ANNALS of Faculty Engineering Hunedoara – International Journal of Engineering
Tome XVIII [2020] | Fascicule 2 [May]

ANNALS of Faculty Engineering Hunedoara – International Journal of Engineering
ISSN 1584 - 2665 (printed version); ISSN 2601 - 2332 (online); ISSN-L 1584 - 2665

[1.]Jaykumar Shantilal PATEL

STRATEGIC PLAN TO REDUCE COMPUTATIONAL COST OVER MOBILE WIRELESS NETWORK TO ACQUIRE ROBUST SECURITY

[1.]Chaudhari Technical Institute, Gujarat Technological University, Gujarat, INDIA

Abstract: The proposed research shows the strategic plan to reduce computation cost for mobile wireless network. The research incorporates the base concept of self-healing and mutual healing to save the computation. The proposed work compared with various security features of existing schemes and proved that proposed scheme has significant less computation overhead. The proposed research incorporates the concept of bilinear paring based on elliptic curve points and group based session by session communication to reduce the computation.
Keywords: Computation cost, security, self-healing, mutual-healing, elliptic curve, bilinear pairing

1. INTRODUCTION

In the past several years security in mobile wireless networks has attracted marvelous attention. The broadcast nature, deployment in the hostile environment, limited battery, and limited computational cost are the major constraints in mobile wireless network. The researchers in mobile wireless network security have proposed various security schemes for resource constraint mobile wireless network. A numbers of secure routing protocols [1-4], aggregation protocols [5-10], group formation [11-13] has been proposed by several researchers in the field of mobile wireless network.To acquire robust security asymmetric cryptography is used. The problem with asymmetric cryptography is that it is typically too computationally intensive for the individual nodes in a network. This is true in the general case; however, [14-17] show that it is feasible with the right selection of algorithms. The scrutinized public key algorithms include RSA [18] and Elliptic Curve Cryptography (ECC) [19, 20]. The recent trend for key selection focuses on RSA and ECC algorithms. The main attraction to select ECC is that it proposes the same security for a far smaller key size [21]. Hence it reduces computational overhead.

— Self-healing

Self-healing allows the user to recover the lost broadcasted message by them self without demanding the additional transmission from the group manager, can save the communication cost, reduce the network traffic, as well as reduce the chance of exposure through network traffic analysis. The self-healing can only support the fix number of message loss and it does not support when last message has lost. The situation overcomes with introducing the concept of mutual-healing.

— Mutual healing

The mutual-healing overcomes the problem of self-healing with the help of their neighbor to recover session key. The proposed work uses mutual-healing based on bilinear pairing. The bilinear paring operations provide robust security due to the discrete logarithm.

2. PROPOSED SCHEME

Proposed scheme calculate computation overhead with self-healing and mutual healing approach. Self-healing approach for proposed scheme

» Paring Operation [For $e(P_{Pub}, U_i)$] =Tp
» Paring Operation [For $e(U, S_i)$]=Tp
» Hash Function [For $V_w \oplus H_2(e(P_{pub}, r_w.Q))$]=Th=Tp + Tp + Th, =2Tp + Th

Self-healing approach for Tian's scheme [24]:

- Paring Operation [For e(U1,x1Si)]=Tp
- Paring Operation [For e(Ppub,∑xiUi)]=Tp
- Hash Function [For Vj ⊕ H2(e(U1,x1Si).e(Ppub,∑xiUi))]=Th
- Scalar multiplication=Ts= Tp + Tp + Th + d(Ts) [For d number of user in session], =2Tp + Th + dTs

Mutual-healing approach for proposed scheme:
Ux Calculation:

- Encryption [For {lx}Kc]=Te
- Decryption [For {ly}Kc – When get response]=Td
- PRF=Tm

Ux Calculation for Key Confirmation

- Encryption [For {Ny+1}Kt]=Te
- PRF=Tm
- Total For Ux:=2Te + 2Tm + Td

Uy Calculation

- Encryption [For {ly}Kc]=Te
- Decryption [For {lx}Kc – To see the location of lx] =Td
- PRF=Tm

Uy Calculation for Respond to Key Confirmation

- Decryption [For {Ny+1}Kt]=Te
- PRF=Tm
- Total For Uy:=2Td + 2Tm + Te
- Total= Ux + Uy = 2Te + 2Tm + Td + (2Td + 2Tm + Te – if responding node)

Mutual-healing approach for Tian's scheme:
Ux Calculation (Requesting Node)

- Paring Operation [For e(LKj,H(IDi||li)) –For Shared key Kji]=Tp
- Hash Function [For H(IDi || li) –For Shared key Kji]=Th
- Decryption [For (Bt)Kji – When get response]=Td
- Total For Ui:=Tp + Th + Td

Uy Calculation(Responding Node)

- Paring Operation [For e(LKj,H(IDi||li)) –For Shared key Kji]=Tp
- Hash Function [For H(IDi || li) –For Shared key Kji]=Th
- Encryption [For (Bt)Kji – When get request]=Te
- Total For Uj:=Tp + Th + Te
- Total = Tp + Th + Td + (Tp + Th + Te – if responding node)

3. SECURITY ANALYSIS

The security analysis consists five different properties - Forward and backward secrecy, Resistance to Collusion, Resistance to Impersonation, Secured Node Location, Mutual Authentication, and Key Confirmation.

— Forward Secrecy and Backward Secrecy

Forward secrecy ensures that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future. Backward secrecy ensures that compromise of a session key does not reveal the past session keys or long-term keys.

— Mutual Authentication

Two nodes participating in a mutual healing session can mutually authenticate each other, since the communication between two parties contain code generated from a common secret using PRF().

— Location Secrecy

Location is secured in terms of providing confidentiality, hence during mutual-healing process unauthorized node may note able to get request or response.

— Key Confirmation

Key confirmation uses the concepts of nonce to confirm the responding node, that the correct message has been received by requesting node.

4. COMPARATIVE SECURITY ANALYSIS

Table 1 gives the comparison of major security features of various security schemes with proposed scheme including Tian et al. [24] scheme. The proposed scheme uses the base concept of Tian et al. [24] scheme.

Table 1: Comparison of major security features

	Lee et al. [22]	Varadharajan et al. [23]	Tian et al. [24]	Proposed Scheme
Forward Secrecy	No	No	Yes	Yes
Backward Secrecy	No	No	Yes	Yes
Mutual Authentication	No	No	No	Yes
Location Secrecy	No	No	No	Yes
Key Confirmation	No	No	No	Yes

5. PERFORMANCE ANALYSIS

The performance of proposed scheme is measured using computation cost. A node after receiving the broadcast message, takes the jth component and for computing e(PPub, rt.Q), it needs to perform two bi-linear pairing operations, one for e(PPub, Ux) and other for e(U, Sx). Then it computes the hash H2(e(PPub, rt.Q)) and finally extracts key Kj using an XOR operation. The hash and XOR computation cost is negligible in comparison to the bi-linear pairing computation. So, if Tp is the cost of bi-linear pairing computation, then self-healing and key extraction operation takes only 2 * Tp. In Tian et al. [24], Group manager needs to define a |Gj-1| X |Gj| matrix and compute |Gj-1| additional ECC points using public keys of the members of current communication group, in order to construct the broadcast message. Secondly, the responding node encrypts the requested broadcast message with a key generated from the location based key using bi-linear pairing operation. The requesting node needs to calculate the same key again using bi-linear pairing operation. In my proposal, I avoid the need of any matrix, additional ECC points computations and bi-linear pairing operations for authentication. For mutual healing, one symmetric key encryption, and one PRF() computation for request message, one symmetric key decryption, one simple comparison to check Euclidean distance and one PRF() to verify response, and one symmetric key encryption for key confirmation. In case node is responder, then it needs one symmetric key decryption, one simple comparison to check Euclidean distance and one PRF() to verify request, then for response one symmetric key encryption, and one PRF() computation. When compared with Tian et al. [24], I find that computation cost is greatly reduced, as now a node does not require to solve system of linear equations and also it could avoid doing scalar multiplication with respect to all other nodes in the group in order to recover a key. Following Table 2 shows the comparative performance analysis of proposed scheme and Tian et al [24] scheme.

Table 2: Comparative Performance Analysis

Attributes	Computation Overhead	
	Self-healing	Mutual-healing
Tian et al. [24]	2Tp+Th+dTs	Tp+Th+Td+(Tp+Th+Te-if responding node)
Proposed Scheme	2Tp+Th	2Te+2Tm+Td+(2Td+2Tm+Te-if responding node

6. RESULTS AND ANALYSIS

Computation overhead is measure in from of timerequiring for the execution of specific equation. The Computational overhead based on Time to perform paring, Time to perform hash, Time to perform scalar multiplication, number of nodes, time to perform encryption, time to perform decryption and time to perform MAC.

— Pairing Time (Tp)

The pairing time for proposed scheme is far less compare with the Tian et al. [24]. When paring time increased simultaneously the computational cost of Tian et al. [24] as well as proposed scheme is increase.

Table 3: Computational Overhead for Paring Time

Paring Time	Computation Overhead (Bits)	
	Tian et al. [24]	Proposed Scheme
20	1680	220
25	1700	230
30	1720	240
35	1740	250
40	1760	260
45	1780	270

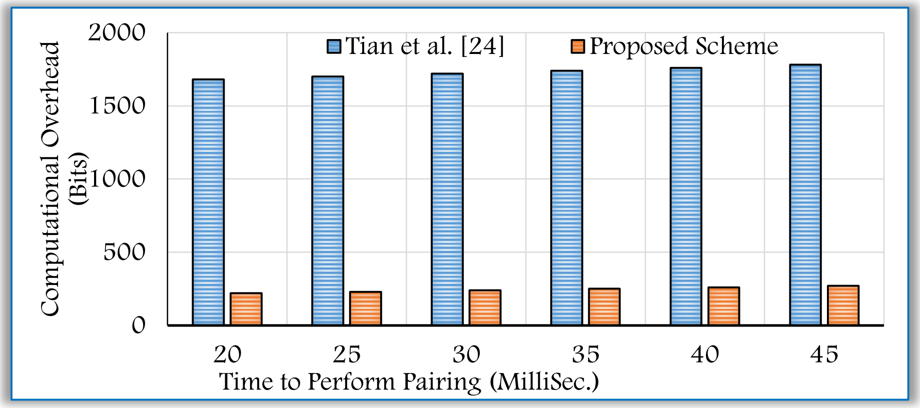

Figure 1: Computational Overhead vs. Paring Time

— **Hash Time (Th)**

The hash function execution time for proposed scheme is far less compare with the Tian et al. [24]. When hash time increased simultaneously the computational cost of Tian et al. [24] as well as proposed scheme is increase.

Table 4: Computational Overhead for Hash Time

Hash Time	Computation Overhead (Bits)	
	Tian et al. [24]	Proposed Scheme
20	1680	220
25	1695	225
30	1710	230
35	1725	235
40	1740	240
45	1755	245

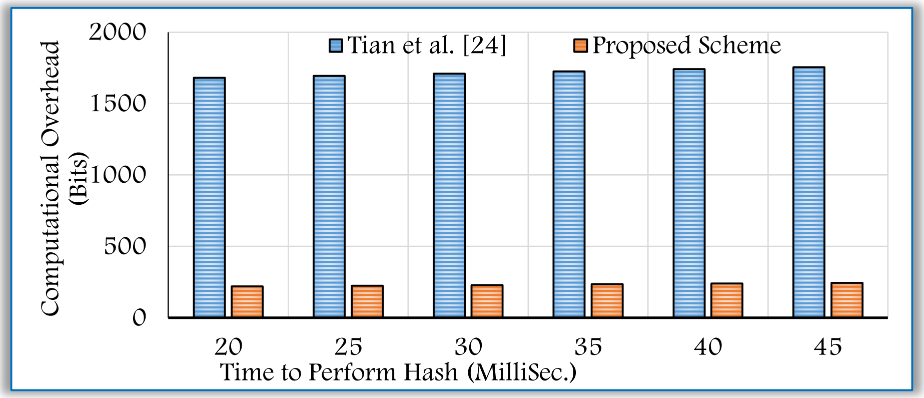

Figure 2: Computational Overhead vs. Hash Time

— **Scalar Multiplication Time (Ts)**

The Scalar multiplication is a part of Tian et al. [24], so the value for proposed scheme may remain constants. When Number of scalar multiplication increased simultaneously the computational cost also increased in Tian et al. [24]. While the value for proposed scheme may remain constants.

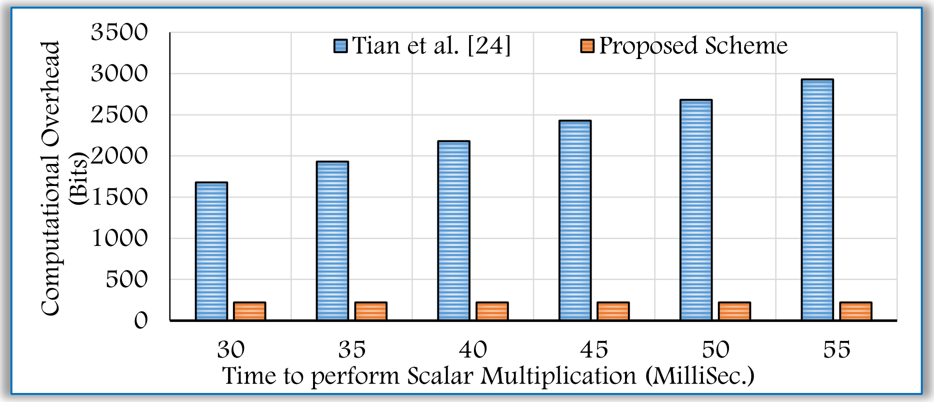

Figure 3: Computational Overhead vs. Scalar Multiplication

Table 5: Computational Overhead for Scalar Multiplication

Scalar Multiplication	Computation Overhead (Bits)	
	Tian et al. [24]	Proposed Scheme
30	1680	220
35	1930	220
40	2180	220
45	2430	220
50	2680	220
55	2930	220

— **Number of Nodes (d)**

When number of nodes increases the computational overhead increases in Tian et al. [24]. The proposed scheme has constant values for the computational overhead.

Table 6: Computational Overhead for Number of Nodes

Number of Nodes	Computation Overhead (Bits)	
	Tian et al. [24]	Proposed Scheme
50	1680	220
55	1830	220
60	1980	220
65	2130	220
70	2280	220
75	2430	220

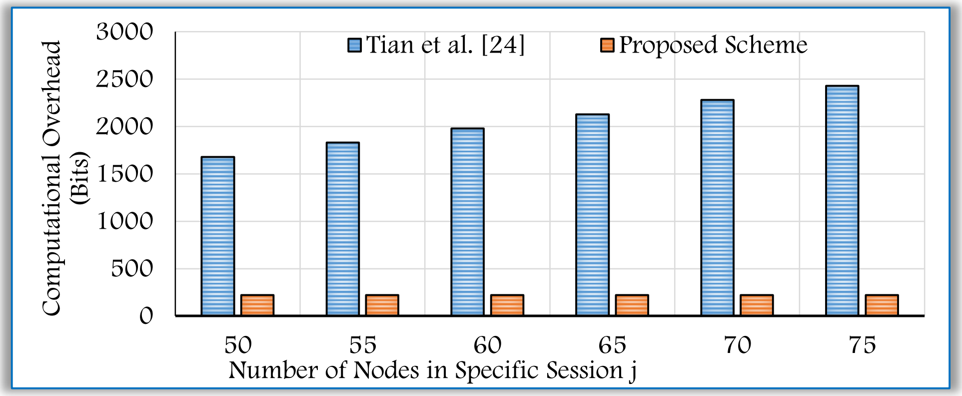

Figure 4: Computational Overhead vs. Number of Nodes

— **Encryption Time (Te):** When encryption time increased simultaneously the computational cost of Tian et al. [24] as well as proposed scheme is increase.

Table 7: Computational Overhead for Encryption Time

Encryption Time	Computation Overhead (Bits)	
	Tian et al. [24]	Proposed Scheme
20	1680	220
25	1685	235
30	1690	250
35	1695	265
40	1700	280
45	1705	295

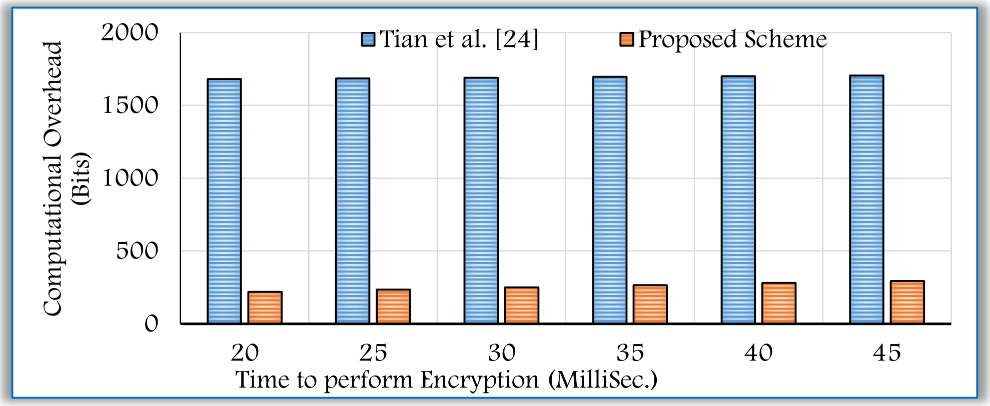

Figure 5: Computational Overhead vs. Encryption Time

— **Decryption Time (Td):** When decryption time increased simultaneously the computational cost of Tian et al. [24] as well as proposed scheme is increase.

Table 8: Computational Overhead for Encryption Time

	Computation Overhead (Bits)	
Decryption Time	Tian et al. [24]	Proposed Scheme
20	1680	220
25	1685	235
30	1690	250
35	1695	265
40	1700	280
45	1705	295

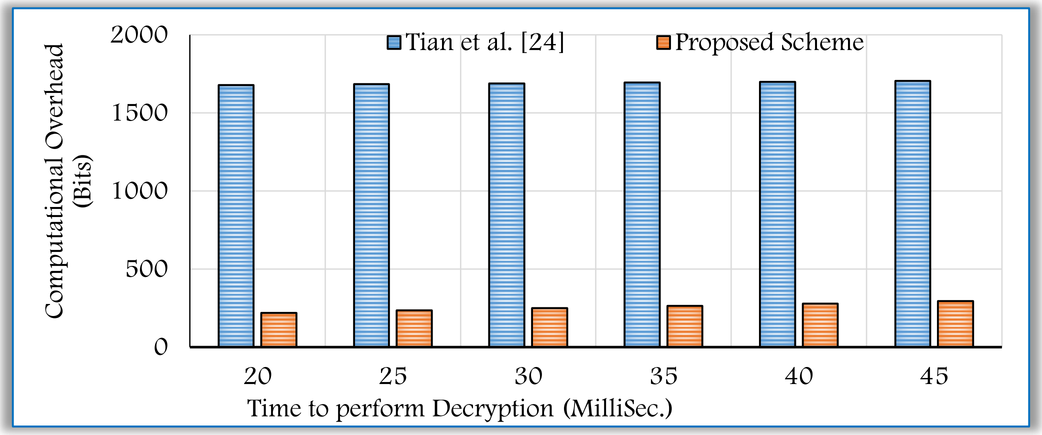

Figure 6: Computational Overhead vs. Decryption Time

— **MAC Time (Tm):** When MAC time increased for the authentication simultaneously the computational cost of Tian et al. [24] as well as proposed scheme is increase.

Table 9: Computational Overhead for MAC Time

	Computation Overhead (Bits)	
MAC Time	Tian et al. [24]	Proposed Scheme
10	1680	220
15	1680	240
20	1680	260
25	1680	280
30	1680	300
35	1680	320

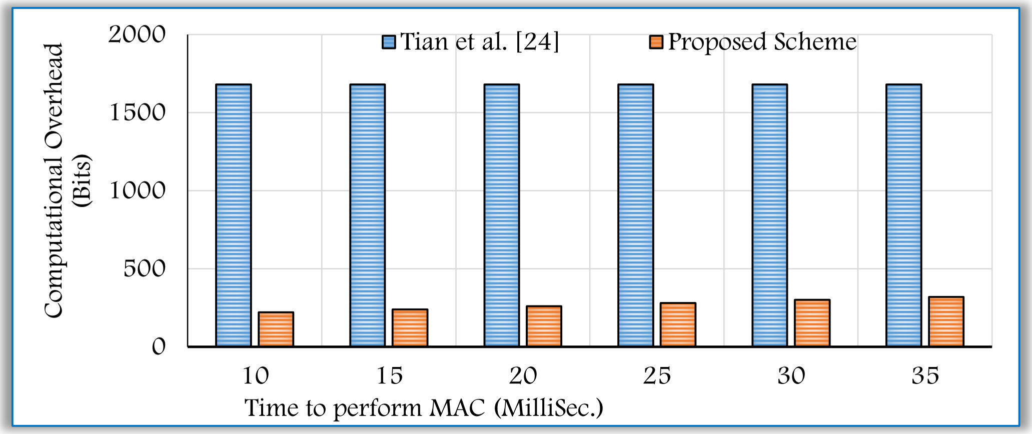

Figure 7: Computational Overhead vs. MAC Time

7. CONCLUSION

The proposed research offer robust security with less computation overhead. The research uses the concepts of self-healing and mutual healing. The research incorporate elliptic curve cryptography over bilinear pairing to ensure robust security with less computation cost. The research paper also shows the security analysis through Forward Secrecy, Backward Secrecy, Mutual Authentication, Location Secrecy and Key Confirmation. The result analysis done through different approaches:

Pairing Time, Hash Time, Scalar Multiplication Time, Number of Nodes, Encryption Time, Decryption Time, MAC Time.

In future the proposed work should be extending to reduce the communication overhead and storage overhead. The extension done through enhancing security level in constraint based mobile wireless devises adhering the life time of battery.

References:

[1] J. Deng, R. Han, and S. Mishra, “INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks”, Technical Report CU-CS-939-02, Department of Computer Science, University of Colorado at Boulder, November 2002.
[2] B. Karp and H. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom’00), pp. 243-254, Boston, Massachusetts, USA, August 2000.
[3] P. Papadimitratos and Z. Haas, “Secure Routing for Mobile Ad hoc Networks”, In Proceedings of the SCS Communication Networks and Distributed System Modeling and Simulation Conference (CNDS 2002), pp. 27-31, San Antonio, TX, USA, January 2002.
[4] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, “Routing on Trust and Isolating Compromised Sensors in Location-Aware Sensor Networks”, In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (ACM SenSys’03), pp. 324-325, Los Angeles, USA, November 2003.
[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next Century Challenges: Scalable Coordination in Sensor Networks”, In Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom’99), pp. 263-270, Seattle, Washington, USA, August 1999.
[6] L. Hu and D. Evans, “Secure Aggregation for Wireless Networks”, In Proceedings of the International Symposium on Applications and the Internet (SAINT’03) Workshops, IEEE Computer Society, Orlando, Florida, USA, January 2003.
[7] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: A Tiny Aggregation Service for Ad-hoc Sensor Networks”, ACM SIGOPS Operating Systems Review (Special Issue), pp. 131-146, 2002.
[8] B. Przydatek, D. Song, and A. Perrig, “SIA: Secure Information Aggregation in Sensor Networks”, In Proceedings of the 1st International Conference on Embedded Networked Systems (SenSys’03), pp. 255-265, New York: ACM Press, 2003.
[9] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, “Medians and Beyond: New Aggregation Techniques for Sensor Networks”, In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (ACM SenSys’04), pp. 239-249, Baltimore, Maryland, USA, 2004.
[10] F. Ye, L. Luo, and S. Lu, “Statistical En-Route Filtering of Injected False Data in Sensor Networks”, In Proceedings of the 23rd IEEE Joint Annual Conference of Computer and Communication Societies (IEEE INFOCOM’04), Vol. 4, pp. 2446-2457, Hong Kong, China, March 2004.
[11] A. Beresford and F. Stajano, “Location Privacy in Pervasive Computing”, IEEE Pervasive Computing, Vol. 2, No. 1, pp. 46–55, 2003.
[12] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz, “Secure multicast groups on ad hoc networks”, In Proceedings of the 1st ACM workshop on Security of Ad hoc and Sensor Networks (SASN ’03), pp. 94–102, ACM Press, 2003.
[13] S. Rafaeli and D. Hutchison, “A survey of key management for secure group communication”, ACM Computer Survey, Vol. 35, No. 3, pp. 309–329, 2003.
[14] S. Burman, “Cryptography and security - future challenges and issues”, Invited Talk, in proc. of ADCOM, 2007.
[15] S. Price and K. Kosaka, “A secure key management scheme for sensor networks”, Proceedings of the Tenth Americas Conference on Information Systems, New York, August 2004.
[16] C. Mustafa and G. Jolly, “A low-energy key management protocol for wireless sensor networks”, Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC'03), pp. 1530-1546, 2003.
[17] Z. Martina and B. Erik-Oliver, “An efficient key establishment scheme for secure aggregating sensor networks”, ASIACCS'06, pp. 233-241, ACM 1-59593-272-0/06/0003, March 2006.
[18] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, Communications of the ACM, Vol. 26, No. 1, pp. 96-99, 1983.
[19] V. Miller, “Use of elliptic curves in cryptography”, In Lecture Notes in Computer Sciences: 218 on Advances in Cryptology- CRYPTO 85, Springer, pp. 417-426, New York, 1986.

[20] N. Kobiltz, "Elliptic curve cryptosystems", Mathematics of Computation, Vol. 48, pp. 203-209, 1987.
[21] A. Esam, A. Hagras, H. Aly, and D. El-Saied, "An Efficient Key Management Scheme based on Elliptic Curve Signcryption for Heterogeneous Wireless Sensor Networks", International Journal of Computer Science and Technology, ISSN:2229-4333(Print) | ISSN:0976 -8491(Online), December, 2010.
[22] J. Lee and C. Chang, "Secure communications for cluster-based ad-hoc networks using node identities", Journal of Network and Computer Applications, Vol. 30, No. 4, pp. 1377-1396, 2007.
[23] V. Varadharajan, R. Shankaran, and M. Hitchens, "Security for cluster based ad hoc networks", Computer Communications, Vol. 27, No. 5, pp. 488-501, 2004.
[24] B. Tian, S. Han, J. Hu, and T. Dillon, "A mutual-healing key distribution scheme in wireless sensor networks", Journal of Network and Computer Applications, Vol. 34, Issue 1, pp. 80-88, 2011.

ANNALS of Faculty Engineering Hunedoara – International Journal of Engineering
ISSN 1584 - 2665 (printed version); ISSN 2601 - 2332 (online); ISSN-L 1584 - 2665