ENERGY-EFFICIENT SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS USING FREQUENCY HOPPING AND PERMUTATION CIPHERING

Ismail Mansour, Gérard Chalhoub and Michel Misson
Clermont Université/LIMOS CNRS, Complexe scientifique des Cézeaux, 63177 Aubière cedex, France

Keywords: Wireless sensor network, Security, Permutation ciphering, Frequency hopping, Energy efficiency.

Abstract: The security aspect of wireless sensor networks has taken the attention of numerous researchers in the past several years. It has recently been proven that public keys are now feasible in wireless sensor networks but still consume a lot of processing time and memory. In this paper we propose the use of public keys based on ECC to exchange symmetric keys that will be used to encrypt critical information. In addition, we propose a time segmentation approach that enables the use of frequency hopping time slotted communications. Nodes secretly exchange frequency hopping sequences that enable them to fight against jamming and eavesdropping. We use permutation ciphering technique to protect the information exchanged between nodes.

1 INTRODUCTION

Wireless sensor networks are more and more deployed for various applications including home monitoring, health, industrial, military, etc. It is known that wireless networks are easy to attack because of the nature of the shared medium which makes it relatively easy for intruders to eavesdrop, tamper or inject data into the network. Sensor nodes are known to have limited computation, storage, power and transmission capacities, but attackers are not necessarily using the same technology to launch their attacks.

The security level required might vary from one application to another according to the importance of the information that is being exchanged. In this paper, we present a security mechanism using Elliptic Curve Cryptography (ECC) to establish a secured communication based on frequency hopping in an energy efficient context. The main contribution of this paper is the proposition of a secured protocol for critical applications supporting a dynamic topology and hundreds of nodes, and where security is a requirement and energy efficiency is a necessity. We present a cryptographic system based on ECC that enables us to exchange secret frequency hopping sequences and encrypt exchanged data using a permutation ciphering technique.

2 RELATED WORK

In this section, we go through the related work in the field of security in wireless sensor networks including key management and frequency hopping.

2.1 Key Management in Wireless Sensor Networks

In order to provide security in wireless sensor networks, communication should be encrypted and authenticated. The main issue is how to set up secret keys between nodes to be used for the cryptographic operations, which is known as the key agreement. Cryptographic systems are generally based on two techniques: symmetric keys and asymmetric keys.

In symmetric key systems, nodes encrypt and decrypt messages using the same cryptographic key. These systems are known for their light weight cryptographic operations compared to public key systems, thus they were the first to be considered for wireless sensor networks where computational resources are scarce. One of the most known symmetric key systems that were proposed for wireless sensor networks is SPINS (Perrig et al., 2002).

ZigBee is one of the most known standards for wireless sensor networks. In the ZigBee-2007 specifications (Zigbee, 2008), the protocol proposes a

Mansour I., Chalhoub G. and Misson M. (2011).
ENERGY-EFFICIENT SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS USING FREQUENCY HOPPING AND PERMUTATION CIPHERING .
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems*, pages 277-282
DOI: 10.5220/0003371302770282

cryptographic mechanism based on 128-bit symmetric keys to secure communications between devices. The most common key agreement for these systems is key pre-distribution prior to network deployment. After deployment, only the nodes sharing the same key are able to communicate. Most of the key management protocols for symmetric systems, lack resilience to an observer that is able to listen to the discovery process during which shared keys are exchanged.

Public key systems use a pair of keys: a private key which is kept private and a public key which is known by anyone. Data encrypted with one key is decrypted with the other, which avoids sharing the same key for encryption and decryption. The most common criticism on using public key cryptography in sensor networks is its computational complexity and communication overhead. Nevertheless, recent work such in (Hong, 2008) has been done to prove that public key systems are feasible for wireless sensor networks where authors prove that public-key cryptography based on ECC is very viable on small wireless devices. ECC is based on the problem of logarithm discrete. The main attraction of ECC over competing technologies such as RSA and DSA is the use of smaller parameters but with equivalent level of security (Lopez and Dahab, 2000). For example, 160-bit ECC key is equivalent to 1024-bit RSA key. The performance of ECC depends mainly on the efficiency of finite field computation and fast algorithms for elliptic scalar multiplications.

2.2 Frequency Hopping as a Security Mechanism

Frequency hopping is a known technique that enhances robustness against interferences in wireless networks. By frequency hopping we mean the ability to change the frequency that is being used on the physical layer from one transmission to another. Benefits of this technique in wireless sensor networks have been proven in (Watteyne et al., 2009). Frequency hopping can also be used to hide the communication against eavesdropping and this by making it difficult on an intruder to guess the frequency hopping sequence.

In (Jones et al., 2003), the authors propose a holistic approach to secure wireless sensor network by using symmetric key encryption and secured frequency hopping sequences to protect and hide the communications. Frequency hopping sequences are updated every now and then in such a way that an intruder has not the time to guess the sequence. A sink node divides the network into sectors and wedges, allocating each sector a frequency hopping sequence and an encryption key. In this paper, the applications covered are only sink oriented data collection, and the network size depends on the coverage area of the sink node.

Standardizations tendency (ISA, 2009; HART, 2008) towards frequency hopping is a proof that it is an important in wireless communication in order to make links more robust to interferences. It is also a proof that the time synchronization needed for the slot allocation is feasible as discussed in (Kerkez et al., 2009). By adopting a secret frequency hopping sequence like in (Jones et al., 2003), this mechanism can enhance security for wireless sensor network.

3 OUR PROPOSITION

A lot of attacks (Kavitha1 and Sridharan, 2010) like sinkhole, black hole, spoofing and tampering can be avoided using public key encryption infrastructure that offers the means to ensure entity authentication, source authentication, and data integrity and confidentiality. Our proposition is based on ECC encryption, frequency hopping and permutation ciphering.

We use ECAES encryption for ensuring data integrity and authentication, and we use session keys (symmetric keys) for data encryption in addition to permutation ciphering. Session keys and permutation ciphers are exchanged using ECAES encryption. To fight against jamming and interferences, we use a frequency hopping mechanism based on shared secret frequency hopping sequences. Each node in the network negotiates, in a secured manner, a frequency hopping sequence with all the nodes that it is allowed to communicate with. This negotiation allows nodes to choose the best frequencies as observed by the nodes.

We consider a hierarchical topology as depicted on figure 1, where the node *S* is the sink of the topology and the default destination for the traffic. We suppose that the sink has more storage capacities than other nodes. Hierarchical topologies are known to be more convenient for energy efficiency in wireless sensor networks (Akkaya and Younis, 2005). Nodes are grouped into stars, each star has one central node that we call router and several end-devices. Routers have more processing and memory capacities then end-devices and they constitute about 10 % of the network size. End-devices represent the sensors and actuators of the network. End-devices are only allowed to communicate with the router of the star to which they belong. The creation of stars is done during the network deployment phase. Routers are allowed to communicate with each other but not with end-devices that are not associated to them. This clustering tech-

nique is similar to the one used in IEEE 802.15.4 and MaCARI (Chalhoub et al., 2008b). We suppose that routers are able to aggregate data on the network layer level.

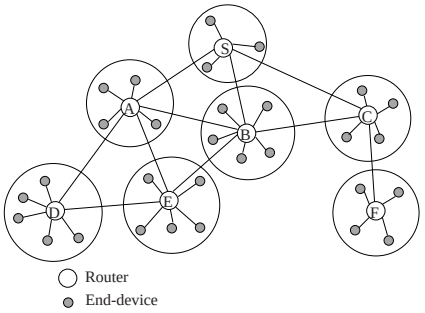

Figure 1: Network architecture. The network is decomposed in stars. Each star is under the control of one router. Sensor and actors are associated to a router and belong to only star.

3.1 Time Synchronization for Frequency Hopping and Energy Efficiency

As discussed earlier, frequency hopping helps against interference and jamming attacks. In this paper, we use a subset of the available frequencies to hop between them for two consecutive transmissions. In order to establish a time synchronization that allows nodes to hop from one frequency to another, we use a synchronization mechanism based on beacon propagation as used in the MaCARI protocol (Chalhoub et al., 2008b; Chalhoub et al., 2008a).

Time is divided into cycles that start with a synchronization period during which a beacon frame containing synchronization information is propagated along the topology by the routers. The propagation starts at the sink and is done in a topological order indicated inside the beacon to avoid collisions between beacons. The sink allocates for each router a slot in $[T_0;T_1]$ during which the beacon has to be propagated to reach all the nodes of the network. This beacon contains time segmentation information allowing nodes to know when to be active and when to save energy.

Figure 2 depicts the time segmentation during a cycle. In $[T_1;T_2]$, routers are allocated time intervals during which they communicate with the end-devices and then relays the collected data to the next router. The time interval allocation is done during $[T_0;T_1]$ by the sink according to the traffic size generated in each star. Once information is collected from end-devices, routers exchange messages until the information reaches its final destination. Each router has to keep track of the slots used for routing by neighbor routers.

In order to allow new nodes to join the network in a dynamic manner, we use CSMA/CA during $[T_2;T_3]$. When a new node wants to join the network, it is not yet known by the sink and does not have slots during which it can communicate. This ensures a dynamic network where new nodes (cluster heads or end-devices) are able to join even after the deployment phase. Nodes sends join request destined to the sink during $[T_2;T_3]$. Any router of the network is able to receive the join request and relay it towards the sink through other routers, this is explained in 3.2.

The cycle ends with an inactive period, $[T_2;T_3]$. The duration of the inactive period depends on the duty cycle of the nodes and the reactivity requirements of the application. This time segmentation allows all the nodes of the network to save energy by switching to sleep mode when they are not concerned by the communication during a given time slot.

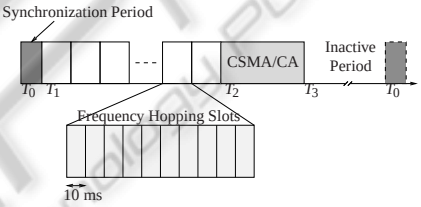

Figure 2: Time segmentation for the frequency hopping mechanism.

During the join phase (the phase during which nodes are associated to the network), end-devices and routers exchange the set of frequencies that they would like to use for reception, they also inform the sink of the traffic they will generate at the application layer level. The traffic information enables the sink to allocate enough time slots for the nodes. That is, every entity is able to scan all possible frequencies and decide which of these frequencies are less busy so it is able to have a better reception rate using them. Frequencies that seem good for the sender may not seem good for the receiver.

At the end of the association phase, the router indicates to the end-device the slots during which it will communicate with it, the frequency to be used during each slot and the communication direction (whether it is a reception slot or a transmission slot). This exchange is encrypted using a session key created between the router and the end-device which is detailed in 3.2. The duration of a communication slot is 10 *ms*, which is enough time to send a frame of maximum size and wait for its acknowledgment. This is the slot duration of TSMP, WirelessHart and ISA100(Pister and Lance, 2006; HART, 2008; ISA, 2009).

The order in which the frequency set is hopped is specified by the router during the join phase. It is then updated every l cycles, where l is maximum number of cycles tolerated before compromising the frequency hopping order. So every l cycles, end-devices receive a new order from the router encrypted using the pairwise session key. l is a parameter that is specified by the router during the join phase and can be modified by the router when it updates the frequency hopping order. During the cycle number l, the frequency hopping slots are used in reception mode by the end-devices.

3.2 Key Management

Our key management is based on three phases: before deployment phase, join phase and neighbor discovery. The objective is to generate a symmetric key with every neighbor for the routers, and a symmetric key for every router and each of its end-devices.

3.2.1 Pre-deployment Phase

Keys are generated and predistributed before deployment. Nodes in a wireless sensor network come from the same manufacturer, unlike the internet where nodes are very heterogeneous. This feature makes it easier to be able to distribute keys to nodes in a controlled manner. For each node in the network (routers and end-devices) we generate a pair of asymmetric keys (a public key and a private key). We install these two keys in addition to the public key of the sink in each node of the network.

3.2.2 Join Phase

First step after the deployment is joining phase. Let us consider a new router R that wants to join the network through the router F (see figure 1). R sends a join request message (*joinReq*) to F containing frequency hopping information (the set of frequencies that R wants to use for reception) and its public key (P_R). This join request message is encrypted with the public key of S (P_S) using ECAES as described in (Lopez and Dahab, 2000). F receives the join request message and relays it to C in order to reach the sink node S. S examines the join request and decide whether to accept it or not. In case it is accepted, S sends back a join response encrypted using the private key of S (K_S). When F receives the join response encrypted with the private of S, it can be sure that the sink has accepted that R joins the network. F decrypts the join response using the public key of S (P_S) and retrieves the public key of R. Then, F sends to R the join response encrypted with the public key of R. This join response includes the public key of F (P_F) and a session key (SK_{FR}). The session key SK_{FR} is used to encrypt communication between F and R. This communication is detailed on figure 3.

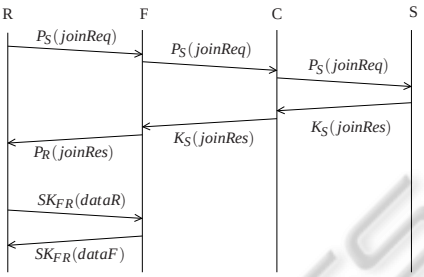

Figure 3: Join phase.

3.2.3 Neighbor Discovery

The sink plays a critical role in the join phase. It needs to authenticate the public keys of every node that wants to join the network. When this authentication fails, it sends back a reject message to that node. Otherwise, it sends a join response. The sink saves all the public keys of the nodes in the network. This way, whenever a node wants to know the public key of another node, it sends a public key request to the sink with the identifier of the other node (the identifier can be its physical address). The sink sends back the public key encrypted using its private key.

Session key exchange is done with every neighbor detected by the routing protocol. In the case of session key creation between neighboring routers (for example, routers A and B in figure 1), the node that creates the session key is the node with the higher priority. The priority of a node is derived from its identifier. The node with the smaller identifier has the higher priority. When node A identifies node B as a neighbor, it checks if it has a higher priority. If it is the case, A sends a public key request (*pkReq*) to the sink to obtain the public of node B (P_B). S sends back a public key response (*pkRes*) to A containing the public key of B (P_B). Then, A sends a session key establishment (*skEst*) message to B encrypted with P_B including the frequency hopping information, a session key (SK_AB) and the public key of A (P_A). B replies with an acknowledgment (*ACK*) including its frequency hopping information. Frequency hopping information indicates the time slots used by a router to communicate with other routers, the frequency and the permutation cipher used in each slot. If B has the higher priority, it is the other way around. This phase is represented on figure 4.

Session keys are then updated every n cycles, where n is the maximum number of cycles tolerated

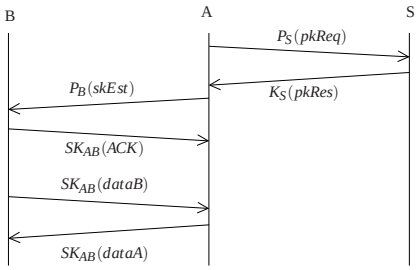

Figure 4: Neighbor discovery phase.

before compromising the session key. In order to make encrypted diffusion possible using symmetric cryptography, the sink creates a global session key. This session key is then propagated to all the routers of the network in a hop by hop manner along the parent child relationship starting from the sink. This session key is used do encrypt diffused messages such as the beacons of the synchronization period. The global session key is updated every *k* cycle and diffused in the beacon frame, where *k* is the maximum number of cycles tolerated before the global session key is compromised. The new global session key is encrypted using the old session key. Local common session keys could also be created for each star by the router of the star. This local session key is used for diffusion inside the star. Each router specifies a reserved slot that is used for diffusion inside the star on a certain frequency.

3.3 Permutation Ciphering

As cryptographic operations take a lot of time on sensor nodes, we only encrypt critical information using the session keys. To accelerate the data exchange, we use permutation ciphering. Indeed, the frequency hopping information contains, in addition to the frequency used in each slot, a permutation cipher that enables nodes to encrypt data that they exchange. We suppose that the permutation should change from one slot to another. So for each slot a node should specify a permutation cipher and a frequency.

The permutation cipher is applied on words. Each word is a group of 31 symbols (a symbol is a group of 4 bits). The rationale behind this grouping is that the IEEE 802.15.4 physical layer uses the O-QPSK modulation were data is grouped in symbols (IEEE 802.15, 2006).

When a node has a data to transmit (whether its an end-device or a router), it divides it into 8 words composed of 31 symbols each as explained on figure 5. In this manner, we are able to apply a permutation maximum frame size of 124 bytes of data on the MAC level. When the data that a node needs to send is less than 124 bytes, the frame is stuffed with bits of 0.

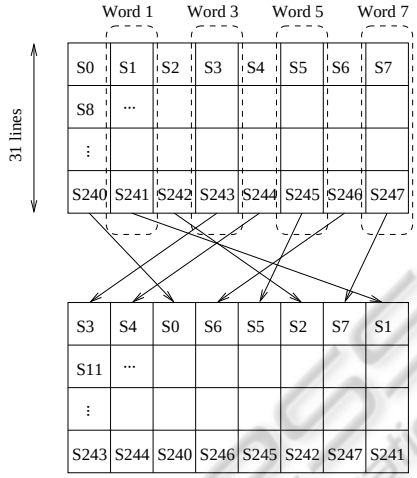

Figure 5: Word permutation using the permutation cipher (2, 7, 5, 0, 1, 4, 3, 6).

The transmitted data will be the concatenation of the words. Hence, in the example shown on figure 5, the order in which the words are transmitted is the following: word 3, word 4, word 0, word 6, word 5, word 2, word 7, word 1.

In order to decipher the received frame, a node needs to have the permutation cipher. The permutation cipher is deduced from the frequency hopping information exchanged either during the join phase or the neighbor discovery phase. This information contains the frequency used during each slot and the permutation cipher associated to that slot.

4 SECURITY AND ENERGY EFFICIENCY ANALYSIS

In what follows we go through the different security aspects that are taken into considerations in this architecture.

Confidentiality. During the join phase after the deployment, exchanges are encrypted using the ECAES protocol. Once pairwise session keys are established between nodes, data is exchanged and encrypted using a symmetric key using AES encryption for critical data like frequency hopping information and beacons, and application data is encrypted using the permutation cipher.

Source Authentication and Integrity. Message integrity and source authentication can be ensured using

ECDSA for generating the hash of the message and signing it using the private key of the sender.

Availability. In our solution the intruder can only paralyze the system by physically accessing the individual sensor nodes. Whereas our frequency hopping mechanism is resilient against a denial of service attack. It makes it hard on an attacker to jam the network, it will need to jam over all the available channels simultaneously. Even if it does so, when the MAC layer is not able to find a free channel to use, it will stop sending messages. By not sending messages, the network supervisor is able to notice the malfunction and suspect a jamming over all the available channels.

Energy-efficiency. The time segmentation approach allows nodes to save energy and sleep during the time slots where they are not concerned by the communication. End-devices are active only during the time slots specified by the router to which they are associated. A router is active during the time slots of its end-devices, the time slots of its router neighbors and $[T_2; T_3]$. In addition, all nodes save energy during $[T_3; T_0]$.

5 CONCLUSIONS

In this paper we presented a energy-efficient security management for wireless sensor networks based on pre-distributing ECC public keys before deployment. These public keys are then used to create session keys to encrypt the exchanged critical information such as frequency hopping information and permutation ciphers. In addition, we proposed to use frequency hopping in a secure way by exchanging secret frequency hopping sequences which enable nodes to resist against deny of service attack and eavesdropping. Since cryptographic operations are very expensive in wireless sensor networks, we propose a permutation ciphering technique to protect data.

Our solution is dynamic as it allows nodes to join the network at any time after deployment given that this node has a valid public key. In our proposition, the sink node does not have to be in range of the routers. The sink node can reach all the routers of the network in a guarantees hop-by-hop manner. The security aspect of our proposition is still in progress. We are working on evaluations on telosB motes.

This work is partially funded by FEDER (European fund for regional development).

REFERENCES

Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3:325–349.

Chalhoub, G., Guitton, A., Jacquet, F., Freitas, A., and Misson, M. (2008a). Medium access control for a tree-based wireless sensor network: Synchronization management. In *IFIP Wireless Days*.

Chalhoub, G., Guitton, A., and Misson, M. (2008b). MAC specifications for a WPAN allowing both energy saving and guaranted delay - Part A: MaCARI: a synchronized tree-based mac protocol. In *IFIP WSAN*.

HART (2008). HART field communication protocol specifications. Technical report, HART Communication Foundation Std.

Hong, P. (2008). Feasibility of pkc in resource-constrained wireless sensor networks. In *ICCIT*.

IEEE 802.15 (2006). Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). Standard 802.15.4 R2006, ANSI/IEEE.

ISA (2009). ISA100.11a:2009 wireless systems for industrial automation: Process control and related applications. Draft standard, in preparation, International Society of Automation Std.

Jones, K., Wadaa, A., Oladu, S., Wilson, L., and Eltoweissy, M. (2003). Towards a new paradigm for securing wireless sensor networks. In *New Security Paradigms Workshop*.

Kavitha1, T. and Sridharan, D. (2010). Security vulnerabilities in wireless sensor networks: A survey. *Journal of Information Assurance and Security*, 5:31–44.

Kerkez, B., Watteyne, T., Magliocco, M., Glaser, S., and Pister, K. (2009). Feasibility analysis of optimal controller design for adaptive channel hopping. In *WSNPERF*.

Lopez, J. and Dahab, R. (2000). An overview of elliptic curve crytography. Technical report, University of Campinas.

Perrig, A., Szewczyk, R., Tygar, J., Wen, V., and Culler, D. (2002). SPINS: Security protocols for sensor networks. *Wireless Networks*.

Pister, K. and Lance, D. (2006). TSMP: time synchronized mesh protocol. In *Distributed sensor networks*.

Watteyne, T., Mehta, A., and Pister, K. (2009). Reliability through frequency diversity: Why channel hopping makes sense. In *PE-WASUN*.

Zigbee (2008). Zigbee Specification. Zigbee Standard 053474r17, ZigBee Standards Organization.