ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2017.0237

A STUDY OF BILINEAR MAPS IN WIRELESS SENSOR NETWORKS

Reza Alimoradi[1], Seiied-Mohammad-Javad Razavian[2] and Ali Ramzi[3]
Department of Mathematics and Computer Science, University of Qom, Iran

Abstract

In the past, a large part of security requirements of wireless sensor networks (WSN) were fulfilled by symmetric cryptography systems. But, today by introduction of new needs in these networks and their security development, researchers of security and cryptography try to find new ways to increase efficiency and security of wireless sensor networks. Improving computational power of sensors used in wireless sensor networks made application of public key cryptography in WSN possible. Identity based cryptography is one important type of public key cryptography which using some bilinear functions called pairing functions was seriously applied. In this paper we will look at some researches done to find how to use pairings in wireless sensor networks.

Keywords

Wireless Sensor Network, Bilinear Function, Pairing, Identity Based Cryptography

1. INTRODUCTION

Wireless Sensor networks (WSN) are ad hoc networks which include a large number of small sensors and one or more base station(s). Due to their limited size and high cost, these sensors face limitations in energy consumption, memory and band width. WSN are used for gathering data and controlling the environment and are quite applicable in military or nonmilitary constructions such as battlefield monitoring, looking after the nature, traffic control and health care. They include a charged battery, a microprocessor and a radio transceiver. In recent years, in order to preserve security in WSN, symmetric cryptography systems like skipjack and RC5 were used for confidentially and identification. Symmetric cryptography systems are more appropriate for WSN compared to public key cryptography systems because they are more efficient in energy consumption and memory. However, key distribution and the number of saved keys are two main shortcomings of the symmetric cryptography systems. When unique keys are applied in WSN with n nodes, then each node must save (n-1) keys. Clearly, this is not efficient for large networks. Anyway, there will be no perfect forward security after disclosing a node's key. If a symmetric key is used, then the amount of memory needed will drastically decrease; but, in case of key exposure in one node, the whole network's security will be at risk. To remove this shortcoming, many possible key distribution schemes for the symmetric algorithm have been proposed. In general, all these schemes need pre-distributed keys which increase activities before development of the network. Therefore, asymmetric (public key) algorithms are quite valuable for key agreement and identification in WSN. Now, three types of key agreement schemes used generally in public networks will be introduced. One is a trusted server-based scheme which needs a trusted center for key agreement between the nodes. These schemes are not appropriate for WSN because of their lack of resources in energy and computations. Another scheme is based on pre-distribution of keys; it distributes key data before initiation and development of the network in all the nodes. Finally, a scheme based on public key in which cryptography is used. In the past, implementing public key cryptography on low-power systems, like sensor networks that use microprocessors and have many limitations, seemed very unlikely. Today, by making public key algorithms more efficient and also by increasing computational power of microprocessors, the farfetched dream of using public key cryptography has come true. Recently, many investigations have been done on WSN in order to make public key cryptography applicable [18]. For example, the results offered at [8, 17, 18, 23, 30, 32, 33] show that elliptic curve cryptography (ECC) can be implemented on WSN. At present, elliptic curves are used in many portable systems like PDAs, smart cards, mobile phones and pagers. Sensors of TELOSB [5], MICAz, MICA2 [4] and Imote [3] families are appropriate for implementing public key cryptography. To apply ECC on WSN, attacks of a man in the middle must be prevented by public key identification. To achieve this, public key cryptography uses public key infrastructure (PKI). Clearly, implementing PKI needs a large amount of memory, computations and communications; that makes it inefficient for WSN. Using Identity based cryptography (IBC) can solve this problem. In this type of cryptography user's public identities like their email addresses or IPs can be used as their public key. Some papers in IBE have come in [9-13].Therefore, PKI will no longer be needed. After introduction of pairing based cryptography (PBC), this type of cryptography was applicably used. In fact, IBE seems to be the best solution to use public key cryptography on WSN. Examples of IBC offered for MANET are introduced in [6, 14, 22, 35, 36]. Instances of pairing based cryptography used in limited systems like WSN are also introduced in [19, 20, 25, 27, 34].

In IBE systems, public identity of each WSN member node (sensor) i.e. each node's ID is considered that node's public key. Obviously, IBE systems need a trusted center to produce private key for the users and send it through a secure and private channel to them. In WSN, a base station (BS) can be responsible for key extract. Moreover, each node's private key can be uploaded inside each sensor before the network development. As IBE is more complicated than symmetric cryptography systems, thus, IBE is only used for production of a common key between two (or more) nodes. Public key based protocols for WSN are offered in [31]. These protocols include RSA-based identification and key agreement schemes which are named by Tiny PK. Tiny PK under NesC is implemented on MICAz 8-bit microprocessors. An exponentiation of RSA with 1024-bit key length can be done in 14.5 seconds; as a result, RSA- based structures are inappropriate for almost all applications such as WSN. Elliptic curve based cryptography (ECC) has a shorter key length compared to that of RSA (160 bits versus 1024 bits). Therefore, regarding the mentioned limitations for sensors based on 8-bit microprocessors, using ECC instead of RSA seems a lot more efficient [8]. The main action in ECC is scalar multiplication. In [17] a free software library named Tiny ECC is offered which is one of the fastest

software libraries that apply ECC on WSN. It also supports all 128,160 and 192-bit standard curves of SECG [23]. Of course the software pack offered by SUN Microsystem Company which demands a high cost, is the fastest for this purpose.

Table.1. List of abbreviations

Wireless Sensor Networks	WSN
Elliptic Curve Cryptography	ECC
Identity Based Cryptography	IBC
Pairing Based Cryptography	PBC
Public Key Infrastructure	PKI
Base Station	BS
Elliptic curve	E
Big prime numbers	p,q
Generator point G_1 of the order q	P
Subgroups of the order q	G_1,G_2
Distortion map	$\varnothing$
Multiplicative group of finite filed F_{p^2}	F_{p^2}
Pairing function	e,n_T

2. MATHEMATICAL PRELIMINARIES

Definition: Assuming q is a prime number; if G_1 and G_2 are two cyclic groups of order q; it will be a pairing e: $G_1 \times G_2 \rightarrow G_2$ with these properties:

1. *Bilinear:* $e(aP,bQ) = e(P,Q)^{ab}$ for all $P,Q \in G_1$, $a,b \in Z_q^*$;
2. *Non-Degenerate:* There exists $P,Q \in G_1$, so that $e(P,Q) \neq 1$;
3. *Computable:* There is an efficient algorithm to compute $e(P,Q)$ for all $P,Q \in G_1$.

Example: Assuming q and p are two prime numbers and $q|p^2-1$ and assuming G_1 and G_2 are two cyclic groups of the order q when G_2 is a subgroup of $F_{p^2}^*$, the modified Weil pairing is this map e: $G_1 \times G_2 \rightarrow G_2$ with above characteristics. If $\tilde{e}$ is a Weil pairing, then $e(P,P) = \tilde{e}(P, \varnothing(P))$, with $\varnothing$ as a distortion map.

Example: Assuming E with the equation $y^2 = x^3 + 1$ is defined over F_r when $p = 2(\text{mod}3)$, therefore E is super singular. $w \in F_{p^2}$ is considered as the third root of unity. Because w's order doesn't divide F_q^* order; therefore, $w \notin F_p$. Now, we consider the isomorphism of $\varnothing : E(\bar{F}_p) \rightarrow E(\bar{F}_p)$, $(x,y) \rightarrow (wx,y)$ and $\varnothing(\infty) = \infty$. Note that the order of the points P and $\varnothing(P)$ is the same.

3. ANALYZING EFFICIENCY OF THE SOFTWARE LIBRARY TINY PAIRING ON THE MICAZ SENSOR

As mentioned above, the software pack called Tiny ECC is one of the fasted software libraries existed. So, many researchers compare results of their software implementations with this software as a touchstone. An instance is the software library called Tiny Pairing [34] which supports pairing functions and so is appropriate for pairing based cryptography (PBC). The Table.2 shows efficiency of the Tiny Pairing library [29] on the MICAz sensor. In the Table.2, 10 arbitrary inputs are selected for each action; then average time of the n_T pairing is computed.

Table.2. Running time of some computations of PBC on MICAz by Tiny Pairing [27]

	Time (sec)
Hash-to-Point (16 bytes msg)	0.89
Point compression Point decompression	0.38 0.38
Point Scalar multiplication Point Scalar multiplication Point Scalar multiplication	7.75 2.50 2.45
n_T pairing	5.32

Using these conclusions, there comes a comparison between implementation of some pairing based schemes by the Tiny Pairing with that of some elliptic curve based schemes by the Tiny ECC on the MICAz sensor.

Table.3. Implementation of some cryptography schemes on MICAz [34]

Library	**Tiny ECC**	**Tiny Pairing**
Scheme	ECIES	BF IBE
Set up (sec)	-	3.22
Key generation (sec)	-	2.83
Encryption (sec)	61.40	10.61
Decryption (sec)	31.87	5.35
RAM (bytes)	150	392
ROM (bytes)	12,442	22,598
Size of public key/ID	160 bit after compressing	Arbitrary bit string

Table.4. Implementation of some signature schemes on MICAz [34]

Library	**Tiny ECC**	**Tiny Pairing**	
Scheme	ECDSA	BLS SS [1]	BBSS [2]
Set up(sec)	0	-	-
Key generation (sec)	-	3.18	12.33
Signing (sec)	30.72	4.08	3.0
Verification (sec)	61.80	12. 62	11.03
RAM (bytes)	152	382	392
ROM (bytes)	10,180	22,632	19,742
Size of signature (bit)	320	160	312

4. COMPARING DIFFERENT TYPES OF SENSORS AND SOFTWARE PACKAGES

The comparison is between different types of sensors like MICA2 which is a subset of MICAz and Imote2 and Sky Tmote. In this analysis two types of pairing functions $e(P,Q)$ and $n_T(P,Q)$ are implemented by software packs Nano ECC [26], Tiny Tate [19], Tiny PBC [20] and the one introduced in [27].

Table.5. implementation of pairings defined on F_p, F_{2^m} for Imote2 sensors [27]

	Imote2(13MHz)		**Imote2(104MHz)**	
Pairing	$n_T(P,Q)$	$e(P,Q)$	$n_T(P,Q)$	$e(P,Q)$
Time	0.46s	0.62s	0.06s	0.08s
ROM	29.55KB	44.40KB	29.55KB	44.40B
Consumed energy	12.12mJ	16.34mJ	3.76mJ	5.02mJ

Table.6. Implementation of pairings defined on F_p, F_{2^m} for Tmote Sky and MICAz sensors [27]

	MICA2		**Tmote Sky**	
Parings	$n_T(P,Q)$	$e(P,Q)$	$n_T(P,Q)$	$e(P,Q)$
Time	2.66s	7.43s	1.71s	4.61s
ROM	47.41KB	60.9KB	23.66KB	34.88B
Consumed energy	62.73mJ	175.65mJ	17.70mJ	50.89mJ

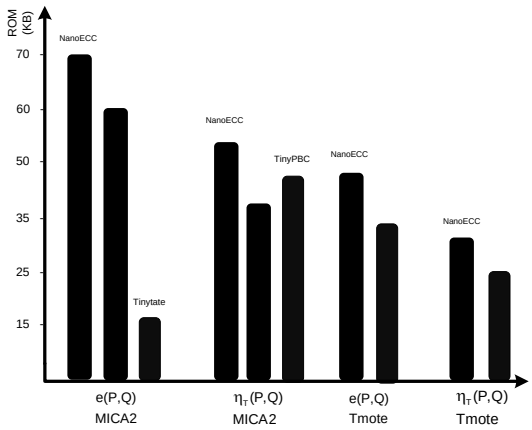

Fig.1. Comparing implementation of two pairings on some sensors regarding with ROM [27]

5. ECDSA ALGORITHM OF THE SENSORS MICAZ, TELOSEB AND IMOTE

Using selected curves secg defined on 128,160,192-bit prime fields, ECDSA algorithm is implemented. Sensors under experiment include MICAz, TeloseB and Imote2. The Table.7 - Table.16 shows the results of implementing ECDSA on MICAz, TeloseB and Imote2.

Note that window length in scalar multiplication method is about w=4.

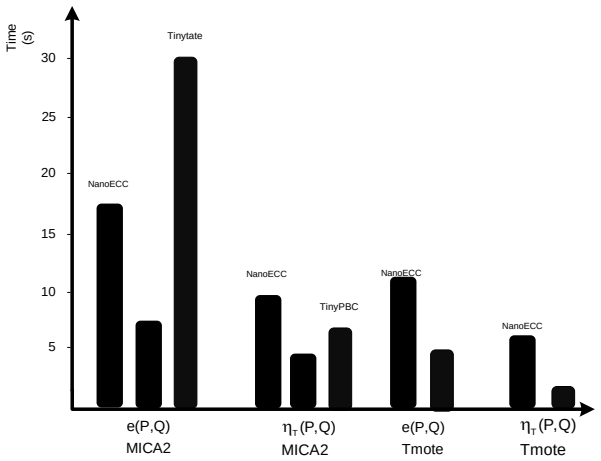

Fig.2. Comparing implementation of two pairings on some sensors regarding with time duration [27]

Table.7. Running time of ECDSA on MICAz for w=4 [15]

Curve	**Set up**	**Sign**	**Verification**
secp 128r1	2.522	1.923	2.418
secp 128r2	2.518	2.069	2.674
secp 160k1	3.553	2.059	2.441
secp 160r 1	3.548	1.925	2.433
secp 160r2	3.543	2.066	2.615
secp 192k1	4.992	3.070	3.612
secp 192r1	4.992	2.991	3.776

Table.8. Running time of ECDSA on TeloseB for w=4 [15]

Curve	**Set up**	**Sign**	**Verification**
secp 128r1	3.861	4.059	5.056
secp 128r2	3.847	4.325	5.618
secp 160k1	5.208	4.433	5.209
secp 160r1	5.225	4.361	5.448
secp 160r2	5.197	4.457	5.609
secp 192k1	7.190	6.695	7.840
secp 192r1	7.204	6.651	8.331

Table.9. Running time of ECDSA on Imote2 (104MHz) for w=4 [15]

Curve	**Set up**	**Sign**	**Verification**
secp 128r1	0.136	0.255	0.317
secp 128r2	0.136	0.255	0.360
secp 160k1	0.151	0.180	0.219
secp 160r1	0.148	0.167	0.205
secp 160r2	0.151	0.187	0.233
secp 192k1	0.190	0.265	0.308
secp 192r1	0.200	0.265	0.325

Table.10. Running time of ECDSA on Imote2 (416 MHz) for w=4 [15]

Curve	Set up	Sign	Verification
ecp 128r1	0.035	0.065	0.083
secp 128r2	0.035	0.069	0.095
secp 160k1	0.038	0.049	0.060
secp 160r1	0.037	0.042	0.054
secp 160r2	0.038	0.047	0.060
secp 192k1	0.050	0.067	0.079
secp 192r1	0.050	0.068	0.084

The Table.10 shows energy required for computing ECDSA on these 3 sensors for a specific curve and various window lengths.

Table.11. Required energy to compute ECDSA for the curve secp160r1 [15]

w	MICAz		TeloseB	
	Sign	Verification	Sign	Verification
2	52.9	58.4	27.5	29.4
4	46.2	58.4	23.5	29.4
8	-	-	-	-

Table.12. Required energy to compute ECDSA for the curve secp160r1 [15]

w	Imote2					
	13MHz		104MHz		416MHz	
	Sign	Verification	Sign	Verification	Sign	Verification
2	2.56	2.72	0.32	0.34	0.08	0.10
4	2.19	2.72	0.28	0.34	0.07	0.09
8	-	-	0.24	0.34	0.06	0.09

Energy needed for signature stage and verification of ECDSA algorithm for secp160r1 curves on different sensors has briefly come in these charts.

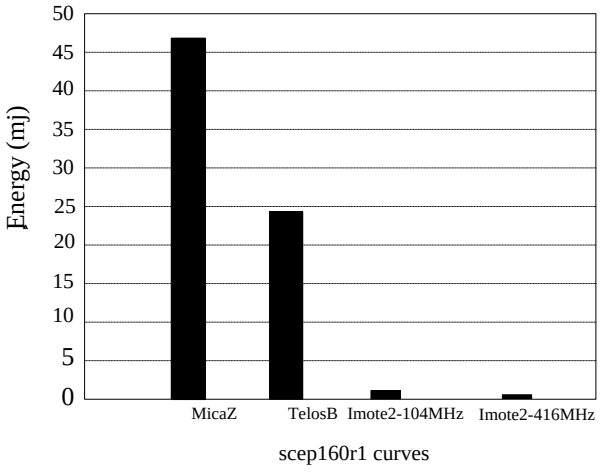

Fig.3. Required energy for the signature stage of ECDSA on some sensors [15]

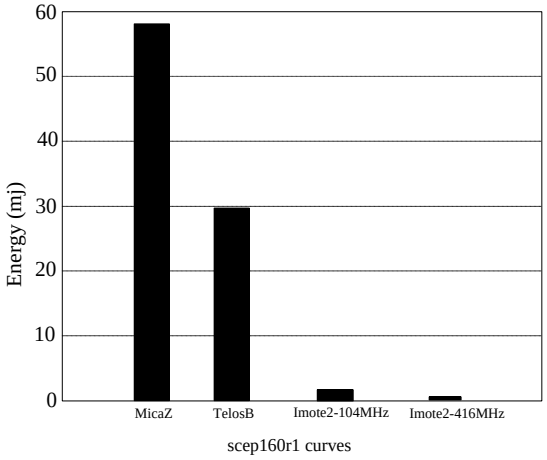

Fig.4. Required energy for the verification stage of ECDSA on some sensors [15]

Also, Time needed for the initiating stages and signing ECDSA algorithm for various curves on various sensors is explained here.

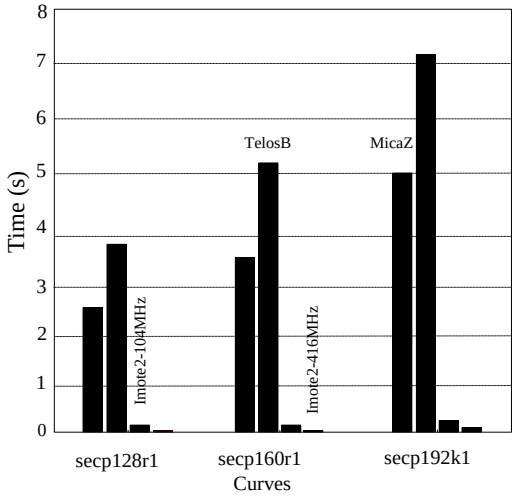

Fig.5. Running time of the initiative stage of ECDSA on some sensors [15]

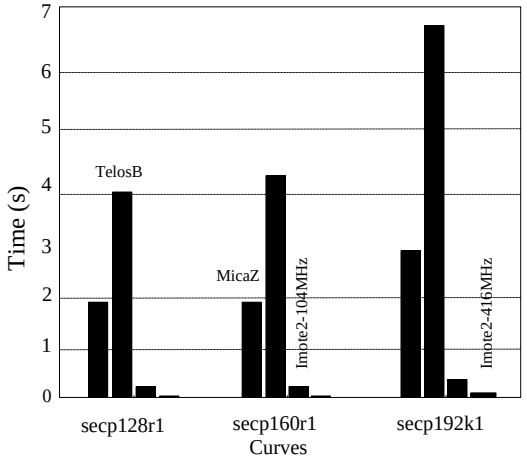

Fig.6. Running time of the signature stage of ECDSA on some sensors [15]

These findings show the modern sensors of the type Imote2 are more efficient to implement elliptic curve based cryptography than MICAz and TeloseB sensors.

6. IMPLEMENTING PAIRING FUNCTIONS ON MICAZ, TELOSEB AND IMOTE2 SENSORS

As mentioned earlier, computing a pairing function like Tate function is expensive. Therefore Imote2 is more efficient than MICAz and TeloseB. Here some results of the researches about implementing Tate function on Imote2 are explained. This analysis explicates size of the written program, time and energy required for implementing Tate function on Imote2 for super singular curves defined on 192, 512-bit prime finite field. Size of the software program written on Imote2 for computing Tate function is explained in Table.12. As RAM memory of Imote2 is 32MB, so, both curves for the size of the program are acceptable. This is also true about ROM memory.

Table.13. Size of program for computation of Tate pairing on Imote2 [15]

Curve	ROM	RAM
ss192k2	13,512	434
ss512k2	13,844	1,034

Table.14. Running time of Tate pairing on Imote2 (104MHz) [15]

Curve	104MHz		
	Miller	Final exponential	Sum
ss192k2	0.459	0.032	0.491
ss512k2	4.405	0.154	4.559

Table.15. Running time of Tate pairing on Imote2 (416MHz) [15]

Curve	416MHz		
	Miller	Final exponential	Sum
ss192k2	0.115	0.008	0.123
ss512k2	1.575	0.055	1.629

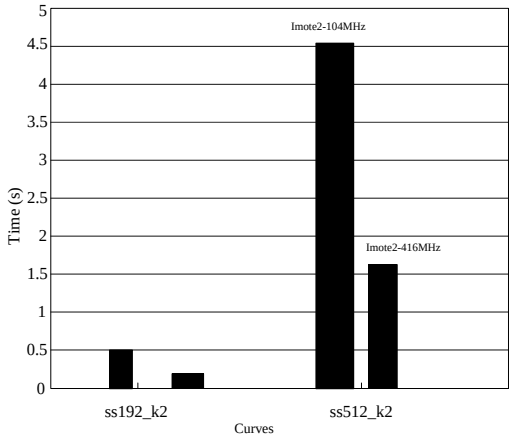

Fig.7. Running time of Tate pairing on Imote2 [15]

Computing Tate function has two main parts; one is miller algorithm and the other is exponentiation at the end of the Tate algorithm. Time required for running Tate algorithm on Imote2 is shown in Table.12 and Table.13. With regard to time of running Tate algorithm for the 512- bit curve which contrary to the 192-bit curve is more secure, this algorithm can be used in actual application of WSN. To prevent DoS attack which is a possible threat, projective coordinates can be used. This makes the computation time, 10 times faster. The coming table and chart show energy required for implementing Tate algorithm on Imote2.

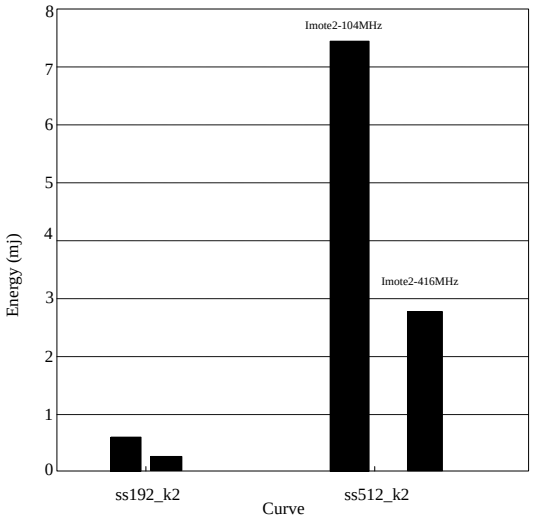

Fig.8. Energy consumption of Tate pairing on Imote2 [15]

Table.16. Energy consumption of Tate pairing on Imote2 [15]

Curve	104MHz	416MHz
ss192k2	0.80	0.20
ss512k2	7.47	2.67

Regarding the small amount of energy needed for computing Tate algorithm on Imote2 compared to the energy needed for ECDSA algorithm verification on TeloseB and MICAz, thus, energy consumption of Tate algorithm on Imote2 is acceptable.

7. CONCLUSION

Computing pairing functions is very expensive. Therefore, the powerful sensor Imote2 is more efficient than MICAz and TloseB sensors. The pack introduced in [27] is functional than Nano ECC, Tiny Tate, Tiny PBC.

REFERENCES

[1] D. Boneh, H. Shacham and B. Lynn, "Short Signatures from the Weil Pairing", *Proceedings of International Conference on Advances in Cryptology*, pp. 514-532, 2001.

[2] D. Boneh and X. Boyen, "Short Signatures without Random Oracles and the SDH Assumption in Bilinear Groups", *Journal of Cryptology*, Vol. 21, No. 2, pp. 149-177, 2008.

[3] Crossbow Technology, "High-Performance Wireless Sensor Network Node", Available at:

http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf.

[4] Crossbow Technology, "Wireless Measurement System", Available at: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf.

[5] Crossbow Technology, "TELOSB Mote Platform", Available at: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[6] H. Deng, A. Mukherjee and D. Agrawal, "Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks", *Proceedings of International Conference on Information Technology: Coding and Computing*, 107-111, 2004.

[7] W. Diffiee and M. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, 1976.

[8] N. Gura, A. Patel, A. Wander, H. Eberle and S.C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs", *Proceedings of 6th International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 119-132, 2004.

[9] M.H. Dehkordi and R. Alimoradi, "Zero-Knowledge Identification Scheme Based on Weil Pairing", *Lobachevskii Journal of Mathematics*, Vol. 30, No. 3, pp. 203-207. 2009.

[10] M.H. Dehkordi and R. Alimoradi, "A New Batch Identification Scheme", *Discrete Mathematics, Algorithms and Applications*, Vol. 1, No. 3, pp. 369-376, 2009.

[11] M.H Dehkordi and R Alimoradi, "Authenticated Key Agreement Protocol", *China Communications*, Vol. 7, No. 5, pp. 1-8, 2010.

[12] M.H Dehkordi and R Alimoradi, "Identity-Based Multiple Key Agreement Scheme", *KSII Transactions on Internet and Information Systems*, Vol. 5, No. 12, pp. 2392-2402, 2011.

[13] M. H Dehkordi and R Alimoradi, "Certificateless Identification Protocols from Super Singular Elliptic Curve", *Security and Communication Networks*, Vol. 7, No. 6, pp. 979-986, 2014.

[14] K. Hoeper and G. Gong, "Identity-Based Key Exchange Protocols for Ad Hoc Networks", *Proceedings of the Canadian Workshop on Information Theory*, pp. 127-130, 2005.

[15] P. T. Kampanakis, "Identity-Based Cryptography: Feasibility and Applications in Next Generation Sensor Networks", Master of Science Thesis, North Carolina State University, 2007.

[16] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks", *Proceedings of 7th International Conference on Information Processing in Sensor Networks*, pp. 245-256, 2008.

[17] J. Lopez, D. Aranha, D. Camara, R. Dahab, L. Oliveira and C. Lopes, "Fast Implementation of Elliptic Curve Cryptography and Pairing Computation for Sensor Networks", *Proceedings of 13th Workshop on Elliptic Curve Cryptography*, pp. 117-121, 2009.

[18] 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, pp. 22-23, 2008.

[19] L.B. Oliveira, D.F. Aranha, E. Morais, F. Daguano, J. Lopez and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes", *Proceedings of 6th IEEE International Symposium on Network Computing and Applications*, pp. 318-323, 2007.

[20] L.B. Oliveira, M. Scott, J. Lopez and R. Dahab, "TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks", *Proceedings of 5th International Conference on Networked Sensing Systems*, pp. 173-180, 2008.

[21] E. Ozturk, B. Sunar and E. Savascedil, "Low-Power Elliptic Curve Cryptography using Scaled Modular Arithmetic", *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 92-106, 2004.

[22] N. Saxena, G. Tsudik and J.H. Yi, "Identity-Based Access Control for Ad Hoc Groups", *Proceedings of* 7th *International Conference on Information Security and Cryptology*, pp. 362-379, 2004.

[23] Standards for Efficient Cryptography Group, Available at: http://www.secg.org.

[24] S.C. Seo, D.G. Han and S. Song, "TinyECCK: Efficient Elliptic Curve Cryptography Implementation over GF(2m) on 8-bit Micaz Mote", *IEICE Transactions on Information and Systems*, Vol. 91, No. 5, pp. 1338-1347, 2008.

[25] M. Shirase, Y. Miyazaki, T. Takagi, D.G. Han and D. Choi, "Efficient Implementation of Pairing Based Cryptography on a Sensor Node", *IEICE Transactions on Information and Systems*, Vol. 92, No. 5, pp. 909-917, 2009.

[26] P. Szczechowiak, L. Oliviera, M. Scott, M. Collier and R. Dahab, "NanoECC: Testing the limits of Elliptic Curve Cryptography in Sensor Networks", *Proceedings of European Conference on Wireless Sensor Networks*, Vol. 4913, pp. 305-320, 2008.

[27] P. Szczechowiak, A. Kargl, M. Scott and M. Collier, "On the Application of Pairing based Cryptography to Wireless Sensor Networks", *Proceedings of 2nd ACM Conference on Wireless Network Security*, pp. 1-12, 2009.

[28] TinyOS, Available at: https://wiki2.org/en/TinyOS.

[29] TinyPairing library for wireless sensor networks, Available: http://www.cs.cityu.edu.hk/_ecc/TinyPairing.

[30] H. Wang and Q. Li, "Efficient Implementation of Public Key Cryptosystems on MICAz Motes", *Proceedings of 8th International Conference on Information and Communications Security*, pp. 519-528, 2006.

[31] R. Watro, D. Kong, S.F. Cuti, C. Gardiner, C. Lynn and P. Kruus, "TinyPK: Securing Sensor Networks with Public Key Technology", *Proceedings of 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 59-64, 2004.

[32] T. Wollinger, J. Pelzl, V. Wittelsberger, C. Paar and G. Saldamli, "Elliptic and Hyper Elliptic Curves on Embedded Platform", *ACM Transactions in Embedded Computing Systems*, Vol. 3, No. 3, pp. 509-533, 2004.

[33] T. Wollinger, "Software and Hardware Implementation of Hyperelliptic Curve Cryptosystems", PhD Dissertation, Department of Electrical Engineering and Information Sciences, Ruhr University Bochum, 2004.

[34] X. Xiong, D.C. Wong, and X. Deng, "Tiny Pairing: Computing Tate Pairing on Sensor Nodes with Higher Speed

and Less Memory", *Proceedings of 8th IEEE International Symposium on Network Computing and Applications*, pp. 187-194, 2009.

[35] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Securing Mobile Ad Hoc Networks with Certificateless Public Keys", *IEEE Transactions on Dependable and Secure Computing*, Vol. 3, No. 4, pp. 386-399, 2006.

[36] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Location-based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 2, pp. 247-260, 2006.