ENERGY EFFICIENT SECURITY ARCHITECTURE FOR WIRELESS SENSOR NETWORKS

Miodrag Živković, *PhD candidate, School of Electrical Engineering, University of Belgrade*, Irina Branović, Dragan Marković and Ranko Popović, *Department of Informatics and Computing, Singidunum University Belgrade*

***Abstract* —One of the primary goals in designing a smart home environment is to minimize energy consumption by efficient management of home appliances. Smart homes often employ wireless sensor networks (WSN) to monitor physical parameters (like light, temperature, or presence of users at home). There is a growing concern about protection of data acquired by easily accessible wireless sensors; on the other hand, smart home networking trend stresses the importance of further protecting the communication of acquired data. Public-key cryptography methods have been long considered as energy inefficient to be applied in constraint environments such as sensor networks. In this paper we revisit public-key data protection in WSN and design a fault tolerant security architecture based on the use of elliptic curve cryptography (ECC), whose primary goal is to maximize lifetime of sensor nodes. We propose the use of Java enabled, heterogenous and clustered WSN platform and discuss its advantages over existing solutions, which commonly imply heavy optimization of security code and procedures.**

***Key words* — energy efficiency, wireless sensor networks, elliptic curve cryptography, Java**

I. INTRODUCTION

Implementation of security in the area of embedded devices is a very active research area, with specific challenges due to limitations in computing power and bandwidth. Typical examples of application of embedded devices in a smart home environment are wireless sensor nodes and RFID tags. The security in wireless sensor networks (WSN) is gaining importance as sometimes a large number of nodes are exposed in hostile environments; if only one node is captured by the attacker, the network as a whole can be compromised. The main security requirements in WSNs, known as confidentiality and network access, respectively, are that the data exchanged in the network should not be read by an unauthorized third party and that the third party cannot join the network. The unique challenges of WSNs are that the nodes have limited energy and radio communication range, there is no device that can act as a trusted server, and the network topology is not known before deployment.

Although security is not a necessary prerequisite for all sensor networks, until recently, its implementation for WSN implied the exclusive use of various private-key cryptography methods for encrypting data and key agreement schemes. Implementation of public-key methods in WSNs has long been considered impractical, if not impossible, because of the huge computational power they require. The reason for this becomes obvious when considering that public-key cryptography methods use key lengths of 1024 bits and more, while a typical sensor node is 8-bit with only few kB of RAM. When the key length far exceeds a platform's native word size, as in the case of public-key methods, additional operations to emulate the "big number" operations are required, which come at a significantly increased time and energy consumption. However, benefits of introducing public-key cryptography in WSNs are twofold. First, WSN could benefit from a public-key scheme to implement services such as authentication, digital signature, as well as more efficient key distribution schemes. Second, the application of public-key methods could reduce power consumption in wireless nodes due to less protocol overhead.

Elliptic curve cryptography (ECC) has emerged as a suitable public-key cryptographic foundation that provides high security for relatively small key sizes (typically few hundred bits). Shorter key lengths ECC uses directly translate to energy savings because of the decreased number of emulated modulo instructions. However, to determine the extent to which ECC methods can be incorporated in next generation sensor nodes, energy and cost efficiency must be carefully examined.

Sensor nodes (often called motes) are often too weak to perform computationally intensive tasks, expensive both in terms of delay and energy. For this reason we design a heterogenous WSN architecture with few higher capacity motes to perform more demanding tasks, and a number of simple motes for data acquisition. Any security protocol developed for WSNs, as a special case of ad hoc network,

Miodrag Živković, School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Beograd, Srbija (phone: 063309918, email: zivkovic_miodrag@hotmail.com)

Irina Branović, Singidunum University Belgrade, Bulevar Zorana Đinđića 44, Beograd (email: ibranovic@singidunum.ac.rs)

Dragan Marković, Singidunum University Belgrade, Bulevar Zorana Đinđića 44, Beograd

Ranko Popović, Singidunum University Belgrade, Bulevar Zorana Đinđića 44, Beograd (email: rpopovic@singidunum.ac.rs)

must also take into consideration fault tolerance, e.g. provide for some degree of redundancy to account for potential loss of nodes. Our security architecture for WSN is a tradeoff between energy efficiency and robustness in fault tolerance and data protection. Besides being heterogeneous, it uses a custom implementation of elliptic curve cryptography methods to save energy.

The rest of the paper is organized as follows: Section 2 describes related work in the area of public-key WSN security architectures. In Section 3, background information on elliptic curve cryptography (ECC) is presented. In Section 4, a heterogeneous WSN architecture using ECC methods is described and analyzed from the point of view of energy efficiency. In Section 5, the simulation environment is described. Finally, Section 6 concludes the paper and presents future work.

II. Related Work

The use of public-key cryptography in WSNs was considered to be too expensive in terms of resources and energy, until [1] described the first implementation of elliptic curve cryptography for MICA2 platform. Subsequently, a number of public key architectures for WSNs have been studied. Most previous research in this area is concerned with efficient key distribution schemes. Since key establishment is the initial step for secure communication, topics on key establishment, agreement and distribution have been extensively studied in the literature, e.g. [2]. This work considers homogeneous sensor networks, where all sensor nodes have the same capabilities, and also describes an efficient ECC key distribution scheme in heterogeneous sensor networks, where sensor nodes have different capabilities in terms of communication, computation, energy supply, storage space, reliability and other aspects. However, a homogeneous WSN suffers from poor fundamental limits and performance, as described in [3]. The use of heterogenous models with higher capacity processing motes preserves energy, as discussed in [4].

The reduced resource usage and energy consumption of ECC make it optimal public-key cryptography scheme for use in WSN, as the energy analysis performed in [5] confirmed. Authors in [6] perform the energy analysis of some commonly used ECC algorithms in WSNs and conclude that listening state should also be considered when assessing the cost of cryptographic protocols on sensor nodes.

Clustering is introduced to WSNs because it has proven to be an effective approach to provide better data aggregation and scalability for large WSNs, and also because it conserves the limited energy resources of the sensors; paper [7] is an excellent survey on clustering algorithms for WSNs.

Most ECC implementations in WSNs rely on custom libraries heavily optimized for limited sensor platforms; perhaps the most known is TinyECC. This configurable library is optimized for use with the TinyOS operating system, developed in NesC language (a variant of C).

Recently, Java enabled wireless sensors appeared, such as SunSPOT. The shift to Java WSNs should increase the number of developers to whom wireless sensor technology will be accessible. SunSPOT architecture and ECC implementation through TLS protocol, are too expensive as they result in a reduction of 70% of network lifetime. Therefore, a Java enabled WSN requires a complete, custom ECC security architecture; our work presented here describes the first steps in filling this void, especially having in mind orientation towards energy efficiency.

III. Elliptic Curve Cryptography (ECC)

Elliptic curve cryptography (ECC) is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields GF. An elliptic curve over binary finite field is the set of points (x, y) that satisfy the following equation:

$$y^2 + xy = x^3 + ax^2 + b \,|\, a, b \in GF$$

together with the point at infinity.

The product of a point and an integer is an operation which, given a point P = (x, y) and an integer k, produces another point Q on the same curve, indicated as [k]P. Multiplication of a point P by an integer k is implemented as a repeated doubling and addition of the point to itself. Given a random integer k, it is relatively easy to compute Q = [k]P, but it is very difficult and time consuming to reverse the operation, i.e. to find k knowing only P and Q; finding such an integer k is equivalent to calculating the elliptic curve discrete logarithm of Q to the base P. This operation forms the basis of security in elliptic curve cryptosystems. A base point, G, is fixed for each curve. The random large integer, k, acts as a private key; while the result of multiplying k by the base point, G returns the corresponding public key. The security of elliptic curve cryptography methods is based on the discrete logarithm problem (finding *a* knowing P and [*a*]P is computationally very difficult). Although elliptic curves can be studied in any finite field, for cryptographic purposes fields with finite number of elements are used. Widely accepted NIST standard recommends use of chosen prime integer fields or binary polynomial fields. This standard can be followed for the choice of the curve, as well as the base point on it.

Elliptic curves can be used for implementing commonly used public-key schemes, such as digital signing and key exchange. For example, the known Diffie-Hellman key exchange scheme based on elliptic curves would be implemented as follows: nodes A and B agree on a particular curve with base point P. They generate their public keys Q_a, Q_b by multiplying their private keys K_a and K_b with the base point:

$$Q_a = P \times K_a, Q_b = P \times K_b$$

After sharing public keys, they generate a shared secret key R by multiplying public keys by their private keys:

$$\mathrm{R} = \mathrm{K}_a \times \mathrm{K}_b \times \mathrm{P}$$

With known values of Q_a, Q_b, and P it is computationally infeasible for an eavesdropper to calculate

private keys K_a and K_b.

TABLE 1: SPECIFICATIONS OF MICAZ AND SUNSPOT PLATFORMS.

	MICAz	***SunSPOT***
Processor	8-bit ATMEL	32-bit ARM9
RAM	4KB	512KB
Program memory	512 KB	4MB
Operating system	TinyOS	none
Radio	CC420	CC420

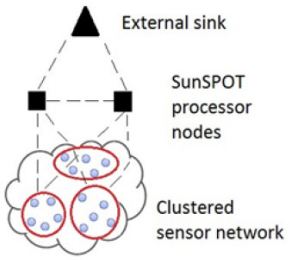

Fig. 1. Heterogeneous WSN using simple clustered motes for information gathering, and two fault-tolerant SunSpot nodes for computationally intensive ECC operations.

IV. ENERGY EFFICIENT WSN ARCHITECTURE

A wireless sensor network (WSN) consists of low-complex and low-size devices called sensor nodes or motes that can sense the environment and communicate gathered information through wireless link to a sink. A sink, also called a base station, is usually an a personal computer configured to collect, save or react according to the data. The network between the nodes and the sink is dynamic and self-organizing. Simple wireless nodes such as MICAz are well suited for sensing and simple processing, but offer very limited hardware and energy resources.

Sun Microsystems have developed a WSN platform that runs Java code “on-the-metal” of their motes, known as Sun SPOTs. This “stronger” node is well suited for processing tasks. The SunSPOT is a wireless sensor node platform based on an ARM920T CPU running at 180MHz and offering 512kB of RAM (Table 1). It runs a Java Virtual Machine “on the metal” (without an underlying operating system), and can therefore execute Java applications. SunSPOT has a more powerful processing unit and a much larger data storage. This makes in-network processing and local data storage possible, and therefore reduces radio usage so as to reduce power consumption.

The "migratable application" functionality of SunSPOTs enables applications (with their complete state information) to be dragged from one Sun SPOT device to another while in execution. It is possible to move software off a SPOT device with low battery power onto another device with more battery life, avoiding loss of state information and enhancing fault tolerance.

A heterogeneous WSN architecture is ilustrated in Figure 1. This sample network topology has clustered low-power sensor nodes, two processing (SunSPOT) nodes, and a single external sink. Low-power nodes perform sensing and basic processing tasks, while an external sink node acts as data collector.

Dedicated SunSPOT nodes are present within the network to perform data processing operations, thereby alleviating the energy-consuming multi-hop data transport to the sink, while overcoming resource limits of low-power platforms. When more powerful processing nodes are integrated with the sensor network, their greater computational capabilities allow them to perform strong cryptography operations within reasonable time limits. Especially, the 32-bit word size and the significantly increased RAM size of the processor nodes reduce the need for instruction emulations and expensive data buffering on external memory, and can thus reduce the required execution time. They serve as trust points which encrypt sensed data before storing it in the cloud, and perform access control for the stored data. Communication is a large obstacle when building wireless sensor networks out of different sensor nodes, because of the incompatible implementations of network protocol stacks on different sensor platforms. SunSPOT and MICAz nodes both are using IEEE 802.15.4 (Texas Instruments’ CC420) radio chip. IEEE 802.15.4 offers a reduced energy consumption, with a trade-off of a lower bandwidth compared to the 802.11. IEEE 802.15.4 only specifies the two lowest layers of the OSI model: physical and link layer.

However, even using the same radio chips on two different sensor nodes does not guarantee working communication, as standards might be interpreted or implemented unequally (different addressing schemes, interpretation of header flags or timing issues). For example, IEEE 802.15.4 standard supports two ways of addressing source and destination inside a packet. Each node is supposed to have a 64-bit address. However, since packets are rather small, the second addressing scheme was defined, using 16-bit addresses. SunSPOT node is assigned a 64-bit address by default, and radio chip is configured to accept all packets bearing this address or broadcast messages, and discard all other addresses. On the other side, MICAz nodes use 16-bit addresses. So, in order to establish connectivity, it is necessary to adapt the addressing scheme. Timing is also an issue, as acknowledgments for packets sent from SunSPOT to MICAz sometimes arrive to late, causing the retransmissions. Time slot on SunSPOT has to be extended, so that SunSPOT will wait longer for the acknowledgment before considering packet as a lost and scheduling retransmission. Routing protocols and forwarding policies must also be harmonized in order to enable coherent network structure when building network of different node platforms. For example, significant difference between two platforms is maximum payload. SunSPOT node has longer payload than MICAz, therefore maximum payload for SunSPOT must be reduced to length of MICAz payload.

To limit the number of communications which consume a lot of energy, we propose node clustering, i.e. dividing a network into smaller groups of clusters. In our case there are two clusters, with one SunSPOT as a cluster head of each cluster. In each step the closest nodes are merged, and theprocess continues until all nodes have been merged into

oneof two top level clusters. Simple sensor nodes are connected directly to the cluster-head, because routing in clusters is not necessary. Nodes transmit information to their cluster heads, and data aggregation is performed at the cluster head. In this way, network lifetime can be prolonged through reduction of the nodes contending for the channel access, load balancing, increased connectivity and reduced delay.

In our case, multi-hop sensor-to-cluster head connectivity is required, as sensor's communication range is limited. Our model assumes tree-based intra-cluster topology, with nodes transmitting sensor data towards the cluster head. Packets are routed by the shortest path algorithm to the cluster head, and then relayed to the base station trough cluster heads.

The security protocol we propose has two phases. First, cluster nodes handshake with the cluster head to setup a session key to secure end to end link between them. For this purpose, Elliptic Curve Diffie-Hellman key exchange argorithm (ECDH) is used. In the second phase, the established session key is used for private-key data encryption to ensure confidentiality of the exchanged data. Since both MICAz and SunSPOT use CC420 radio chip, with built-in hardware AES encryption with 128-bit key, it is the natural choice for private-key encryption algoritm due to its efficiency.

Compared to other public-key schemes, the use of ECC makes the proposed scheme energy efficient and gives a considerable threshold of security. Also, we are developing custom Java code for ECDH algorithm, instead of applying available ECC based TLS suite which has proven to be energy inefficient.

V. SIMULATION ENVIRONMENT

To evaluate a security framework for wireless sensor networks, we are developing a Java based discrete event simulator, built on top of JProwler. JProwler [7] is a discrete event simulator for prototyping, verifying, and analyzing communication protocols of ad-hoc networks. JProwler was selected because it has been widely used in the simulation study of wireless sensor networks. Each node is able to communicate with other nodes in its range using wireless (Gaussian or Rayleigh) channels with the MAC protocol of MICAz nodes. Distance-vector routing protocol is implemented in order to calculate paths. SunSPOT nodes are simulated using the same MAC protocol; the collision avoidance mechanism is not considered at this time, and all nodes are considered to be stationary. Individual node behavior can be described as a sequence of periodically occurring events (testing the channel, transmission of the message). We assume that all nodes are using the same channel for transmitting messages. However, all nodes would fail to receive the message due to packet collisions. Successful transmission depends on the number of the generated nodes and rate of generating new messages. In our simulation, we used two SunSPOT nodes at fixed positions, surrounded by 12 randomly placed MICAz nodes in simulation area 60x60 meters. SunSPOT nodes are generating broadcast messages, and sending them to all paths calculated by the distance-vector algorithm. Distance-vector matrix is calculated by representing the weight of an edge between two nodes as the distance between these nodes.

VI. CONCLUSION

This paper is a first step in designing a Java security framework for wireless sensor networks whose primary goal is to be energy efficient. We propose a heterogeneous WSN consisting of two Java enabled nodes and numerous clustered motes. Java enabled nodes (SunSPOTs) execute computationally expensive security operations such as generating keys and authenticating nodes, and serve as a trusted point towards an external sink. To limit the number of communications which consume a lot of energy, we propose node clustering. The security architecture implies the use of elliptic curve Diffie Hellman for key exchange, and hardware enabled AES encryption for secure communication. We plan to investigate efficient clustering algorithms, physically implement described WSN architecture, and measure energy consumption of different ECC security procedures. A Java ECC library supporting different elliptic curve schemes is also under development. The proposed security architecture should also be expanded to include message authentication and secure routing.

ACKNOWLEDGMENT

This paper has been partially supported by the Serbian Ministry of Education and Science under project grant no. 44006.

LITERATURA

[1] D.J Malan, M. Welsh, and M.D. Smith, "Implementing Public-Key Infrastructure for Sensor Networks", ACM Transactions on Sensor Networks, Vol. 4, No. 4, Article 22, 2008.

[2] W. Gu, N. Dutta, S. Chellappan, and X. Bai, "Providing End-to-End Secure Communications in Wireless Sensor Networks", IEEE Transactions on Network and Service Management, vol. 8, no. 3, pp. 205-218, 2011

[3] X. Du, M. Guizani, Y. Xiao, and H. Chen, "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks", IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1223-1229, 2009..

[4] A. Reinhardt, R. Steimentz, "Exploiting Platform Heterogeneity in Wireless Sensor Networks for Cooperative Data Processing", Proceedings of the 8th GI/ITG KuVS Fachgespracht "Drahtlose Sensornetze", Hamburg, Germany, 2009.

[5] A.Wander, N. Gura, H. Eberle, V. Gupta, S.C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks", Proceedings of the 3rd IEEE Conference on Pervasive Computing and Applications, pp. 324 - 328, 2005.

[6] G. Meulenaer, F. Xavier, and L. Vandendorpe, "On the Energy Cost of Communication and Cryptography in Wireless Sensor Networks", IEEE International Workshop on Security and Privacy in Wireless and Mobile Computing, Networking and Communications, pp. 580 – 585, 2008..

[7] O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, and R. Poliah, "A Survey on Clustering Algorithms forWireless Sensor Networks", Proceedings of 13th IEEE International Conference on Network-Based Information Systems, pp. 358-364, 2010.

[8] JProwler simulator. Available at http://w3.isis.vanderbilt.edu/projects/nest/ jprowler/. (accessed 10 October 2012).