ECDH POTROŠNJA ENERGIJE U SENZORSKIM BEZŽIČNIM MREŽAMA
ECDH POWER CONSUMPTION IN WIRELESS SENSOR NETWORKS

Maja Kukuseva, Biljana Citkuseva Dimitrovska, *Faculty of Electrical Engineering, University Goce Delcev, Stip, R.Macedonia*

Sadržaj: *Kriptografija eliptičnim krivama je kriptografska šema koja kombinuje niski nivo korišćenja snage za generiranje ključeva i visoki nivo sigurnosti u ograničenim senzorskim bezžičnim mrežama. Svaki senzorski čvor prima energiju iz baterije koja bi trebala imati vrlo dugi vek. Ovo ograničava senzorske bezžične mreže u smislu korišćenja energije. Drugi bitni problem je sigurna komunikacija i zbog toga je razvoj kriptografskih šema težak i zahtevan zadatak. Eliptična kriva Diffie-Hellman je sigurnosni protokol za dogovor za razmenu ključeva koji obezbeđuje dvema stranama da imaju par koji sadrži javni i privatni ključ za razmenu poruka kroz neobezbeđeni kanal za komunikaciju.*

Abstract: *Elliptic Curve Cryptography is cryptographic scheme that combines low power usage for key generation and high level of security in constrained Wireless Sensor Networks. Each sensor node is powered by a battery that should last for long period. This constrains wireless sensor networks in terms of energy usage. Another critical issue is secure communications ant thus, the development of cryptographic scheme is difficult and challenging task. Elliptic Curve Diffie- Hellman is security protocol for key agreement that provides both sides to have pair of public and private key for message exchange through unsecure communication channel.*

1. INTRODUCTION

Elliptic Curve Cryptography (ECC) is new cryptographic scheme independently developed by Kobiltz [1] and Miller [2]. ECC uses elliptic curve with cubic equations over finite prime (or binary) field given with Weistrass Equation (1).

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

Where a_1, a_2, a_3, a_4, $a_6 \in F_p$ and discriminate of the curve is not equal to zero.

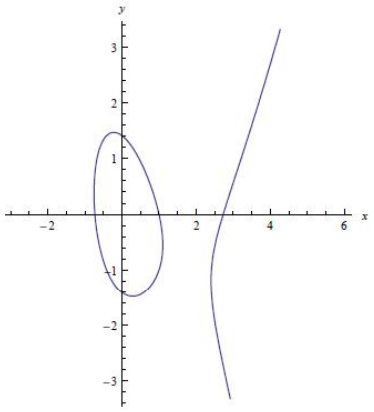

Figure 1. Elliptic Curve $y^2+xy=x^3-3x^2+2$ over finite fields

The set of point x and point y together with special so-called point to infinity O form a commutative group. Elliptic Curve Cryptography uses various arithmetic operations for key generations such as adding, subtraction, multiplication, inversion and squaring defined in [3].

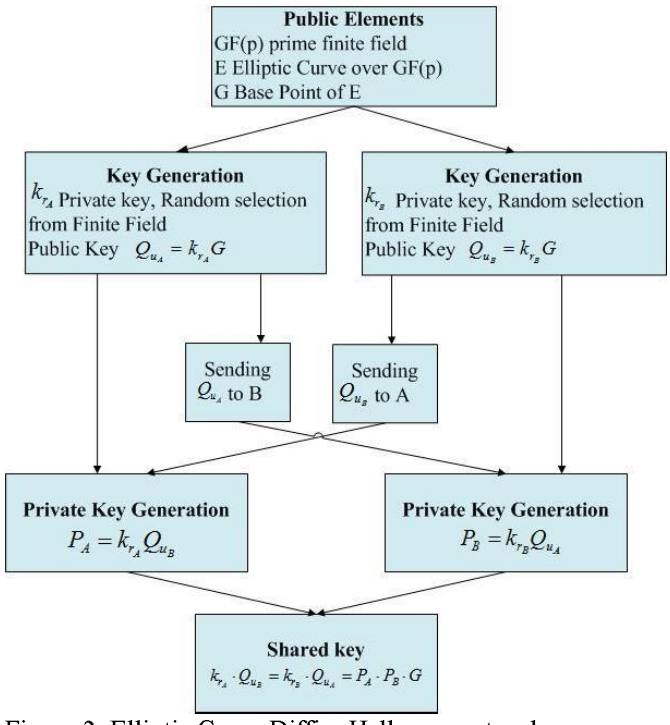

Figure 2. Elliptic Curve Diffie- Hellman protocol

In this paper, power analysis of Elliptic Curve Diffie-Hellman protocol will be represented. The analysis is done using software packet TinyECC1.0 implemented in TinyOS. Sector 2 discuses the basic of Elliptic Curve Diffie- Hellman

protocol. Sector 3 represents the simulation scenarios, obtained results and their analysis. Section 4 concludes.

2. ECDH

Elliptic Curve Diffie- Hellman [4,5] is elliptic- based key establishment scheme using public key methods based on discrete logarithm problem (ECDLP). ECDLP states that for given elliptic curve E defined over finite (or binary) fields F_p (F_2), a point from the curve $P \in E(F_p)$ of order n and point $Q \in E(F_p)$ and there is an integer $k \in [0, n-1]$ such as $Q=kP$. ECC depends on hardness of DLP and the main operation is point multiplication which in fact is point addition and point doubling. For an example, if k is scalar and equal to 23 then, kP=23P=2(2(2(2p)+p)+p)+p).

This, allows sensor nodes to establish a shared key used in algorithms for deriving a private key. Elliptic Curve Diffie- Hellman protocol is illustrated in Figure 2. The node A and B publicly agree EC domain parameters (p,a,b,G,n,h). Node A and B secretly chose random integer values k_{r_A} and k_{r_B}, so that $0 < k_{r_A}, k_{r_B} < n$. By means of scalar multiplication node A calculates $Q_{u_A} = k_{r_A} G$ while node B calculates $Q_{u_B} = k_{r_B} G$. Node A sends its public key Q_{u_A} to node B and node B sends its public key Q_{u_B} to node A. Once node A and B exchange their public keys, node A and B multiply their private keys with the received public key. Both nodes get the same value, thus the shared key is established.

3. SIMULATIONS AND POWER ANALYZE

The simulations are performed on the free open source operating system TinyOS [6]. TinyOS is flexible framework designed for wireless sensor networks which require minimum resource usage. This OS uses the discrete event simulator TinyOS SIMulator (TOSSIM) that allows debugging, testing and analyze of implemented code in nesC. Developers test not only their algorithms but also their implementations. TOSSIM supports radio and power model and has ability to simulate simultaneously thousands of sensor nodes.

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, so PowerTOSSIM includes a port of the mica2 radio stack. This allows the implemented code to be run and to use the advantage of the CC1000's power management features.

Two elliptic curves according to NIST recommendation [7] are used in these simulations. The first curve under the finite fields is secp128r1, where the key is 128-bit from which 64 are for security. The second one is secp192k1 with 192-bit key from which 96 are for security. The simulations are performed on grid and random topology with power management. The number of nodes in the network is integer from square root, starting from 4 up to 100 nodes. Simulation duration is 20 seconds and the consumed RAM and ROM is represented in Table 1.

Table 1. Consumed RAM and ROM

Compiled Ecc to build/mica2/main.exe	
1004 bytes	RAM
24154 bytes	ROM

Two types of point representation were used- affine and projective coordinates. Affine coordinates (x, y) satisfy the equation $E: y^2 = x^3 + ax + b$, where $a, b \in F_p$. The projective coordinates (x,y,z) satisfy the homogeneous Weiestrass equation for elliptic curve $E: y^2 z = x^3 + axz^2 + bz^3$, where $a, b \in F_p$. When $z \neq 0$, projective point (x, y, z) corresponds to affine point (x/y, y/z). Projective coordinates are used when field inversion is significantly more expensive then field multiplication, so the inversion is replaced with multiplication.

During the simulation the mean values of three parameters were examine: CPU Total, Radio Total and Total Energy. The parameter CPU Total represents the mean consumed energy for generating the pair of private and public key and processing of the received message. The parameter Radio Total gives the mean consumed energy or transmission of the public key and the consumed energy for the communications with other sensor nodes. Total Energy is parameter that gives the total consumed energy by the sensor node and is sum of CPU Total and Radio Total. The first simulations scenarios were performed using affine coordinates with power management under the two defined curves. Figure 3, 4 and 5 represent the mean values of parameters CPU Total, Radio Total and Total Energy and comparison between grid and random topology.

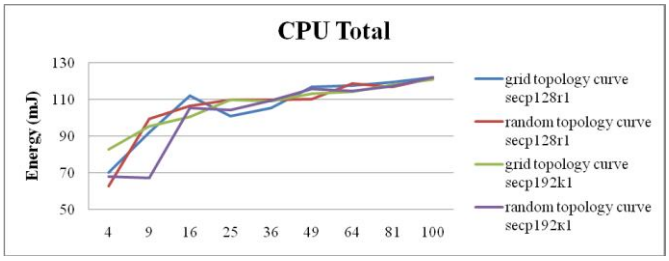

Figure 3. Comparison of CPU Total

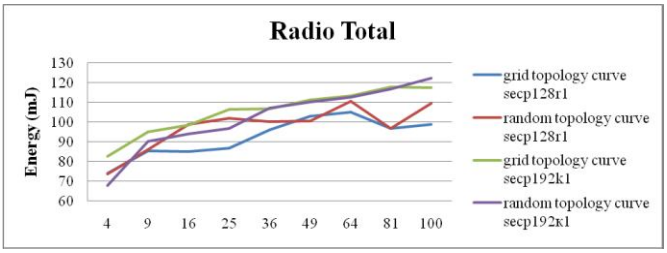

Figure 4. Comparison of Radio Total

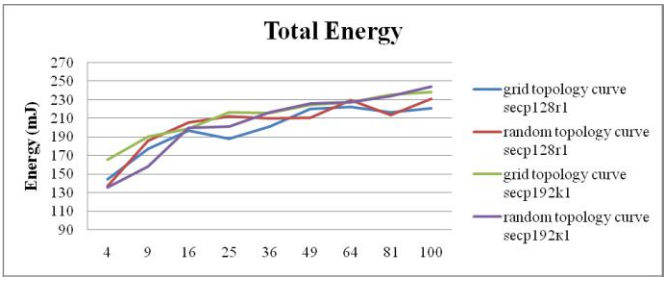

Figure 5. Comparison of Total Energy

The second simulation scenario is performed using projective coordinates with power management. In order to improve the performance of elliptic curve cryptography it was implemented Baretto reduction, algorithm for hybrid multiplication [8,9] and optimization for secp curves [10]. The simulations are performed on grid and random topology and the results are represented in Figure 6, 7 and 8.

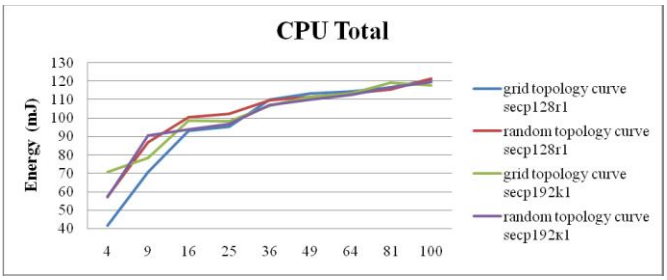

Figure 6. Comparison of CPU Total using grid and random topology

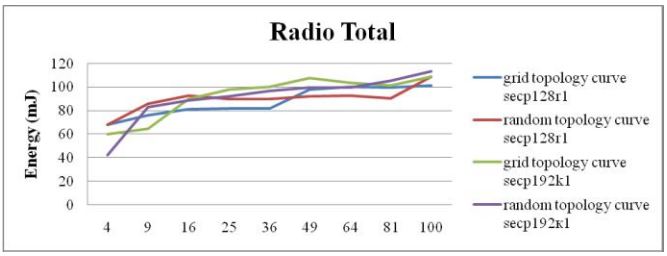

Figure 7. Comparison of Radio Total using grid and random topology

In addition, it is given comparation of the simulations results using affine and projective coordinates. Figure 9, 10 and 11 represent comparation of the parameters CPU Total, Radio Total and Total Energy using curve secp128r1, appropriately. Figure 12, 13 and 14 represent comparison of mesured parameters using affine and projective coordinates on curve secp192k1.

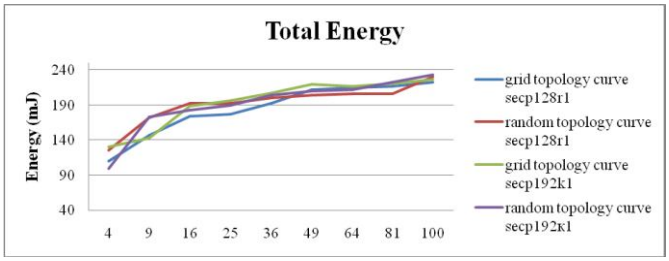

Figure 8. Comparison of Total Energy using grid and random topology

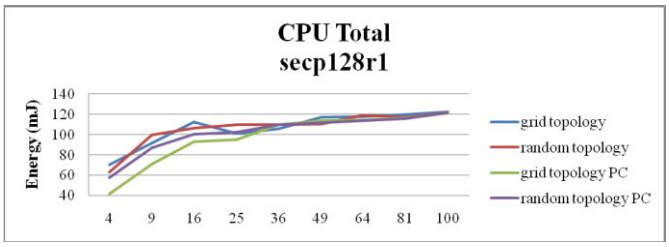

Figure 9. Comparison of CPU Total using affine and projective coordinates

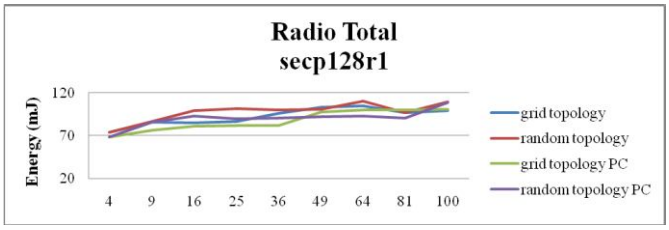

Figure 10. Comparison of Radio Total using affine and projective coordinates

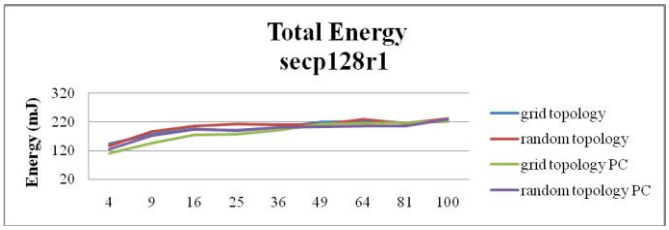

Figure 11. Comparison of Total Energy using affine and projective coordinates

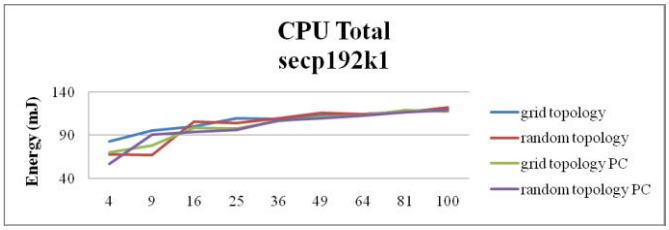

Figure 12. Comparison of CPU Total using affine and projective coordinates

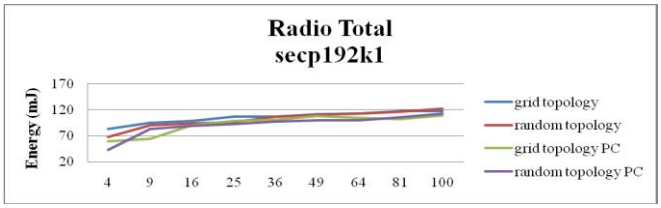

Figure 13. Comparison of Radio Total using affine and projective coordinates

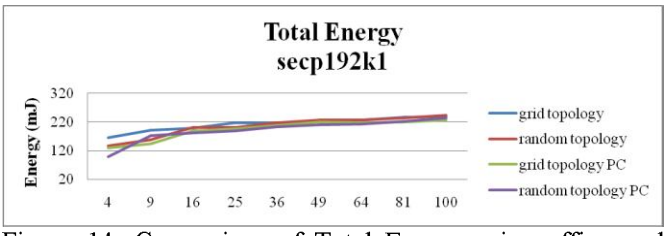

Figure 14. Comparison of Total Energy using affine and projective coordinates

4. CONCLUSION

Wireless Sensor Networks are new emerging technology that faces limitations in energy consumption and security. In this paper it is given analyze of Elliptic Curve Diffie-Hellman in terms of energy usage and power saving. ECDH is tasted on two different network topologies using two different elliptic curve using affine and projective coordinates with power management.

Performed simulations and calculations show that the performance in ECDH is improved when projective coordinates are used. The best performances are achieved if the number of node is smaller or equal to 25. The best performance are achieved for key generations .This is achieved because the implemented projective coordinates reduce the time of performing arithmetic operations especially for field inversion.

REFERENCES

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

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

[3] D. R. Habkerson, A. J. Mandezes, S. Vanstone, "Guide to Elliptic Curve Cryptography", Springer Velag, 2004.

[4] D. Aranha, R. Dahab, J. Lopez, L. Oliveira, "Efficient Implementation of Elliptic Curve Cryptography in Wireless Sensors", *Advanced in Mathematics of Communications,* Vol. 4, No 2, pp. 169- 187, 2010.

[5] D. Johnson, A. Menezes, "The Elliptic Curve Digital Signature Algorithm (ECDSA), *Technical Report CORR* 99- 34, Department of C&O, University of Waterloo, Canada.

[6] P. Levis, S. Madden, J.Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, "TinyOS: An Operating System for Sensor Networks", *Ambient Intelligence*, Springer- Verlag, pp. 115-18, 2004.

[7] Certicom, Standards for efficient cryptography- SEC 2: Recommended elliptic curve domain parameters, *Certicom Research Publications,* 2000.

[8] P. Barett, "Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor", *Proc Ann. Int'l Cryptology Conf. Advances in Cryptology* (*CRYPTO'86*), pp.311-323, 1986.

[9] N. Gura, A. Patel, A. Wander, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs, *In Proceedings of the 2004 Workshop on Cryptographic Hardware ad Embedded Systems* (CHES 2004), pp 119-132, August 2004.

[10] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography" *Springer* 2004.