On optimizing energy consumption: An adaptative authentication level in wireless sensor networks

Martin Peres
LaBRI
Université de Bordeaux
Email: martin.peres@ensi-bourges.fr

Mohamed Aymen Chalouf
LaBRI
Université de Bordeaux
Email: chalouf@labri.fr

Francine Krief
LaBRI
Université de Bordeaux
Email: krief@labri.fr

***Abstract*—Nowadays's authentication methods are essentially based on cryptography or any other kind of secret information. These methods are particularly efficient but they are, in certain cases, very power-hungry. As energy is a scarce resource in wireless sensor networks, we propose a new approach that consists in utilising the physical properties of the transmission medium in order to calculate, when receiving a frame, a confidence rating that we attribute to the source of the frame. This information is important in order to decide if it is necessary to authenticate the source using asymmetric cryptography. In the case where sensors are static and communications are slightly perturbed by the environment, the proposed rating can add another layer of control which enables sensors to check the origin of messages. This paper will also study how collaboration between the network's nodes further enhances detection of malicious or third-party nodes.**

I. Introduction

Wireless sensor networks (WSNs) are used in several fields such as monitoring air-quality, seismic activity, forest fires, structural integrity of a building and also area intrusion detection.

Sensors are small wireless network nodes characterised by their very low processing power and storage capacity (both ROM and RAM). They are also characterised by very limited energy resources. Despite these constraints, sensors are usually expected to operate over a long period (several years), be secure and serve data with a relatively low latency.

Autonomy in such networks is important because there could be hundreds of nodes located in remote or poorly-accessible places. For instance, regularly changing the batteries of these nodes represents a high maintenance cost. So, the higher the autonomy is, the lower the maintenance cost would be.

Security of wireless sensor networks has been a hot research topic for years. This can be explained by the necessity of guaranteeing some security properties such as data reliability and auditability by the administrator. The security guarantees are important because of the expensive nature of wireless sensor networks.

The following study has been conducted as part of a research project called DIstributed Applications and Functions Over Redundant Unattended Sensors (DIAFORUS), funded by the French National Research Agency (ANR). The goal of this project is to develop a framework for in-network reasoning and cooperation in heterogeneous wireless sensor networks. Node cooperation combined with adaptive security mechanisms will help achieve overall energy saving.

In section II, we introduce the state of the art concerning security in wireless sensor networks. Section III describes our proposition which is based on the physical properties of the transmission medium and a collaboration between sensors in order to introduce another authentication layer. Then, section IV studies the scope and limits of this proposition. The last section evaluates the proposed collaboration protocols.

II. State of the art

We can distinguish several security needs:

- Authentication: No node can pretend to be another one;
- Confidentiality: No third party can read the sensors' data;
- Integrity: No third party can modify packets on-the-fly;
- Data freshness: No network node can replay old packets;
- Availability: Both the network and data are accessible.

Most of these needs are usually met using cryptography.

A. Cryptography

1) Symmetric cryptography: A cryptographic algorithm is said symmetric when the same key is used to encrypt and decrypt data.

There are several symmetric cryptography algorithms. One of the most famous is AES [1], which is using a 128bit key. It is the successor of DES and triple-DES which are now considered unsafe. It takes around 3 ms to encrypt a block (128 bits) using AES-128 on a mica2DOT [2] sensor.

2) Asymmetric cryptography: A cryptographic algorithm is said asymmetric when two different keys are needed. One of them should be public while the other one should be kept strictly private.

When encrypting with a key, the other is needed to decrypt. Choosing with which key a message should be encrypted depends if the message should be signed or confidential.

For instance, if Bob wants to transmit secret information to Alice, Bob should encrypt his message using Alice's public key. As only Alice knows her private key, she is the only one who can decrypt the message. This is confidentiality.

If Alice wants to send a public message and prove she is the author, she can encrypt the message or the hash of the message with her private key. This way, when Bob receives

Alice's message, he can decrypt it using Alice's public key and be sure it was encrypted with Alice's private key. This is a digital signature. Asymmetric cryptography is often used to authenticate two network nodes.

The most common asymmetric cryptography algorithm is RSA [3]. It uses 1024 bit keys. However, another algorithm that is particularly interesting in the context of wireless sensor networks is the Elliptic Curves Cryptography (ECC). This shrinks the size of the needed key from 1024 to 160 bits [4] and by so, lowers the needed memory to store keys. Another advantage of ECC over RSA is the reported lower energy consumption [5].

Processing power on wireless sensor nodes is usually so low that using asymmetric encryption induces latencies of up to several seconds [5]. It also consumes more than symmetric cryptography by several orders of magnitude [6].

B. Security in wireless sensor networks

It is important to differentiate two kinds of communications, one-to-one communication achieved through unicast and the one-to-many, achieved through multicast.

1) Security for unicast communications: Security in one-to-one communications is usually achieved by establishing a secure session between two nodes. This session would be based on a secret key shared by the two nodes. This key would then be used to encrypt the communications between the two nodes using symmetric cryptography.

Generating such a secret key can be achieved using the Diffie-Hellman protocol [7] which has now become an RFC [8]. It allows two nodes to mutually generate a symmetric key over an insecure network.

However, the Diffie-Hellman key exchange protocol is not authenticated. It is possible to add an authentication layer by mutually issuing challenges using asymmetric cryptography and a Public-Key Infrastructure [9]. Another possibility could be the use of an ECC-based Diffie-Hellman (ECDH) [10].

It is also possible to generate a secret key by deriving an already-shared key, this is called key derivation. For example, a network could be given a unique big random identifier that would be shared by all the nodes of this network. A symmetric key for communication between two nodes in the network could be calculated by using a cryptographic hash function on the concatenation of the network identifier with the name of the node having the lowest ID, a coma and the name of the other node. Equation 1 describes the generation of a symmetric key for the communication between node "1" and node "3" of the network identified by the ID "NetworkRandomID".

$$key = SHA1(NetworkRandomID+, \text{“}1,3\text{”}) \qquad (1)$$

This technique allows nodes to generate communication keys when it is required without needing a Diffie-Hellman protocol. It increases the number of communication keys compared to having a single network-wide key, and it makes it more difficult for an attacker to infiltrate the network. However, if a network node is corrupted/malicious, it can basically impersonate every other node since the only information needed to compute the session key between two nodes is the "NetworkRandomID". Not all key distribution systems share this security flaw [11][12].

When using key derivation techniques, nodes are not required to remember all the secret keys already generated. This is an interesting specificity that can be useful when memory constraints on a node are too tight such as when a node needs to communicate with hundreds of nodes.

Once a secret key has been established between two nodes, SNEP [11] can be used to guarantee data confidentiality, authentication, integrity and freshness.

It has also been demonstrated that using IPSec over 6LoWPAN is possible [13]. This enables the use of an end-to-end secure communication between a sensor node and an Internet host. Thus, the gateway between the sensor network and the Internet does not have to be trusted anymore.

2) Security for multicast communications: In the case of a one-to-many communications, security could be provided through the use of several protocols. The choice of the protocol depends on user's goals.

Confidentiality can only be achieved by encrypting a message with a key shared by all the recipients of the message. This means that there should be a symmetric key associated with each "secure" multicast address.

Integrity and authentication can be achieved by signing messages like detailed in Figure 1. However, this technique is resource hungry and usually impractical [5] without hardware acceleration [14].

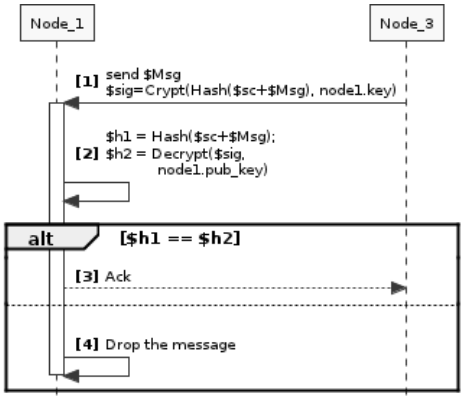

Figure 1. Classic packet signature

Another way of achieving integrity, authentication and data freshness is to use μTESLA [11] at the cost of a higher memory usage. Indeed, μTESLA requires nodes to generate and store a chain of hash.

A chain of hash starts with a unique key that is hashed. The subsequent values of the chain are calculated by hashing the previous value. Knowledge of the 10th hash, implies the knowledge of all the subsequent hashes. However, finding the 9th from the 10th is impossible because of the irreversible property of a cryptographic hash.

μTESLA's security is based on a chain of hash. A node first broadcasts all the messages to send, encrypted with a hash that has never be disclosed. Then, it publishes the hash that is required to decrypt the previously sent messages. Any node can then verify that the sender is the same as it used to be by hashing the hash and by comparing it to the last one disclosed as shown by Figure 2.

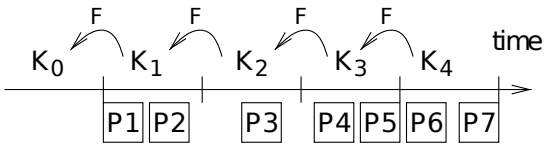

Figure 2. μTESLA's messages and key divulgation. Extracted from SPINS [11]

μTESLA requires an important memory space. Indeed, sensors are required to generate and store a personal chain of hashes and store incoming messages until the key is disclosed to read them later.

C. Key storage security

Basic sensor nodes, such as the mica2DOT, have no physical protection preventing an attacker from tampering with them.

Unless a sensor node uses a Trusted Platform Module (TPM) [15], a smart-card or any other physical protections, keys stored in them should be considered unsafe and accessible by an attacker. Indeed, keys would just sit in the flash memory of the node. As this memory is accessible using the JTAG port, it should not be considered safe.

Even if a sensor uses a safe storage system, it may still be vulnerable if the keys have to be loaded in RAM. Indeed, it is possible to access data stored in RAM by freezing the sensor and doing some forensic later on.

However, this technique requires time and money. Thus, we will consider this a non-issue because the administrator should be able to detect that a node is missing and remotely update the network keys.

D. Trust and reputation

Network security alone is not enough to guarantee the validity of the information as an attacker can tamper with one or many nodes through a physical or a network access [16].

Indeed, wireless sensor networks are not protected against remote buffer-overflow exploitation. This is even simpler as physical access to the nodes may be quite easy and an attacker could download a compiled version of the software using the JTAG port of a sensor node to pentest the software.

Moreover, in the case of a heterogeneous network where sensors from different companies work together, collaboration should be rewarded and selfish behaviours should be punished [17].

The usual solution to these problems is to compute a reputation and trust rating for all the surrounding nodes. These ratings are then used to decide whether to accept or refuse data coming from one or several sensors [17].

These ratings are usually generated using the nodes' behaviour concerning the packet routing. Nevertheless, it is also possible to control the nodes' behaviour concerning other protocols from the physical to the application layers of the ISO model.

E. Outdoor localisation and security

Localising surrounding nodes in a wireless network can be achieved through the use of RSSI (Received signal strength indication) and collaboration (triangulation).

The RSSI is affected by the surrounding environment and by the direction of the antenna of both the receiving and emitting node. This makes accurate localisation impossible in both indoor and outdoor environments because of their unpredictability when the environment characteristics are not known [18].

However, RSSI's variability in outdoor environments is more limited and roughly constant with distance. This make outdoor localisation possible with a roughly constant precision [19].

Localisation can also be used for security and access control in WLANs [20].

III. Proposition

This paper introduces a new information source for the reputation/trust management along with a distributed loose authentication protocol of incoming frames.

First, we introduce the concept of distributed triangulation from a local perspective. Then, we describe how nodes can collaborate to achieve a real distributed triangulation.

A. Distributed triangulation

Today's authentication methods are essentially based on a shared secret and/or cryptography. These methods are efficient but, we want to specify a mechanism which would rely less on cryptography and would instead rely on cooperation between nodes to lower the overhead incurred by security.

This is possible by basing the security on some other immutable rules, physics. Indeed, the communication of two static and in direct-sight sensors is only attenuated by the distance separating them and by the atmosphere. This can be the case in forest fires detection scenarios.

As this attenuation should vary slightly, it is possible to use the reception power of a frame to evaluate the probability for this frame to really come from the presumed source node. Through experience, a sensor can associate a minimal and maximal reception power to each of its surrounding nodes.

The weaknesses and scope of this proposition are explained in section IV.

This concept is illustrated by Figure 3. Nodes 1, 2 and 3 are legitimate nodes of the WSN. Attackers 1, 2 and 3 are trying to impersonate Node 3 when sending messages to Node 1. Node 1 can tell Attacker 1 is not Node 3 because the reception power is out of the usual range. However, it cannot detect the attackers 2 and 3.

The attacker 2 can only be detected by the Nodes 2 and 3. As a mean of collaboration, we propose that nodes detecting

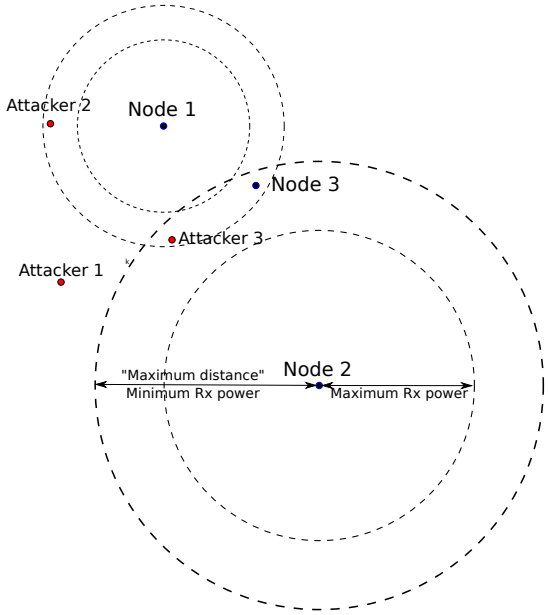

Figure 3. Distributed triangulation: Attackers 1, 2 and 3 try to impersonate Node 3 to send data on the network

any anomaly in the reception power of a frame should alert the recipient of this frame.

This collaboration can be rewarded by increasing the reputation of a node reporting the issue.

When a node decides that it does not trust a received frame, it can react to the attack by lowering the reputation of the node that was supposed to have sent the frame. Since this node is currently "under attack", it is normal to trust it less than other nodes. Moreover, in the case of data aggregation, lowering the reputation of a node also decreases the impact of the data sent by this node.

Another benefit from using distributed triangulation is the early detection of hardware malfunctions. Indeed, a high variability of power reception can mean that the electrical contact between the antenna and the transceiver is imperfect. The network administrator could then be alerted sooner of this problem when the reputation of this node is dropped.

Another answer to an untrusted frame can be to ask for a digital signature of the frame. If this signature is valid, the reputation of the source node should be increased while the one of the nodes who reported the problem should be decreased. If the signature is invalid, the reputation of the node that reported the anomaly should be increased. This protocol tends to favour collaboration but also punishes nodes requesting other ones to perform unnecessary computations.

A more drastic answer to an untrusted frame could be dropping the aforementioned frame. This solution can be useful in the case of a node flooding the network. The best answer to this problem is to drop the frame as the more it is processed, the more energy is consumed.

Lastly, if a node detects it is being impersonated, it should inform the network through a signed broadcast message. This allows surrounding nodes to be aware of the situation and to decrease the reputation of the node temporarily. This benefits the impersonated node as its reputation will return to its original level when the attack stops. It also benefits the surrounding nodes that now know an attack is going on. They can be more suspicious. Figure 4 shows a situation where an attacker tries to impersonate Node_3 when communicating with Node_1. Technically, Node_3 needs to send a signed message telling that it is not the sender of the previous message.

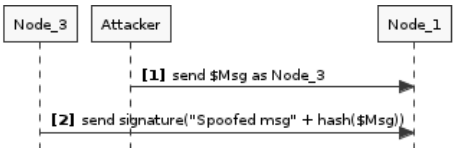

Figure 4. Distributed triangulation: Spoofing alert

B. Use case 1: Unicast communications

In this use case, we will detail how distributed triangulation can help increasing security in unicast communications.

Distributed triangulation can improve security of unicast communications when the key distribution system is unsecure. In this type of network, there is protection against eavesdropping by nodes that are not part of the network but a malicious node belonging to the network can eavesdrop and impersonate any other node. This is the case of some key-derivation-based key distribution systems. As nodes would not be installed in a very small area, the use of distributed triangulation is sufficient to check the origin of a frame.

As an example, we propose a protocol that can be used to further secure unicast communications.

When a node receives a frame, it should check the reception power. If this reception power is out of the usual range, the receiving node should ask the emitting node to transmit the signature associated to this message.

To ensure data freshness, a random number called salt can also be transmitted. It is the emitting node's task to concatenate the message and the salt, hash it, then encrypt this hash with his private key and send the result which can be checked by the destination node.

The salt does not need to be generated from a safe random source. Indeed, we just need to generate a unique number that has never been used as a salt before. Hence, a big-enough number is sufficient as the probability for this number to have already been used before would be quite low.

Thus, a signature will always look different because of one property of a hash function, the avalanche effect. This property means that changing even a single bit of a message changes completely its hash. It would then be impossible for an attacker to replay an old message. By so, it could not pretend to be the node it tries to impersonate because the salt will always be different even if the message does not change.

It is also important to sometimes check the signature even though the reception power is in a valid range in order to force other nodes to implement this protocol. Figure 5 repre-

sents a sequence diagram of a protocol illustrating distributed triangulation from a local point of view.

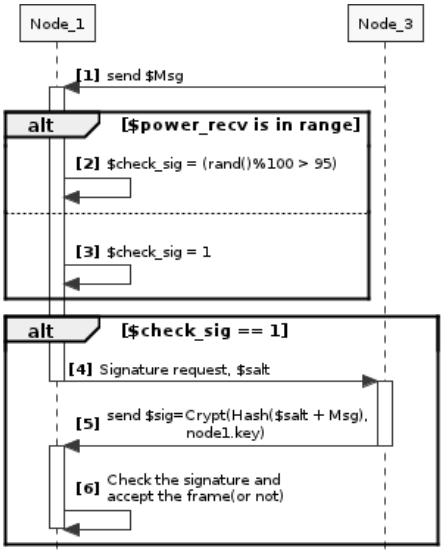

Figure 5. Distributed triangulation and unicast: a local point of view

Figure 6 shows how nodes can collaborate to improve detection accuracy. The proposed authentication solution is based on a digital signature but any other mean of authentication is possible.

When receiving alerts, it is up to Node_1 to ask a signature to Node_3 or not. The decision can be based on the reputation of Node_3 and the reputation of the nodes that reported an alert. It can also be motivated by the number of alerts received or the remaining energy in Node_1.

This proposition is helpful in the case of a WSN composed of sensors with a limited crypto processor that automatically encrypts the outgoing data and decrypt the incoming data using a single key. In this case, the network security comes cheap and may be considered sufficient to beat external attacks. The only remaining risk from a network point of view would be that a malicious node impersonates another one. This can be controlled using distributed triangulation.

C. Use case 2: Multicast communications

Unlike unicast, a multicast communication has no single recipient. This means we need to update the protocol proposed in Figure 5 and 6.

In the case of Figure 6, instead of sending an alert to Node_1 (message [3]), Node_X should send the alarm to all the original recipients. In other words, Node_X should send the alert to the multicast address.

We also need to define who should ask for a signature. We suggest that it should be up to Node_1 to decide what to do. A convention could be set-up stating, for instance, that if more than 3 nodes send an alert, the emitting node should send the message's signature. If the emitting node (Node_3 in Figures 6 and 7) does not abide by this protocol, it will be up

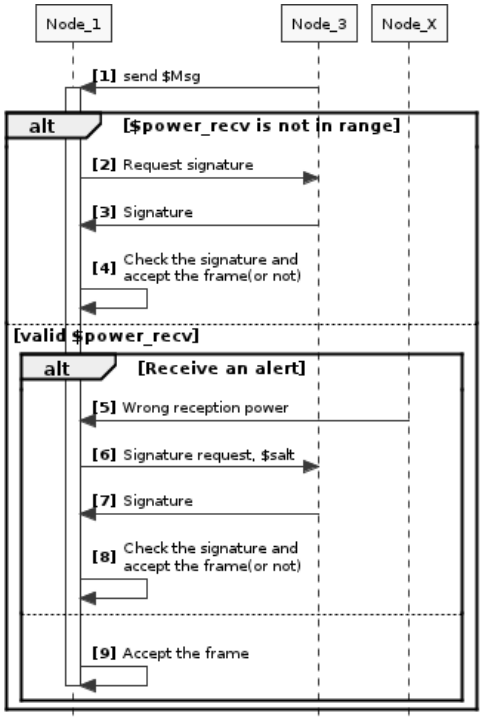

Figure 6. Distributed triangulation and unicast: the collaborative point of view

to each recipient to accept or reject the frame and to decrease the reputation of the emitting node if needed.

The proposed protocol, without reputation management, is summarised by Figure 7.

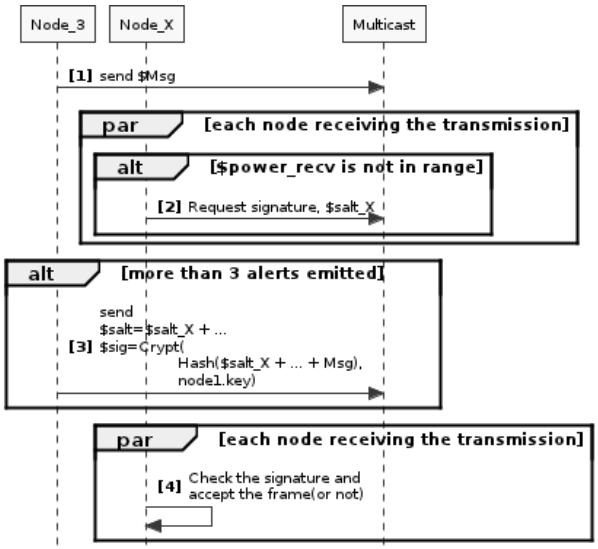

Figure 7. Distributed triangulation and multicast: the collaborative point of view

Distributed triangulation in a multicast communication really increases security, contrarily to the use of a key shared by all the multicast recipients. Besides, as described in Section V, our protocol requires less memory than μTESLA because it

does not need to store all the messages sent by a node while waiting for the key to decipher them.

IV. Scope and inherent weaknesses

It is impossible to accurately locate a node using triangulation. This is due to the variable attenuation of the communication medium. This attenuation is partly due to the surrounding environment and the variability of air's moisture levels.

However, the more nodes in the network, the better the precision will be. Indeed, errors will tend to counteract each others up to a certain extent. Unfortunately, it is impossible to differentiate two sensors situated 10cm away from each others.

Distributed triangulation works better with collaboration. Using a directional antenna, an attacker could send frames to a single node without having the other nodes of the network noticing this emission. In this case, a node would only rely on its reception power which is not sufficient. An attacker could then impersonate another node if he/she is able to calculate the usual attenuation.

However, we may consider that using symmetric keys to encrypt messages sent on the network brings sufficient security because it denies access to third-party nodes. In this case, directional antennas are not a security threat.

Distributed triangulation can be used by static sensor networks and works best when sensors are always in clear-sight. This is because it lowers the noise and perturbation brought by the environment. However, if the environment around isn't dynamic, this system can still be used if sensors are not in a radio-clear-line-of-sight as, from a local perspective, each node will always receive the same signal strengh when receiving a frame from a given sensor.

From a security perspective, it is safer if sensors cannot be located precisely in order to make it harder for attackers to impersonate sensors by using directional antennas.

V. Evaluation

Evaluation of energy costs and calculation time is done on a mica2DOT because of the already existing surveys for it [6][21][22], allowing us to fill both Table I and Table II.

Transmission time with AES-128 encryption is equal to the encryption time + the transmission time + the decryption time.

Type	time	power
Encryption of 128 bits, AES-128	2.69 ms	38 μJ
Decryption of 128 bits, AES-128	3.22 ms	45.6 μJ
Hashing 512 bits, SHA-1	16.125 ms	3.02mJ
Signing ECDSA-160	1650ms	27.23 mJ
Verifying ECDSA-160	3220ms	53.96 mJ

Table I
THE COST OF CRYPTOGRAPHIC OPERATIONS ON A MICA2DOT

A transmission cost is evaluated by adding the transmission cost of a 6loWPAN header to the transmission cost of the message's body. Transmission energy cost is defined by Equation 2 while the reception one is defined by Equation 3. Latency induced by this transmission is given by Equation 4.

Field	Value
LR06 battery	8640 J
Effective data rate	12.4 kbps
Energy to transmit	59.2 μJ/byte
Energy to receive	28.6 μJ/byte
Time to transmit	0.66 ms/byte
Energy to transmit (AES-128)	97.2 μJ/byte
Energy to receive (AES-128)	74.2 μJ/byte
Time to transmit min (AES-128)	1.03 ms/byte
Max. emitted on an LR06	145.94 MB
Max. received on an LR06	302.1 MB
Max. emitted with AES-128	88.89 MB
Max. received with AES-128	116.44 MB
6LoWPAN [23] header size	25 bytes
Energy to transmit (AES-128)	2.43 mJ
Energy to receive (AES-128)	1.86 mJ
Time to transmit min (AES-128)	25.75 ms
Energy to transmit an SHA-1 (AES-128)	1.94 mJ
Energy to receive an SHA-1 (AES-128)	1.48 mJ
Time to transmit min (AES-128)	20.6 ms
ECDSA-160 transmission cost	2.368 mJ
ECDSA-160 reception cost	1.144 mJ
Time to transmit	41.2 ms

Table II
TRANSMISSION COSTS ON A MICA2DOT

$$t\_energy = 2.43mJ + msg.length * 97.2\mu J \quad (2)$$

$$r\_energy = 1.86mJ + msg.length * 74.2\mu J \quad (3)$$

$$t\_latency = 25.75ms + msg.length * 1.03ms/byte \quad (4)$$

There are two cases that need to be distinguished, the cost when all the reception powers are nominal; and the cost when one or several nodes detect an abnormal power reception.

A. Cost during normal operation

Cost in normal operation is composed by:

- 1) Listening to every frame's header;
- 2) Updating the reception power values for each surrounding node;
- 3) Comparing the reception power with usual ones.

The cost of 1) is null because we already have to listen to every frame's header as it is also a requirement for receiving communications.

The cost of 2) depends on the way we store the reception power for each node. The minimum we can do is storing the minimal and maximal reception power for each node. If stored in dB, a byte is sufficient for storing a reception power. This means that the cost in RAM is linear as shown by Equation 5. Updating the values in the table is considered free in the case of an indexed access. It can also be linear if the table elements need to be iterated to find the right value to update.

$$cost = 2 * sizeof(byte) * nr\_surrounding\_nodes \quad (5)$$

Similarly to the cost of 2), the cost of 3) depends on the way the table is stored. At best, the cost should be considered free. At worse, it should be considered linear if the table elements needs to be iterated to find the right value to compare with.

In the case of this study, we will not consider the energy and latency cost during normal operation given how insignificant it should be. This also means that security provided by distributed triangulation is considered free in normal operations.

B. Cost on an abnormal reception power

The global cost of an abnormal reception power depends on the number of nodes in the network. In this simulation, we consider 3 different network sizes, 2, 10 and 100 nodes. We also consider the network is not under attack and a probability of occurence can be associated to abnormal reception powers. This probability is the second parameter of our simulation. It spreads the cost of false positives across all the cases where the system works as expected.

The study is based on the network described by Figure 3, when attacker 2 is trying to impersonate Node_3.

The protocol used for the study is the multicast one, described by Figure 7. We will reference messages in these figures with this notation [X] where X is the X^{th} message exchanged since the beginning.

We consider the content of Message [1] to be 32 bytes long. The transmission cost of Message [1] is the transmission cost of the 6loWPAN's header + the transmission cost of the message's body. The message is then sent to the multicast address. So, the reception cost is incurred on all the nodes except the emitter. Using Equations 2, 3 and 4, we can calculate the costs associated to this transmission. The total emission cost is 5.54 mJ while the reception one is 4.23 mJ. Transmission takes 58.71 ms.

Message [2] can be sent by any node that received the multicast communication. We consider that the size of the salt number should be lower than 5 bytes. So, we can assume that the total size of the message should be under 10 bytes. The message is sent to the multicast address so the reception cost is incurred on all the nodes except the emitter. The energy emission cost is 3.4 mJ while the reception one is 2.6 mJ. Transmission takes 36.05 ms.

Message [3] is created and sent by the original emitter. Computing a signature requires hashing the message to be signed and then encrypting it using the node's private key. Since the message's size is under 512 bit, we can hash it in a single round. The final message contains the hash of the original message (20 bytes), the message type (5 bytes) and the signature (40 bytes). Signing the message (hash + asymmetric cryptography) takes 1666.13 ms and costs 30.25 mJ.
The message is sent to the multicast address so the reception cost is incurred on all the nodes except the emitter. There is an emission energy cost of 39 mJ and a reception one of 6.7 mJ. Transmission takes 1733 ms.

Finally, checking the signature costs 53.96 mJ to each recipient and takes 3220 ms. Globally, the operation takes around 5 s.

Method / Nr of nodes	2	10	100
ECC-signature	77.23	562.87	5419.27
AES-128 encryption	0.087	0.3452	2.9192
DT p=1	105.66	705.6	6705
DT p=0.1	10.566	70.56	670.5
DT p=0.001	0.10566	0.7056	6.705
DT p=0.0001	0.010566	0.07056	0.6705

Table III
THE GLOBAL ENERGY COST (MJ) INCURRED BY THE USE OF DISTRIBUTED TRIANGULATION COMPARED TO SIGNING ALL MESSAGES AND SYMMETRIC CRYPTOGRAPHY

Nr of nodes	2	10	100
Energy consumption	9.63	47.7	428.4
ECC-signature overhead	801.9%	1178%	1265%
AES-128 encryption	0.9%	0.7	0.7%
D.T. overhead (proba=1)	1097%	1479%	1565%
D.T. overhead (proba=0.01)	10.97%	14.79%	15.65%
D.T. overhead (proba=0.0001)	0.1097%	0.1479%	0.1565%

Table IV
THE GLOBAL ENERGY COST (MJ) INCURRED BY THE EMISSION OF A MULTICAST 32 BYTES-LONG MESSAGE. COMPARAISON WITH THE OVERHEAD OF DISTRIBUTED TRIANGULATION, ECC-SIGNATURE AND AES-128 ENCRYPTION

The global cost for the network can be calculated by adding all the transmission energy costs and all the reception ones for each recipient node.

Table III summarises the overhead evaluation of distributed triangulation on a wireless sensor network in the worst case scenario. This scenario happens when every node reports an anomaly in the reception power. It shows energy costs in mJ depending on the probability for the occurrence of an unusual reception power and the number of nodes in the multicast.

Results presented in Table III should be compared to the original cost of the transmission without distributed triangulation as shown in Table IV. Distributed Triangulation outperforms the use of ECC-based signature of every message if the probability of receiving an unusual power is 0.1. Such a probability should be equal or lower than 0.0001 in order for distributed triangulation to be consume less than classic AES-128 symmetric encryption.

Distributed Triangulation's cost depends heavily on frame's probability to have an unusual reception power. The overhead of distributed triangulation can be up to 1500% or down to 1.5% depending on this probability. Assessing attenuation's variability is necessary in order to evaluate the usefulness of this method.

VI. FUTURE WORKS

A. Real-life experiment

As mentioned before, efficiency of distributed triangulation heavily depends on the attenuation's variability.

We are planning to experiment it in different environments such as:

- A parking lot: This is possibly the noisiest environment from a signal point of view (multipath and occlusion);

- In the University: Human and car movements, multipath due to the surrounding buildings;
- Next to a tram and a road: An open space with man-made radio perturbations;
- On a football pit: No multipath, non-noisy environment.

The experiments would be done at different distances:

- 1m: Test how surrounding radio perturbations/noise alter the signal, very low perturbation due to multipath;
- 50m: Test radio noise and medium distance multipath;
- 300m: Test radio noise and long distance multipath.

We expect the signal to be slightly altered by surrounding radio emissions. However, we think that signal strength's variability will be greater in percentage at smaller distances than at longer distances because of multipath.

B. Signal signature

This paper demonstrates that collaboration between nodes can help detecting attackers by relying on the physical layer of the ISO model. Power reception is not the only information we can get from the physical layer. In complex environments, it is also possible to extract a signal signature because of the phenomenon called multipath.

Relying on signal signature instead of power reception means it would be even harder for attackers to impersonate another sensor. Moreover, the more complex the environment is, the harder it would be to impersonate someone else.

The problem with this signature is that it has to be resistant to small perturbations like cars driving nearby. We think people researching on watermarking should be able to overcome this issue.

VII. Conclusion

Even though it is impossible to identify a node by anything else than cryptography, it is however possible to significantly decrease the cost of frame authentication by using the introduced technique of distributed triangulation. Distributed triangulation works best when communication medium's attenuation varies slightly and when sensors cannot be easily and precisely located. A real-life evaluation of communication's attenuation variability will be assessed in future works. Besides, instead of using only reception power as a mean of authentication, we are also assessing the use of other properties of the physical layer of the ISO model such as the perturbation incurred by multipath to compute a signal signature.

Acknowledgements

This work is part of research project called DIstributed Applications and Functions Over Redundant Unattended Sensors (DIAFORUS) [24], funded by the French National Research Agency (ANR).

References

[1] J. Daemen, J. Daemen, J. Daemen, V. Rijmen, and V. Rijmen, "Aes proposal: Rijndael," 1998.

[2] Crossbow, "Mica2dot," www.cmt-gmbh.de/Mica2dot.pdf, 2011, [Online; accessed 30-May-2011].

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, pp. 120–126, February 1978.

[4] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery, "On the security of 1024-bit rsa and 160-bit elliptic curve cryptography," Cryptology ePrint Archive, Report 2009/389, 2009.

[5] E. oliver Blaß and M. Zitterbart, "Towards acceptable public-key encryption in sensor networks," in *in The 2nd International Workshop on Ubiquitous Computing (ACM SIGMIS*, 2005, pp. 88–93.

[6] 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 Third IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 324–328.

[7] W. Diffie and M. E. Hellman, "New directions in cryptography," 1976.

[8] E. Rescorla, "Diffie-Hellman Key Agreement Method," RFC 2631 (Proposed Standard), Internet Engineering Task Force, Jun. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2631.txt

[9] S. Blake-Wilson and A. Menezes, "Authenticated diffie-hellman key agreement protocols," 1998.

[10] C. Lederer, R. Mader, M. Koschuch, J. Großschädl, A. Szekely, and S. Tillich, "Energy-efficient implementation of ecdh key exchange for wireless sensor networks," in *3RD WORKSHOP ON INFORMATION SECURITY THEORY AND PRACTICE — WISTP 2009*. Springer Verlag, 2009, pp. 112–127.

[11] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wirel. Netw.*, vol. 8, pp. 521–534, September 2002.

[12] P. Kotzanikolaou, D. D. Vergados, G. Stergiou, and E. Magkos, "Multilayer key establishment for large-scale sensor networks," *Int. J. Secur. Netw.*, vol. 3, pp. 1–9, December 2008.

[13] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6lowpan with compressed ipsec," in *7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2011)*, Barcelone, Span, Jun 2011.

[14] G. Gaubatz, J. peter Kaps, and B. Sunar, "Public key cryptography in sensor networks - revisited," in *In 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004*, 2004, pp. 2–18.

[15] T. C. Group, "TPM Main Specification," http://www.trustedcomputinggroup.org/resources/tpm_main_specification, 2011, [Online; accessed 05-July-2011].

[16] W. Zhang, S. Das, and Y. Liu, "A trust based framework for secure data aggregation in wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, sept. 2006, pp. 60 –69.

[17] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei, *Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, 2008, ch. Reputation and Trust-Based Systems for Ad Hoc and Sensor Networks.

[18] J. Andersen, T. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *Communications Magazine, IEEE*, vol. 33, no. 1, pp. 42 –49, jan 1995.

[19] M. L. Sichitiu, V. Ramadurai, and P. Peddabachagari, "Simple algorithm for outdoor localization of wireless sensor networks with inaccurate range measurements." in *International Conference on Wireless Networks*, W. Zhuang, C.-H. Yeh, O. Droegehorn, C.-T. Toh, and H. R. Arabnia, Eds. CSREA Press, 2003, pp. 300–305.

[20] D. B. D. Faria, "Scalable location-based security in wireless networks," Ph.D. dissertation, Stanford, CA, USA, 2007, aAI3242547.

[21] G. de Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 580–585.

[22] 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*, ser. SASN '06. New York, NY, USA: ACM, 2006, pp. 169–176.

[23] J. Hui and D. Culler, "Ipv6 in low-power wireless networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1865 –1878, 2010.

[24] LaBRI, "Diaforus," https://diaforus.labri.fr/doku.php, 2010, [Online; accessed 05-July-2011].