, France
youattar@femto-st.fr

Abstract—**Wireless sensor networks are often deployed in open and uncontrolled environments that make them more vulnerable to security attacks. Cryptographic algorithms can be used to protect the data collected by the sensors against an intruder. The cost in terms of energy to provide enough security can be quite large as these algorithms may be very complex. As communication is the main energy consumer, a way to save energy is to use data compression. We propose to measure the impact of the well-known DES algorithm on the energy consumption for various number of rounds and then, we show that energy-free security may be possible. We combine a cryptographic algorithm with a compression algorithm and show through a model that a node can provide security without consuming more energy. The only counterpart is the time for ciphering and compressing. We get some results from experiments on energy consumption of cryptographic and compression algorithms and establish the level of security that can be achieved in various cases, from a single node to a random network.**

Keywords–Wireless sensor networks; security; compression; energy

I. Introduction

A wireless sensor network (WSN) is a specific ad-hoc network with a large number of nodes that have limited energy. These networks are used for collecting information about natural phenomena and other applications. As they are generally deployed in open and uncontrolled environments, wireless sensor networks are more vulnerable to security attacks [1][2][3][4].

Many cryptographic protocols have been proposed to deal with security issues [2][5][6]. They rely on cryptographic primitives like public key cryptographic algorithms or secret key cryptographic algorithms or cryptographic hash functions. The impact on energy consumption of some of these primitives has already been evaluated [7].

As communication is the main energy consumer, a way to save energy in wireless sensor network is to use data compression [8]. Again, the impact on energy consumption of compression algorithms has been evaluated [9][10][11].

In this paper, we first propose to measure the impact of the well-known Data Encryption Standard (DES) algorithm on the energy consumption of a MSP430-based node. Then, our main contribution is to show that energy-free security is possible. We combine a cryptographic algorithm with a compression algorithm and show through a model that a node can provide security without consuming more energy. The only counterpart is the time for ciphering and compressing involving CPU cycles, which is largely less consuming than communication.

In section II, we analyze the related work regarding security and compression algorithms in wireless sensor networks. Then, in section III, we provide experimental results for DES on a MSP430 based node from the Senslab plaform. In section IV, we give an energy consumption model for cryptographic algorithms and compression algorithms and in section V, we show that it may be possible to have energy-free security. In section VI, we use a linear network to achieve better security considering the energy of the whole network. Finally, in section VII, we show that, even on random networks, it is possible to have strong energy-free security with high probability.

II. Related work

Regarding security in general, there are two main families of cryptographic algorithms: public key cryptographic algorithms like RSA or ElGamal and secret key cryptographic algorithms like DES or Advanced Encryption Standard (AES). The advantages of public key cryptography is the availability of authentication and key exchange mechanisms. Public key cryptography is secure and reliable as it is based on strong mathematical theorems. Meanwhile it needs complex arithmetical and logical operations. Strong public key cryptography can affect the lifetime of a node [12][4].

The other solution is to use secret key cryptography. Secret key cryptography relies on simple operations like bit shifting and basic bitwise logical operations (or, and, xor) that can easily be adapted to sensor nodes. Lee et al [7] considered some well-known cryptographic algorithms (AES, RC5, Skipjack, XXTEA) and studied the influence of some parameters (number of rounds, size of the key) on the energy consumption of MicaZ and TelosB sensor nodes. In particular, they show that the energy consumption of RC5 encryption increases linearly with the number of rounds.

In a wireless sensor node, communication is the main energy consumer. An idea to save energy is to apply a compression algorithm before sending data so that the energy used for compression is counterbalanced by the energy saved for communication [8][11]. Capo et al. [9] used a MSP430-based node and measured the consumption of different compression algorithms: S-LZW, Run-Length Encoding (RLE) and K-RLE.

III. Experiments with DES on MSP430

A. Data Encryption Standard (DES)

DES is a symmetric key cryptographic algorithm that was standardized in 1977 and has been used widely since then. DES is based on a Feistel scheme with a 56-bit key and 16

TABLE I. ENERGY CONSUMPTION OF DES WITH VARIOUS NUMBER OF ROUNDS

Rounds	2	4	8	16
Energy (μJ)	21.48	24.75	30.27	40.55

rounds. Each round consists in a fixed set of four operations called Expansion, Key mixing, Substitution and Permutation that operates on 64-bit blocks that are divided in two 32-bit half-blocks [13]. In our experiments, we use a custom implementation of DES in C that can be tuned to reduce the number of rounds.

B. The Senslab testbed

The Senslab platform [14] is an experimental platform for wireless sensor networks. Its aim is to automate the deployment, test, and monitoring of wireless sensor network applications. Each of the four sites of the platform has 256 MSP430-based nodes that can be used to make tests. Each node can be monitored with several probes. The frequency of the measures can be chosen for each experiment. At the end of the experiment, the measures are stored in a simple file for each node.

In our experiment, we used the Senslab platform and we measured the power consumption of a node that was compressing some data with the DES algorithm. The frequency of the measures was set to 100ms.

C. Experiment description

The experiment consists in measuring the power consumed by the DES algorithm on a MSP430-based node of the Senslab platform. For each experiment, we used random input data of $\lambda = 64$ bits. The number of rounds ranges over the values 2, 4, 8 and 16. Each experiment is repeated five times and the average value is given.

D. Results

Table I shows the results obtained in the previous experiment with various number of rounds.

Figure 1 shows the same results on a plot. We observe that the relation between energy consumption and the number of rounds is linear, of the form: $E = E_0 + r \times E_r$ where r is the number of rounds. E_0 is a constant that represents the energy for constant operations in the algorithm, i.e., operations that do not depend on the number of rounds : initial and final permutation, permuted choice 1 (PC1) in the key schedule. E_r is the energy per round. Applying a linear regression on the data of table I, we find $E_0 = 19.149\mu$J and $E_r = 1.348\mu$J/round. The estimation of this linear regression is also shown on figure 1.

As a conclusion of this experiment, we found a linear relationship between energy consumption and the number of rounds for DES. It is of the same kind as the one found in [7] for RC5. These uncorrelated results can lead to a general model for encryption with symmetric cryptographic algorithm. This model is described in the next section.

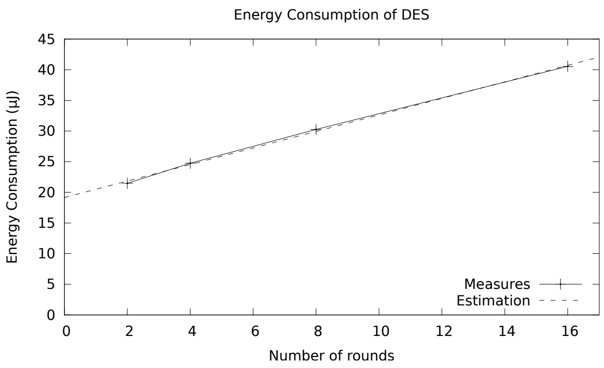

Figure 1. Energy consumption of DES with various number of rounds

TABLE II. VALUES OF E_0^{ENC} AND E_r^{ENC} FOR VARIOUS ENCRYPTION ALGORITHMS

Algorithm	E_0^{enc} (nJ/bit)	E_r^{enc} (nJ/round/bit)
DES	299.2	21.06
RC5 [7]	336.4	173.28

IV. MODELING COMPRESSION AND ENCRYPTION IN WSN

A. Modeling encryption

As seen before, we can model the energy consumption of encryption as:

$$E^{\text{enc}}(\lambda, r) = \lambda \times (E_0^{\text{enc}} + r \times E_r^{\text{enc}}) \tag{1}$$

In addition to the previous experiment, we introduce λ, the length of the input data, in our model. There should be another constant factor that does not depend on the length of the input data, but we assume this constant factor is negligible. For example, in DES, the only operation that does not depend on the number of rounds and the length of the input data is PC1, which is a very simple operation compared to the rest of the algorithm.

Table II shows the values of E_0^{enc} and E_r^{enc} for various algorithms. The DES values are computed from our experiment. The RC5 values are computed from figure 3 in [7]. We took the values of the TelosB node as it is a MSP430-based node, and we summed the ”setup” phase and ”encryption” phase to have the full algorithm. We assumed the length of data to be $\lambda = 8$ bytes $= 64$ bits, by comparing this figure with figure 5 in [7] and taking into account that the word size was divided by two (16 versus 32 respectively).

Figure 2 shows the energy consumption of RC5 that was computed from [7] and the estimation that we did for this algorithm.

RC5 and DES have a similar constant term E_0^{enc} whereas the energy per round E_r^{enc} is much higher for RC5 than for DES with a factor greater than 8. As both algorithms are based on the same basic bitwise operations, this difference can be explained by the implementation and the quality of the measures. The important point is that table II gives us a good idea of the order of energy consumption of a symmetric cryptography algorithm.

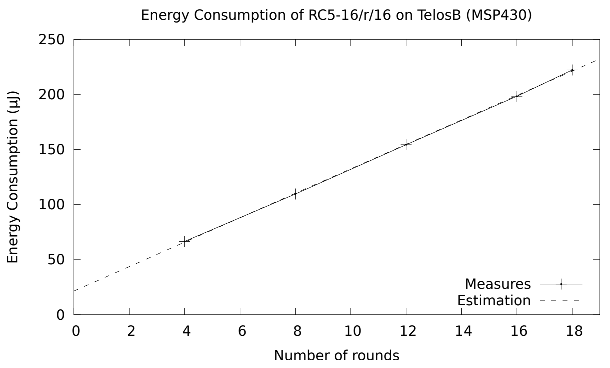

Figure 2. Energy consumption of RC5 with various number of rounds [7]

TABLE III. VALUES OF E_0^{COMP} AND COMPRESSION RATIO α FOR VARIOUS COMPRESSION ALGORITHMS

Algorithm	E_0^{comp} (nJ/bit)	α
RLE	1.325	17%
S-LZW	5.6	53%
K-RLE	2.575	56%

B. Modeling compression

Now, we model compression in the same manner as encryption. We use the results of [9] to model the energy consumption of compression as:

$$E^{\text{comp}}(\lambda) = \lambda \times E_0^{\text{comp}} \tag{2}$$

We assume that the energy consumption for compression algorithms is only proportional to the length of the input data. Generally, compression algorithms do not have a setup phase, they only take decisions according to the input data.

Table III shows the values of E_0^{comp} and the compression ratio α for various algorithms: S-LZW [11], RLE and K-RLE. S-LZW and RLE are lossless compression algorithms while K-RLE is a lossly compression algorithm. All measures are taken from [9] on the same sets of data of length $\lambda = 500$ bytes $= 4000$ bits.

C. Modeling communication

As we want to compare different scenarios with the simple scenario of just sending the data, we need a communication model. We take the communication model from [15]:

$$E^{\text{trans}}(\lambda) = \lambda \times (E_0^{\text{trans}} + \epsilon \times d^2) \tag{3}$$

In this model, E_0^{trans} is the electrical energy and is set to 50nJ/bit, ϵ is the transmit amplifier and is set to 100pJ/m^2/bit, and d is the distance to the receiving node.

V. ANALYSIS OF DIFFERENT SCENARIOS WITH COMPRESSION AND ENCRYPTION

In this section, we examine several scenarios using compression and encryption and the models described in (1), (2) and (3).

- *Scenario T*: in this scenario, we only consider the transmission of λ bits of input data.
- *Scenario CT*: in this scenario, we consider the compression of λ bits of input data with a compression ratio of α that is then transmitted.
- *Scenario ET*: in this scenario, we consider the encryption of λ bits of input data that is then transmitted.
- *Scenario CET*: in this scenario, we consider the compression of λ bits of input data with a compression ratio of α that is then encrypted and transmitted.

There is no need for a fifth scenario with encryption followed by compression and then by transmission as it would consume more energy than scenario CET.

Our goal is to show that scenario CET can consume as much energy as scenario T which would provide energy-free security.

A. Energy for the different scenarios

The energy consumption for scenario T is given by:

$$\begin{aligned} E^{\text{T}}(\lambda) &= E^{\text{trans}}(\lambda) \\ &= \lambda \times (E_0^{\text{trans}} + \epsilon \times d^2) \end{aligned} \tag{4}$$

The energy consumption for scenario CT is given by:

$$\begin{aligned} E^{\text{CT}}(\lambda) &= E^{\text{comp}}(\lambda) + E^{\text{trans}}((1-\alpha) \times \lambda) \\ &= \lambda \times (E_0^{\text{comp}} + (1-\alpha) \times (E_0^{\text{trans}} + \epsilon \times d^2)) \end{aligned} \tag{5}$$

The energy consumption for scenario ET is given by:

$$\begin{aligned} E^{\text{ET}}(\lambda, r) &= E^{\text{enc}}(\lambda, r) + E^{\text{trans}}(\lambda) \\ &= \lambda \times (E_0^{\text{enc}} + r \times E_r^{\text{enc}} + E_0^{\text{trans}} + \epsilon \times d^2) \end{aligned} \tag{6}$$

The energy consumption for scenario CET is given by:

$$\begin{aligned} E^{\text{CET}}(\lambda, r) &= E^{\text{comp}}(\lambda) + E^{\text{enc}}((1-\alpha).\lambda, r) + E^{\text{trans}}((1-\alpha).\lambda) \\ &= \lambda \times (E_0^{\text{comp}} + (1-\alpha) \times (E_0^{\text{enc}} + r \times E_r^{\text{enc}} + \\ &\quad E_0^{\text{trans}} + \epsilon \times d^2)) \end{aligned} \tag{7}$$

Table IV shows the energy consumptions with $\lambda = 64$ bits, $r = 8$ rounds and $d = 25$m in the different scenarios. $\lambda = 64$ bits is a typical size for a physical scalar data like temperature. $r = 8$ rounds is rather weak for DES and RC5. $d = 25$m is a typical distance in wireless sensor networks. We observe that, as expected, with any choice of algorithms, scenario CT consumes less energy than scenario T that consumes less energy than scenario CET that consumes less energy than scenario ET.

Table V shows the energy consumptions with $\lambda = 64$ bits, $r = 16$ rounds and $d = 75$m in the different scenarios. In this case, the number of rounds is $r = 16$, which is the maximum for DES and which is nearly the recommended number of rounds for RC5. The distance has been extended to 75m. We note that the energy consumption of scenario CET is

TABLE IV. ENERGY CONSUMPTIONS (IN μJ) WITH $\lambda = 64$ BITS, $r = 8$ ROUNDS AND $d = 25$M

Algorithms	E^{T}	E^{CT}	E^{ET}	E^{CET}
DES + RLE	7.200	6.061	37.132	30.904
DES + S-LZW	7.200	3.742	37.132	17.810
DES + K-RLE	7.200	3.333	37.132	16.503
RC5 + RLE	7.200	6.061	117.449	97.567
RC5 + S-LZW	7.200	3.742	117.449	55.559
RC5 + K-RLE	7.200	3.333	117.449	51.842

TABLE V. ENERGY CONSUMPTIONS (IN μJ) WITH $\lambda = 64$ BITS, $r = 16$ ROUNDS AND $d = 75$M

Algorithms	E^{T}	E^{CT}	E^{ET}	E^{CET}
DES + RLE	39.200	32.621	79.914	66.414
DES + S-LZW	**39.200**	18.782	79.914	**37.918**
DES + K-RLE	**39.200**	17.413	79.914	**35.327**
RC5 + RLE	39.200	32.621	238.168	197.765
RC5 + S-LZW	39.200	18.782	238.168	112.298
RC5 + K-RLE	39.200	17.413	238.168	104.959

less than the energy consumption of scenario T, with the DES algorithm combined with S-LZW or K-RLE. In the other case, the compression algorithm does not have a good enough compression ratio (RLE), or the encryption algorithm consumes so much that it cannot be counterbalanced by compression (RC5). These figures, with realistic parameters, show that it is possible to have energy-free security but we need a more precise condition.

B. Energy-free security

In this section, we try to precise the previous result, i.e., we try to compute the maximum number of rounds $r_{\max}$ for various values of d and the given compression algorithms. The following theorem gives the computation of $r_{\max}$:

Theorem 1: Security is free if and only if:

$$r \leq \underbrace{\frac{\alpha \times (E_0^{\mathrm{trans}} + \epsilon \times d^2) - (1-\alpha) \times E_0^{\mathrm{enc}} - E_0^{\mathrm{comp}}}{(1-\alpha) \times E_r^{\mathrm{enc}}}}_{= r_{\max}(\alpha, d)}$$

The proof is straightforward, it directly comes from (4) and (7) with the condition that $E^{\mathrm{CET}}(\lambda, r) \leq E^{\mathrm{T}}(\lambda)$. We note that the condition does not depend anymore on the length of the data which is normal because, in each scenario, the energy is proportional to the length of the data.

Table VI shows $r_{\max}(\alpha, d)$ for the three compression algorithms and the DES algorithm, with d varying from 25m to 100m. We observe that for distance of 25m, security can not be free whatever the compression algorithm is, i.e., $r_{\max}(\alpha, d) < 0$. The RLE algorithm do not compress enough and can never provide free security.

The results for S-LZW and K-RLE are quite close. For a distance of 50m, the maximum number of rounds is 1 and 3 respectively, which provides no security at all as there exists some easy known attacks on DES. For a distance of 75m, as already seen, the full DES with 16 rounds can be used for free with both compression algorithms. For a distance of 100m, Triple-DES, that has 48 rounds and provides strong security, can be used for free in the case of K-RLE.

Table VII shows $r_{\max}(\alpha, d)$ for the three compression algorithms and the RC5 algorithm, with d varying from 25m to 100m. RC5 consumes more energy than DES and the results for RC5 are not very good. Even for a distance of 100m, the maximum number of rounds is 4 and 5 for S-LZW and K-RLE respectively, which does not provide any security. The solution in this case is to optimize the implementation of RC5 or to find a better compression algorithm.

TABLE VI. $r_{\max}(\alpha, d)$ WITH THE DES ALGORITHM

Algorithms	25m	50m	75m	100m
RLE	–	–	–	–
S-LZW	–	1.291	18.024	41.450
K-RLE	–	3.645	22.531	48.970

TABLE VII. $r_{\max}(\alpha, d)$ WITH THE RC5 ALGORITHM

Algorithms	25m	50m	75m	100m
RLE	–	–	–	–
S-LZW	–	–	1.976	4.823
K-RLE	–	0.228	2.524	5.737

VI. ANALYSIS WITH A LINEAR NETWORK

In this section, we try to improve the previous results considering a linear network. Our idea is that the energy consumed on the sending node can be counterbalanced globally over the network by the savings of the other nodes, due to the size of the compressed data. This could improve the security of the data while still competing with scenario T.

A. Model

We use a linear network of $n+1$ nodes, the first node that generates data and n relays in the multi-hop communication to the base station. Each node only communicates with its closest neighbors that are at distance d. The last node communicates with the base station that is at distance d too.

On this linear network, we examine the global energy in scenario T and scenario CET. In each scenario, the data is sent by the first node, then received n times and transmitted n times until the base station.

We still use the communication model from [15] for receiving:

$$E^{\mathrm{recv}}(\lambda) = \lambda \times E_0^{\mathrm{recv}} \tag{8}$$

E_0^{recv} is the electrical energy and is set to 50nJ/bit.

The energy consumption for scenario T with a linear network is given by:

$$\begin{aligned} E_{\mathrm{net}}^{\mathrm{T}}(\lambda, n) &= E^{\mathrm{trans}}(\lambda) + n \times (E^{\mathrm{recv}}(\lambda) + E^{\mathrm{trans}}(\lambda)) \\ &= \lambda \times (n \times E_0^{\mathrm{recv}} + (n+1) \times (E_0^{\mathrm{trans}} + \epsilon \times d^2)) \end{aligned} \tag{9}$$

The energy consumption for scenario CET with a linear network is given by:

$$\begin{aligned} E_{\mathrm{net}}^{\mathrm{CET}}(\lambda, r, n) &= E^{\mathrm{CET}}(\lambda, r) + n \times (E^{\mathrm{recv}}((1-\alpha).\lambda) + \\ &\quad E^{\mathrm{trans}}((1-\alpha).\lambda)) \\ &= \lambda \times (E_0^{\mathrm{comp}} + (1-\alpha) \times (E_0^{\mathrm{enc}} + r \times E_r^{\mathrm{enc}} + \\ &\quad n \times E_0^{\mathrm{recv}} + (n+1) \times (E_0^{\mathrm{trans}} + \epsilon \times d^2))) \end{aligned} \tag{10}$$

TABLE VIII. $r^{\mathrm{NET}}_{\max}(\alpha, 25, n)$ WITH THE DES ALGORITHM AND A LINEAR NETWORK

Algorithms	$n = 1$	$n = 2$	$n = 5$	$n = 10$	$n = 15$
RLE	–	–	–	2.615	10.517
S-LZW	–	8.653	34.756	78.262	121.767
K-RLE	2.134	11.955	41.416	90.518	139.620

TABLE IX. $r^{\mathrm{NET}}_{\max}(\alpha, 25, n)$ WITH THE RC5 ALGORITHM AND A LINEAR NETWORK

Algorithms	$n = 1$	$n = 2$	$n = 5$	$n = 10$	$n = 15$
RLE	–	–	–	0.103	1.064
S-LZW	–	0.837	4.010	9.297	14.585
K-RLE	0.045	1.238	4.819	10.787	16.754

Now we can state an extension of theorem 1 for a linear network and compute $r^{\mathrm{net}}_{\max}(\alpha, d, n)$:

Theorem 2: Security is free in a linear network of $n + 1$ nodes if and only if:

$$r \leq \underbrace{\frac{n.\alpha.E_0^{\mathrm{recv}} + (n+1).\alpha.(E_0^{\mathrm{trans}} + \epsilon.d^2) - (1-\alpha).E_0^{\mathrm{enc}} - E_0^{\mathrm{comp}}}{(1-\alpha) \times E_r^{\mathrm{enc}}}}_{=r^{\mathrm{net}}_{\max}(\alpha,d,n)}$$

B. Results

Table VIII shows $r^{\mathrm{net}}_{\max}(\alpha, d, n)$ for the three compression algorithms and the DES algorithm, with $d = 25$m and n varying from 1 to 15. We observe that with only one-hop before the base station, security is free with the K-RLE compression algorithm, even if the number of rounds provides very weak security. For $n \geq 5$, the maximum number of rounds for S-LZW and K-RLE exceeds the number of round for DES, and for $n \geq 10$, it exceeds the number of rounds for Triple-DES. This shows that very strong security can be achieved for free on a wide network.

Table IX shows $r^{\mathrm{net}}_{\max}(\alpha, d, n)$ for the three compression algorithms and the RC5 algorithm, with $d = 25$m and n varying from 1 to 15. In this case, the situation is better than in the experiment with a single node, but the level of security is not as strong as the level for DES. For $n \geq 15$, the maximum number of rounds is 14 and 16 for S-LZW and K-RLE respectively, which a little less than the recommended 18-20 rounds for good security. Once again, the implementation of the algorithm must be improved in order to achieve better results.

VII. Analysis with random networks

In this section, we compute $r_{\max}$ on random networks. We focus on DES and K-RLE as it's the best combination of a compression algorithm and an encryption algorithm that we have.

A. Experiment

The difficulty in this experiment is to choose an application to make the measures. We decided to test a simple routing application on a square area of width w. The network is composed of n nodes uniformly distributed on the area. Two nodes can communicate if their distance is less than $0.4 \times w$ so that there is, on average, half of the nodes in the neighborhood of each node, whatever the width of the area.

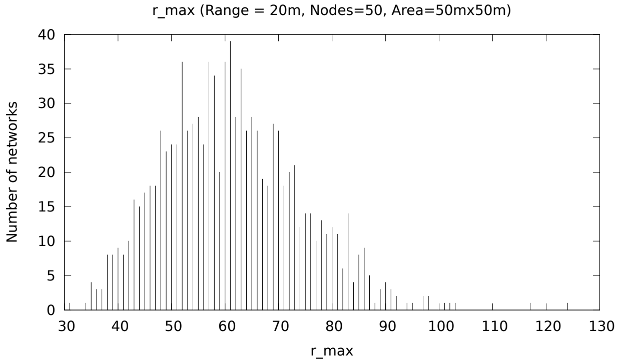

Figure 3. Distribution of $r_{\max}$ for n =50 nodes and w = 50m

Among the n nodes, 20 nodes are chosen to be sources of messages of size $\lambda = 64$bits. Those messages are routed to a sink which is placed at the coordinates $(0.9 \times w, 0.9 \times w)$ thanks to a shortest path algorithm which takes into account the square of the distance between each node (as the energy for transmitting a message is proportional to the square of the distance). Then, each source sends a message to the sink along the chosen path.

To compute $r_{\max}$ for a given network, we first compute the energy E_T that is consumed for scenario T. Then, we compute the energy that is consumed for scenario CET with $r = 0$ rounds, which is necessarily less than the E_T (scenario CET with $r = 0$ rounds is similar to scenario CT). Then, the number of rounds is increased until the energy is more that E_T which means we have reached $r_{\max}$.

This experiment is repeated on 1000 different random network for each value of (n, w).

B. Results

Figure 3 shows the distribution of $r_{\max}$ for n =50 nodes and $w = 50$m. We observe that this distribution is not symmetric and is quite wide so that the computation of an average is not very relevant. That's why we decided to compute the first decile of the measures, i.e., the value of $r_{\max}$ that divide the data set in 10% of low values and 90% of high values. In this case, the first decile is 45 which means that for a random network with our simple routing application, taking DES with 45 rounds (nearly Triple DES) and K-RLE is energy-free with a probability of 0.9.

This result is not a surprise as it can be compared to the results of table VIII. The range of the network is a little shorter in the case of random networks (20m), but the average number of hops from the sources to the sink must be high enough so that the energy for encryption is counterbalanced by the savings along the paths.

Table X shows the computation of the first decile for many values of (n, w). This table shows that in any case, it is possible to have strong energy-free security on a random network. We observe that, for a fixed w, $r_{\max}$ increases sub-linearly w.r.t. n. Adding more nodes on the area make paths shorter, but not short enough so that the gain in energy can bring many more rounds. We also observe that for a fixed n, $r_{\max}$ increases over-linearly w.r.t. w. In this case, the distances

TABLE X. First decile of $r_{\max}$ for a network of n nodes on a square area of width w

w n	50m	100m	200m	300m	400m
50	45	52	72	102	141
100	76	84	97	119	148
150	98	108	121	138	163
200	101	125	137	152	173

between nodes is increased and the gain in energy can be used to do many more rounds.

VIII. Conclusion

We show that it is possible to provide strong and free security thanks to a careful choice of compression algorithm and cryptographic algorithm. We provide models for the energy consumption of compression algorithms and encryption algorithms. Our models are derived from experiments done by ourselves or found in the literature, with a MSP430-based node.

It would be interesting to implement these algorithms on real nodes and check that security is really free. The consumption of the transmission comes from a model that is not derived from real experiments so an extension of this work could be to verify this model with various radio chips.

References

[1] D. Vu and D. K. Vu, “Wireless sensor network architecture and its security challenges,” Master’s thesis, California State University, Sacramento, 2010.

[2] D. Martins and H. Guyennet, “Security in wireless sensor networks: a survey of attacks and countermeasures,” International Journal of Space-Based and Situated Computing, vol. 1, no. 2, 2011, pp. 151–162.

[3] H. Saxena, C. Ai, M. Valero, Y. Li, and R. Beyah, “DSF - A Distributed Security Framework for Heterogeneous Wireless Sensor Networks,” in Military Communications Conference, 2010-Milcom 2010. IEEE, 2010, pp. 1836–1843.

[4] G. Gaubatz, J. Kaps, E. Ozturk, and B. Sunar, “State of the art in ultra-low power public key cryptography for wireless sensor networks,” in Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on. IEEE, 2005, pp. 146–150.

[5] X. Ren and H. Yu, “Security mechanisms for wireless sensor networks,” International Journal of Computer Science and Network security (IJCSNS), vol. 6, no. 3, 2006, pp. 155–161.

[6] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, “SPINS: Security protocols for sensor networks,” Wireless networks, vol. 8, no. 5, 2002, pp. 521–534.

[7] J. Lee, K. Kapitanova, and S. Son, “The price of security in wireless sensor networks,” Computer Networks, vol. 54, no. 17, 2010, pp. 2967–2978.

[8] N. Kimura and S. Latifi, “A survey on data compression in wireless sensor networks,” in Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, vol. 2. IEEE, 2005, pp. 8–13.

[9] E. Capo-Chichi, H. Guyennet, and J. Friedt, “K-RLE: A new data compression algorithm for wireless sensor network,” in Sensor Technologies and Applications, 2009. SENSORCOMM’09. Third International Conference on. IEEE, 2009, pp. 502–507.

[10] F. Marcelloni and M. Vecchio, “A simple algorithm for data compression in wireless sensor networks,” Communications Letters, IEEE, vol. 12, no. 6, 2008, pp. 411–413.

[11] C. Sadler and M. Martonosi, “Data compression algorithms for energy-constrained devices in delay tolerant networks,” in Proceedings of the 4th international conference on Embedded networked sensor systems. ACM, 2006, pp. 265–278.

[12] K. Piotrowski, P. Langendoerfer, and S. Peter, “How public key cryptography influences wireless sensor node lifetime,” in Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks. ACM, 2006, pp. 169–176.

[13] N. (NIST), “FIPS 46-3, Data Encryption Standard (DES).”

[14] C. Burin des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noël, and J. Vandaele, “Using SensLAB as a first class scientific tool for large scale wireless sensor network experiments,” NETWORKING 2011, 2011, pp. 147–159.

[15] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. IEEE, 2000, pp. 8020–8029.