POWER ANALYZE OF ELLIPTIC CURVE CRYPTOGRAPHY FOR USAGE IN WIRELESS SENSOR NETWORKS

Maja Kukuseva[1], Aleksandar Risteski[3], Dusan Bikov[2], Dejan Milcevski[1]

[1] Faculty of Electrical Engineering, St. 22-ri Oktomvri bb, 2420 Radovis, Macedonia, maja.kukuseva@ugd.edu.mk, dejan.milcevski@ugd.edu.mk

[2] Faculty of Computer Science and Information Technologies, St. Krste Misirkov bb, 2000 Stip, Macedonia, dusan.bikov@ugd.edu.mk

[3] Faculty of Electrical Engineering and Information Technologies – Skopje, Karpoš II bb, 1000 Skopje, Macedonia, acerist@ukim.edu.mk

***Abstract* –Wireless Sensor Networks are new emerging technology used in various applications for habitat, health and air pollution monitoring, vehicle tracking etc. Every sensor node is powered with battery that must last for mounts or years which constrains the WSN in term of energy used. Other critical issue is the secure communication between the nodes and the deployment of cryptography method is challenging task. Elliptic Curve Cryptography (ECC) is low- power consuming cryptographic scheme that combines high- level of security and low power usage for key generation and its distribution through the network.**

***Keywords* – Wireless Sensor Network (WSN), Elliptic Curve Cryptography (ECC), Elliptic Curve Discrete Logarithm Problem (ECDLP), Elliptic Curve Diffie- Hellman (ECDH).**

1. INTRODUCTION

The term Wireless Sensor Network (WSN) refers to a heterogeneous network that combines autonomous tiny nodes named as sensors (data acquisition) with computing elements (data processing). WSN is consisted of hungers or thousands of low-cost and low-power limited nodes, called sensor nodes, mostly with fixed locations used for environment monitoring and data acquisition. The sensor nodes have local power sources such as onboard battery used for communication and data processing. Battery must last for long time periods (days, months and even years) so it is necessary for low- power operations and transmission. Every sensor network faces several challenges in term of limited hardware, scalability, secure wireless communication (jamming, eavesdropping) and physical layer attacks.

Several standardization bodies are working on deploying standards for wireless sensor networks. One example is IEEE 802.15.4 that defines physical layer and MAC layer. IEEE 802.15.4 is wireless communication protocol for low- cost devices with limited resources. The PHY layer operates on three frequency bands: 868.0-868.6MHz (Europe, one channel), 902-9128MHz (North America, 30 channels), 2400-2483.5MHz (word-wide, 16 channels). Data transmission rates are 20kpbs (868MHz band), 40kbps (902MHz band) and 250kbps (2.4GHz band). The Medium Access Control (MAC) is responsible for channel access, beacon and GTS management, frame validation and acknowledgment. To regulate channel access 802.15.4 uses carrier sense multiple access with collision avoidance (CSMA-CA) scheme. IEEE 802.15.4 has basic built security in MAC layer that relies on symmetric cryptography. Thus, security became crucial issue particularly for applications where WSN is developed in hostile environment. In order to establish secure network it is necessary to develop and design secure protocols to deal with the encryption issues and key management

Wireless Sensor Networks use Elliptic Curve Cryptography (ECC) due difficulty of solving discrete logarithm problem (DLPP). Elliptic Curve Cryptography is emerging technology with low memory resources usage, low processing power and higher security against powerful attacks. These makes it suitable for usage in energy constrained wireless senor networks.

In this paper power analysis will be represented with offered optimization. The analysis is done using software packet EccM-2.0 implemented in TinyOS. Sector 2 discuses the basic of Elliptic Curve Cryptography. Sector 3 represents the basics of ECDH and ECDSA. Sector 4 represents the simulation scenarios, obtained results and their analysis. Section 5 concludes.

2. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

ECC is new public key cryptographic scheme independently developed by Kobilz [1] and Miller [2] in 1985. An elliptic curve is two- dimensional curve used in many mathematical and cryptographic algorithms (e.g. public key cryptosystem, pseudo-random number generator etc). The elliptic curve is a cubic equation over binary (or prime) field F given with Weiestrass Equation (1).

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

Where a_1, a_2, a_3, a_4, a_6,$\in F$ and discriminate of the curve $\Delta \neq 0$.

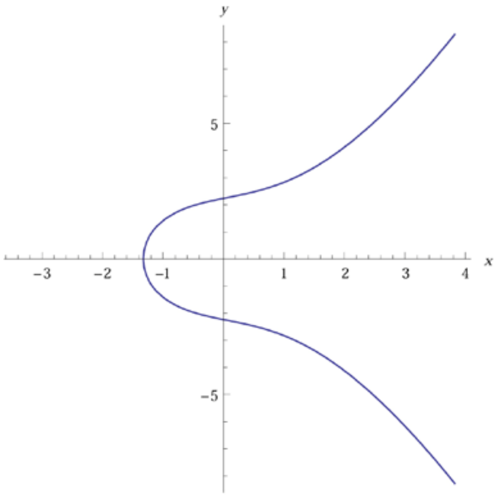

Fig. 1 – Elliptic Curve y^2+xy=x^3+2x^2+1over binary fields

Over binary fields the Weiestrass equation is transformed into simpler form (2)

$$E:\, y^2 + xy = x^3 + ax^2 + b \qquad (2)$$

Where a, b $\in F_{2^m}$ and $\Delta = 4a^3 + 27b^2 \neq 0$. Fig. 1 shows the most commonly used elliptic curve y^2+xy=x^3+2x^2+1 over binary fields. The set of points x and y together with a special so-called point to infinity O form a commutative group. The arithmetic operation such as adding, subtraction, multiplication, inversion and squaring are defined in [3].

The security of ECC lies in the difficulty to solve the discrete logarithm problem over finite fields. In order to solve this problem integer k must be found, so that for a given point P from the elliptic curve, Q=kP. This problem is practically insoluble for sufficiently large integer value of k. The main operation with the points of elliptic curve is finding geometric sum $P(x_3, y_3)$ of two points $Q(x_1, y_1)$ and $R(x_2, y_2)$ over binary field. This operation is not simple and is performed using Equations (3) and (4) over binary fields.

$$x_3 = \left(\frac{y_2 + y_1}{x_2 + x_1}\right)^2 + \left(\frac{y_2 + y_1}{x_2 + x_1}\right) + x_1 + x_2 + a \qquad (3)$$

$$y_3 = \left(\frac{y_2 + y_1}{x_2 + x_1}\right)(x_1 + x_3) + x_3 + y_1 \qquad (4)$$

Note that these operations are per module p. But these operations cannot be used for point addition by itself because the denominators in the Equation (3) and (4) are zero and division by zero is obtained. Point addition by itself is a special case of two point's addition known as duplication of the point over binary fields. It is performed using the equation (5) and (6).

$$x_3 = x_1^2 + \frac{b}{x_1^2} \qquad (5)$$

$$y_3 = x_1^2 + \left(\frac{x_1 + y_1}{x_1}\right)x_3 + x_3 \qquad (6)$$

Multiplication kP is obtained by performing addition operation k times using point addition and point doubling. Let k be a scalar and equal to 23 then the multiplication kP=23P=2(2(2(2P) +P) +P).

As mention before, the weight of ECDLP is the problem of determining integer k. ECC relies on the difficulty of solving ECDLP because if the eavesdropper is able to solve it, then eavesdropper will be able to break down the system. But, based on arithmetic operation of elliptic curve it is difficult to calculate k because the problem is associated with factorizations of large integers. The main advantage of ECDLP over traditional public key cryptographic schemes is that obtains same level of security with smaller key size.

3. EC DIFFIE- HELLMAN (ECHD) AND EC DIGITAL SIGNATURE (ECDSA)

Elliptic Curve Diffie- Hellman (ECDH) is a protocol for key agreement that allows sensor nodes to establish a shared key that can be used for private key algorithms. The shared key is calculated using public data and private data of the sensor node. Any third party that doesn't have access to private data of each node will not be able to calculate the shared key.

First step in generating shared key, sensor nodes A and B must agree for EC domain parameters that define the elliptic curve. Both nodes have a key pair consisting of a private key d, which is randomly selected integer less than the order of the curve n, and public key Q=dG, where G is generator point. Let (d_A, Q_A) and (d_B, Q_B) are the pair private key- public key of sensor nodes A and B appropriately. The sensor node A computes K=(x_K,y_K)=d_A*Q_B, while sensor node B computes L=(x_L,y_L)=d_B*Q_A. Since d_AQ_B=d_Ad_BG=d_Bd_AG=d_BQ_A. Therefore K=L and hence the shared key is x_K since x_K=x_L.

Elliptic Curve Digital Signature (ECDSA) [4], is a form of a digital signature algorithm applied on elliptic curves to authenticate the senor nodes or message sent by the sensor node. The digital signature is a value calculated from the value of the secret (private) key and is modern replacement of handwritten signature. For sending a signed message sensor nodes A and B must agree upon EC domain parameters. In order node B to authenticate a message sent by node A, node A must signs the message using its private key. The message and the signature are simultaneously sent to node B, and since node B knows node's A public key easily can verify the message. ECDSA is an asymmetric signature scheme that provides resistance to various attacks because the signature and the message are independent from each other.

4. SIMULATION SCENARIOS, RESULTS AND ANALYSE

The simulations are performed on the free open source operating system TinyOS that has component-based and event-based architecture. TinyOS [5] is not a general purpose OS but is a flexible framework designed for wireless embedded sensor networks which require minimum hardware (RAM and ROM memory). One of the most important aspects of TinyOS is its modular design. Running TinyOS program is simply a combination of different modules connected together to comprise the end program. Every application is set of tasks and processes written in NestedC (nesC) programming language. NesC is an extension of C structured programming language optimized for the memory– restricted wireless sensor networks.

TOSSIM (TinyOS SIMulator) is TinyOS discrete event simulator which allows debugging, testing and analyze of the implemented EccM-2.0 code [6]. The main advantage of this simulator is ability to simulate simultaneously thousands of sensor nodes and compiles directly from TinyOS code. Thus, developers can test not only their algorithms but also their implementations. TOSSIM simulates the TinyOS network stack at the bit level, allowing experimentation with low-level protocols in addition to top-level application systems. As TOSSIM runs on a PC, users can examine TinyOS code using debuggers and other development tools. TOSSIM supports radio and energy (power) model. The simulations presented below are done using the energy model. The energy model does not model the consumption of energy, but easily request information from the components if their state is changed (e.g. whether the sensor node is on or off).

EccM-2.0 represents large integers using a byte array known as "big integer" or bigints. The right most element of the array represents the 8 least significant bits of the big integer. The left- most elements represents the 8 most significant bits. EccM-2.0 performs modular arithmetic on bigints. It takes two byte arrays that represents bigints and bit length of the bigint, uses the second bigint argument as the modulus and then places the result in the first bigint argument. The keys are 163-bits and are randomly selected number between 0 and n (n is order of the elliptic curve) with equal probability from which 80 bits are for security.

The simulations presented below are performed according the NIST recommendation for Elliptic Curve Cryptography (ECC). EccM2.0 implements ECC over binary finite fields with the irreducible polynomial $f(x) = x^{163} + x^7 + x^6 + x^3 + 1$.The curve's order (the number of points) is *n*=0x400000000000 0000000020108a2e0cc0d99f8a5ef and the base point is *G*(*x*,*y*) where *x* =0x2fe13c0537bbc11acaa07 d793de4e6d5e5c94eee8 and *y* = 0x289070fb05 d38ff58321f2e800536d538ccdaa3d9. After compiling EccM-2.0 the consumed RAM and ROM are represented in Table 1.

Table 1Consumed RAM and ROM

Compiled EccM2.0	
40382 bytes	RAM
1067 bytes	ROM

In this paper are presented simulations on two different types of network topologies, grid and random with or without power management. Fig 2 shows random and grid topology of 4 nodes represented in TinyViz. . The number of nodes must be an integer form square root thus, each set of simulations contain from 4 up to 100 nodes. During the simulation three parameters were examine: CPU Total, Radio Total and Total Energy. The parameter CPU Total is consumed energy for generating the pair of private and public key and processing of the received messages, while the parameter Radio Total is the consumed energy for transmission of the public key and consumed energy due communication with other sensor nodes. The parameter Total Energy gives the overall energy consumption and is sum of parameters CPU Total and Radio Total.

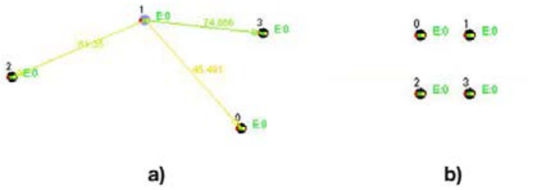

Fig. 2 – a) Random topology b) grid topology

The first simulation scenario was without power management on grid and random topology. Each topology was tested in simulation with duration of 10 and 20 seconds. The private key is 163- bits randomly generated. Fig. 3, Fig. 4 and Fig. 5 represent the average values for parameters CPU Total, Radio Total and Total Energy all measured in mJ.

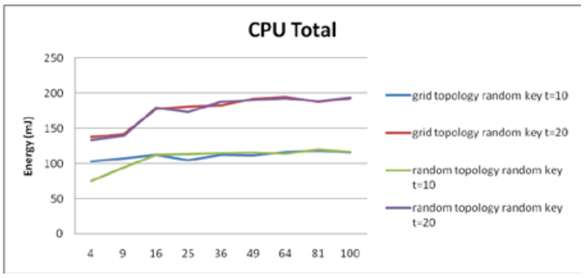

Fig. 3 – CPU total energy consumption

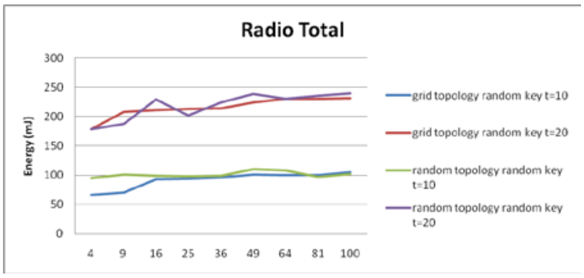

Fig. 4 – Radio Total energy consumption

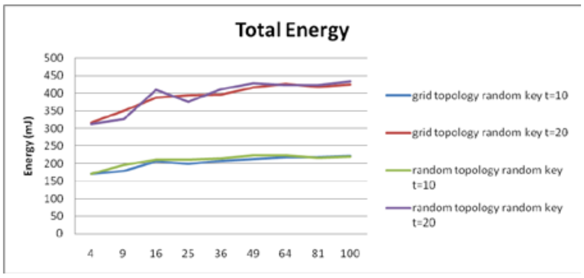

Fig. 5 – Total Energy consumption

In order to prevent from multiple-key single- message attack the code is changed and the private key is fixed. The next set of simulations is performed under the same conditions as the previous set. The result from testing parameters CPU Total, Radio Total and Total Energy are represented on Fig. 6, Fig. 7 and Fig. 8, appropriately.

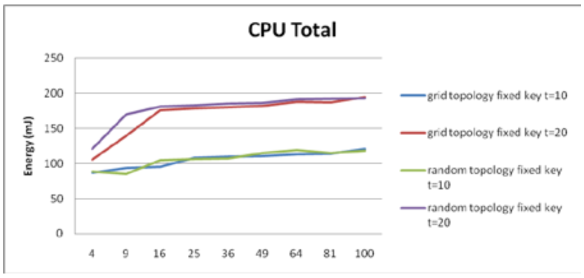

Fig. 6 – CPU Total energy consumption

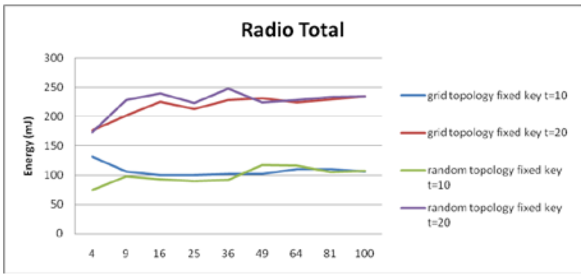

Fig. 7 – Radio Total energy consumption

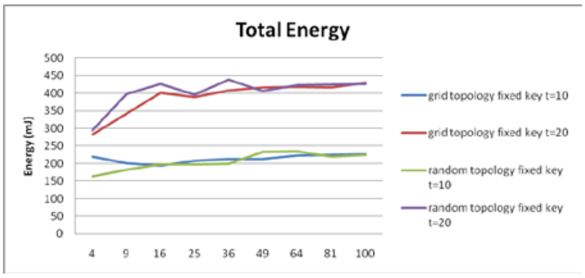

Fig. 8 – Total energy consumption

These two simulation scenarios show that energy consumption is increased as the number of sensor nodes grows. Also energy consumption does not depend from the network topology. The fix key prevents only from multiple-key single- message attack and does not influent on energy consumption.

By default, TOSSIM uses an old version of the mica radio stack (40Kbit RFM), including the MAC, encoding, synchronous acknowledgements and timing but does not support power management and tuning transmission power. It does not simulate the mica2 ChipCon CC1000 stack by default. PowerTOSSIM includes a port of the mica2 radio stack, so it is now possible to run programs that take advantage of the CC1000's power management features. The next simulation scenarios are with duration of 10 and 20 seconds and tasted on grid and random topology for random and fixed keys. Fig. 9, Fig. 10 and Fig. 11 represent the average values for parameters CPU Total, Radio Total and Total Energy measured in mJ with included power management using random key. Fig. 12, Fig. 13 and Fig. 14 represent the average values for parameters CPU Total, Radio Total and Total Energy, appropriately, when fixed key is used for preventing multiple- key single- message attack. The results obtained by this simulations match with the previous results. Increasing the number of sensor nodes led to more power consumption.

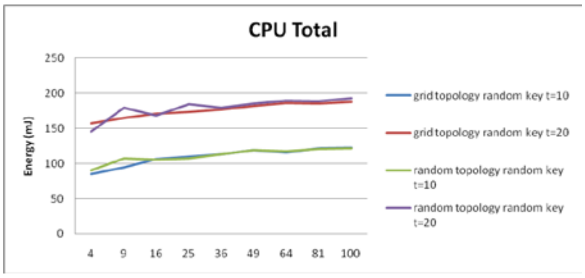

Fig. 9 – CPU Total energy consumption

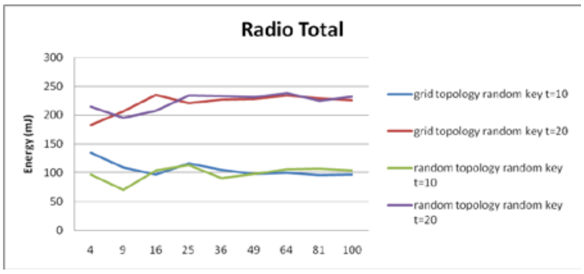

Fig. 10 – Radio Total energy consumption

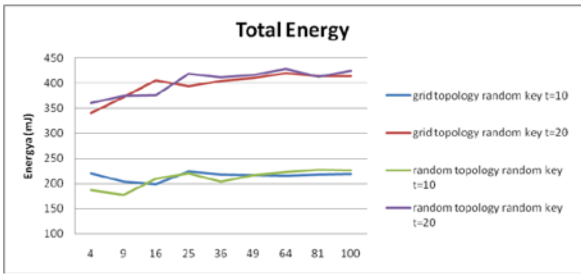

Fig. 11 – Radio Total energy consumption

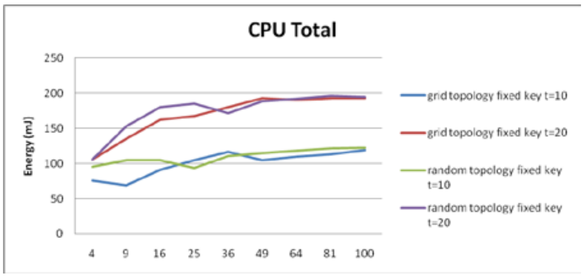

Fig. 12 – Radio Total energy consumption

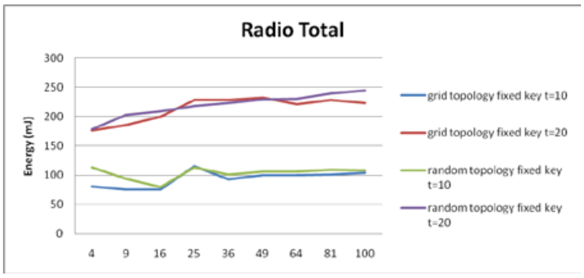

Fig. 13 – Radio Total energy consumption

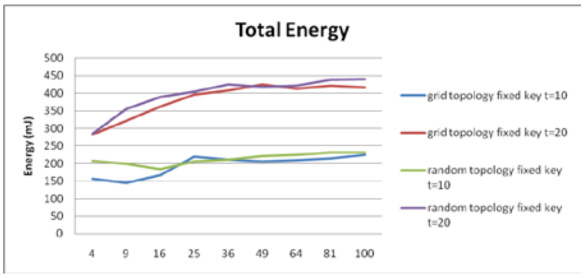

Fig. 14 – Radio Total energy consumption

The implementation of power management function led to slightly decreasing of the energy which is critical in networks with constrained resources. Also the energy consumption increases when the duration of the simulations is extended. This is because every sensor node generates pair of random and private keys for every new communication with other sensor node although these sensor.

5. CONCLUSION

Wireless sensor networks face limitations in term of energy consumption and supply. Therefore, implemented Elliptic Curve Cryptography must trade-off low-energy usage and high- level of security. In order to perform analyze simulation with two different network topologies and different number of sensor nodes were performed. The simulations were performed using PowerTOSSIM which is scalable simulation environment for WSN that provides accurate, per node estimation of the power consumption. This allows researching, testing and analyzing wireless sensor networks in the process of planning the network.

Performed simulations and calculations show that fixing the key does not influence the energy consumption. Fixed key only prevents from multiple-key single- message attack. Also, increasing the number of sensor nodes led to constant increase of energy consumption. For further decreasing of the energy consumption power management was implemented which slightly decreased the consumed energy.

6. REFERENCES

[1] Neal Koblitz: *Elliptic Curve Cryptosystems*, Mathematics of Computation, Vol. 48, No 177, January 1987, pp. 203- 209.

[2] Victor Miller: *Use of Elliptic Curves in Cryptography*, Advances in Cryptography- CRYPTO, LNCS, Vol. 218, 1987, pp. 417- 426.

[3] Darrel R Habkerson, Alfred J. Mandezes, Scott Vanstone: *Guide to Elliptic Curve Cryptography*, Springer Velag, 2004.

[4] Don Johnson, Alfred Menezes, Scott Vanstone: *The Elliptic Curve Digital Signature Algorithm (ECSSA)*, International Journal of Information Security, Vol.1, No. 1, pp.36-63, July 2001.

[5] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Alec Woo, David Gay, Jahonson Hill, Matt Welsh, Eric Brewer, David Culler: *TinyOS: An operating system for sensor networks*, Ambient Intelligence, Springer- Verlag, pp. 115- 148, 2004.

[6] Phil Levis, Nelson Lee, Matt Welsh, David Culler: *TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Application*, In proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2003.