Authentication Protocols for WSN using ECC and Hidden Generator

Ayaz Hassan Moon
NIELIT J & K
SIDCO Electronics Complex
Rangreth ,Srinagar, J & k, 191132

Ummer iqbal
NIELIT J & K
SIDCO Electronics Complex
Rangreth ,Srinagar, J & K,191132

ABSTRACT
Authentication is an important Security primitive in any class of network both at the entity level as well as at the message level. All the network entities comprising of WSN including nodes, cluster heads and base station need to be authenticated before sending or receiving any kind of communication within them. Public Key Cryptography offers broad based solutions to address all the security concerns. However such solutions are far too expensive to be applied directly to WSN owing to its resource constraints. ECC and its variant Tiny- ECC offers the scope and the potential to build light weight solutions for WSN based networks.

In this paper, light weight authentication protocols for Base to Node, Node to Base and Node to Node have been presented. These protocols are based on Elliptical Curve Cryptography and Hidden Generator Concepts. Hash Chains which are computationally light have been also used. The protocols have been developed in TinyOS, the defacto operating system for WSN and simulated in Tossim .The protocols have been ported to WSN hardware targeting MicaZ mote. The paper also brings out performance parameters of the developed protocols.

Keywords
Authentication, Hidden Generator ,WSN, ECC

1. INTRODUCTION
WSN is a kind of adhoc network which is infrastructure-less, the presence of base-station notwithstanding. These networks are characterized by severe resource constraints in terms of energy, computational power, bandwidth, storage. The typical characteristics of a MicaZ [1] mote are 8-Bit micro-controller, 4 KB of RAM, 128 KB of ROM, bandwidth of 250kbps is powered by two AA lithium cells with 2000mAH energy. As per a conservative estimate, this much energy may be just enough to last for 4 days of continuous operation of a mote. Primarily deployed for monitoring environmental parameters is now finding increased use in commercial, domestic, military and health applications. There deployment in hostile terrains and sometimes for mission critical applications touching human lives call for addressing their security issues. The range of applications, envisioned to be developed especially under IoT and smart city projects, would be tightly coupled to the physical world and human beings where security aspects would be paramount[2]. Researchers are therefore, hardening the security services to be embedded into WSN based applications by employing techniques which are suited to the resource constraint nature of the WSN.

Traditional cryptographic algorithms employing Public crypto are highly resource intensive to directly fit into the WSN architecture [3]. For example, it takes 14 seconds for an exponential operation of 1024-bit RSA on Mica1 motes. Though efforts have been made to implement RSA based solutions in WSN, but researchers are still apprehensive about the computational and storage overheads involved in such implementations. Among all the security primitives, authentication which may also cover data integrity, data freshness and sequencing is the most important requirement. Barring certain cases involving military and reconnaissance, confidentiality may not be the prime requirement as compared to authentication.

2. AUTHENTICATION REQUIREMENT
Authentication is an assurance about the identities of communicating nodes or principals in any network. It involves a process to ascertain that the data has come from the alleged source and has not been modified en route. An adversary may spoof the source address and then inject malicious packets into the network so as to camouflage its originality. Authentication works at two levels: one at the level of entity called as entity authentication or Identity and other at the level of data called as message authentication also known as data integrity. Major difference between an entity authentication and Message Authentication lies in the fact that entity authentication is realized when both the claimant and the verifier exchange communication in real time without revealing any meaningful message other than the claim of being a particular entity. While as message authentication in itself does not provide any timeliness guarantee in terms of as to when the message was created.

The identity authentication could be broadly based on 3 parameters .i.e. What you know e.g. password based, what you possess e.g. smart cards and what you are e.g. biometrics. In WSN, the authentication is mostly based upon what you know. The secret knowledge will be related to keys or any cryptographic material used by identities to prove their uniqueness. Thus generation of keys and their distribution is the most important step and the bedrock of all security services.

To make WSN truly ubiquitous, there are several security challenges to be surmounted. Wireless communication being broadcast in nature is more prone to different kind of attacks like eves dropping, intercept, inject and alter transmitted data. In conventional networks, authentication, data integrity and confidentiality is achieved through end-to-end mechanism like SSH, SSL, IP-Sec. In end-to-end communication it is neither necessary nor desirable for the content of the message, beyond headers, to be made available to intermediate routers. The case is different for WSN, as the dominant traffic pattern is many-to-one implemented over multi-hop topology using station.

A Typical wireless sensor network as shown in Figure 1 has following important resources:

- Base station
- Cluster head
- Sensor node

A sensor node is responsible for sensing some physical phenomena and transmitting these readings to the cluster head or directly to the base station depending upon the topology being used. In either case, it becomes imperative to authenticate the data from the sensor nodes as based upon these values, certain actuations or alarms may be triggered. Thus it becomes quite essential to provide authentication in the following cases assuming a flat topology:

- Node to Node Communication
- Node to Base Communication
- Base to Node Communication

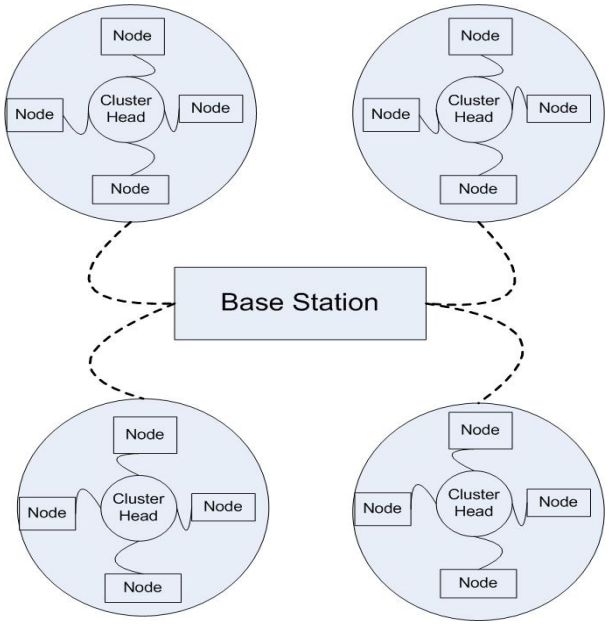

Fig 1: Typical Wireless Sensor Network

Key establishment plays a pivotal role in ensuring authentication. An uncomplicated yet efficient method to share secret keys in a WSN is based on ECDH [4]. However ECDH suffers from Man in the Middle Attack [5]. The Man in the Middle Attack can be overcome by using Hidden Generator Point concept as illustrated in this paper.

3. CONTRIBUTION OF THIS PAPER

In this paper, light weight authentication protocols covering node-node, node-base and base-node have been developed using Hidden generator concept. The method leverages the computationally fast and low overheads associated with hash chains and Elliptical Curve Cryptography. The protocols have been developed on TinyOS the defacto operating system of WSN and simulated using TOSSIM. The protocols have been ported successfully to MICAZ mote. Considering the proximity of base station to the sensor node, the communication model is assumed to be single hop, flat topology. The performance analysis of the developed protocols has also been carried out.

Rest of the paper is organized as: Section 4 provides a insight about the related work, Section 5 presents the Authentication protocols based on Hidden Generator ,Section 6 brings out the Implementation and protocol analysis and Section 7 concludes the paper.

4. RELATED WORK

Area of securing wireless sensor networks is relatively new. Security protocols like SPINS [6], TinySec[7] and LEAP[8] have been built based on different assumptions. Perrig et al introduced "Security Protocols for sensor Networks"(SPINS). SPINS comprises of sensor network encryption protocol (SNEP) and µTESLA. SNEP provides confidentiality, two party data authentication, integrity and freshness. Through a process of randomization of Initialization vectors and use of counters, SNEP achieves semantic security, which means the same plain text is encrypted differently each time the counter value is incremented. All cryptographic primitives i.e., encryption, message authentication code (MAC), hash, random number generator are constructed out of a single block cipher for code reuse.

TESLA is a broadcast authentication protocol. It authenticates the initial packet with a digital signature, which is expensive for sensor nodes. µTESLA provides authenticated data broadcasts for severely resource-constrained environment like that of WSN. To authenticate the broadcast messages, µTESLA uses delayed key disclosure and one-way hash function to generate key chain . The base station selects a random value K_n as the last key in the key chain and repeatedly performs a pseudorandom function F to compute all the other keys:

$$K_i = F(K_{i+1}) \quad , \; 0 \le i \le n-1 \, ,$$

Where the secret key K_i (except $K_{0)}$ is assigned to the i_{th} time interval. With the help of the initial key $K_{0,}$ which is called the chain commitment , receiver can authenticate any key in the chain by performing pseudorandom hash function operations.

Zhu et al proposed Localized Encryption and Authentication Protocol (LEAP) a key management protocol which supports the establishment of four types of keys for each sensor node. It includes an individual key shared with base station, a pair wise key shared with another node, a cluster key shared with multiple neighboring nodes and a group key which is shared by all the nodes. LEAP provides efficient protocol mechanism for inter-node traffic authentication. LEAP also provides schemes for sensor nodes to establish and update individual keys, pair wise shared keys, cluster keys and group keys, revocation and subsequent rekeying mechanism.

The KDC[9] based authentication mechanism employs a centralized approach with base station acting as a trusted third party. Two sensor nodes establishing a secure communication link will have a symmetric key stored in memory. The key can be used for a node to authenticate itself to the base station, and then the base station generates a session key and relays it through single hop or multi-hop transmission to the node. The disadvantage of this method is that the base station can be the single point of failure and the scheme lacks scalability.

Certificate-based public key authentication system has been used widely in the wired network, such as the PKI(Public Key Infrastructure) system, in which for authentication both sides must hold a certificate issued by the third party called CA(Certification Authority) . The two sensor nodes need to have the same configuration at the same time. The authentication scheme of TinyPK[10] based on RSA can be used conveniently to realize the WSN entity authentication.

The constraint of WSN present serious inhibition to implementation of TinyPK.ECC has shown more promise for application of asymmetric techniques for authentication in WSN. ECC [11]can achieve same level of security as RSA with smaller key size e.g. 160 Bit ECC can provide comparable security to the conventional 1024 Bit RSA[5]. Smaller key size often brings the advantage of faster computation efficiency and saving of bandwidth, memory and energy. That makes ECC better suited for resource constraint devices like WSN[12].

5. AUTHENTICATION PROTOCOLS BASED ON HIDDEN GENERATOR

This Section presents the protocols based on Hidden Generator for Node to Node, Node to Base and Base to Node authentication. The notations used in the protocols are tabulated in Table 1

Table 1: Symbol Table

S.No	Symbol	Description
1	G_a	Generator Point of $Node_i$
2	G_b	Generator Point of $Node_j$
3	k_I	Hash Seed at $Node_i$
4	k_j	Hash Seed at $Node_j$
5	z	Large Integer
6	$h^z(k_i)$	Public Hash Commitment of $Node_I$
7	$h^z(k_j)$	Public Hash Commitment of $Node_J$
8	G_s	Hidden Generator point
9	R	Secret chosen by the prover in Node to Node authentication using split Shares
10	x	Private Key of the Node
11	$Pn = (x.G_s)$	Public Key of the Node
12	s	Private Key of the Base
6	$Pb = (s.G_s)$	Public Key of the Base
7	IDA	Node ID
8	F_p	Prime field defined for ECC

Generator point as one of the parameters of Elliptical Curve need not be to be made public to avoid man-in-the-middle attack [16]. In that case, the two communicating nodes i.e., $Node_i$ and $Node_j$ shall establish a common generator point on Elliptical curve without revealing their respective Generator points to each other. The hidden generator algorithm for generating a common generator point using hash chains for the authentication is shown in the Table 2

Table 2: Hidden Generator Algorithm

Step	$Node_i$	$Node_j$
	Hash Commitment and Generator Exchanges	
1	Compute $G_a.(h^{z-1}(k_i)$ and send it to $Node_j$	
2		Compute $G_b . (h^{z-1}(k_j)$ and send it to $Node_i$
3	Send $h^{z-1}(k_i)$ to $Node_j$	
4		Send $h^{z-1}(k_j)$ to $Node_i$
	Authentication of Received Messages	
5	$Node_i$ verifies: $h(h^{z-1}(kj))= h^z(k_j)$, if true, then it computes : $G_b . h^{z-1}(kj). [h^{z-1}(kj)]^{-1} = G_b$	
6		$Node_j$ verifies: $h(h^{z-1}(k_i))= hz(k_i)$, if true, then it computes: $G_a.h^{z-1}(k_i).[h^{z-1}(k_i)]^{-1} = G_a$
	Hidden Generator Point G_s established at $Node_i$ and $Node_j$	
7	$G_a+G_b=G_s$	$G_a+G_b=G_s$

5.1 Node to Node Authentication Using Split Shares

The concept of hidden generator has been extended to develop a Node to Node authentication protocol. The sequence of steps is shown in Table 3.The Prover node uses a secret R and divides it into two parts for achieving authentication. The usage of Private key of Prover and Shared Generator Gs makes the algorithm robust against attacks like impersonation[17], man-in-the-middle etc.

Table 3: Node to Node Authentication Using Split Shares

Step No	Operation
Step 1	Prover node Computes its Public key $Pn= X.G_s$ using the Private Key X and shared generator point G_s
Step 2	Prover node generates a Secret R

Step 3	Prover node Computes : $E=(X+R).G_s$ and sends E to verifier node.
Step 4	Verifier node Computes: $E-X.G_s= R.G_s$
Step 5	Verifier node sends $R.G_s$ back to the Prover node as an acknowledgement
Step 6	Prover node Splits R=(r1,r2)
Step 7	Prover node generates $F=r1.G_s+X.G_s$ and sends it to the Verifier node
Step 8	Verifier node generates $A=(F-X.G_s)=r1.G_s$
Step9	Prover node generates $H=r2.G_s+X.G_s$ and sends it to the Verifier node
Step 10	Verifier node generates $B= (H-X.G_s)=r2.G_s$
Step 11	Verifier computes C=A+B
Step 12	Verifier node Checks if $C==R.G_s$ then Prover node is authenticated by Verifier Node

5.2 Base to Node Authentication

In this authentication algorithm, a node authenticates Base. This algorithm utilizes the Hidden Generator point along with public key of Base available with node to authenticate the Base. The sequence of steps is shown in Table 4.

Table 4: Base to Node Authentication

Step No	Operation
Step 1	Node Selects a Private Key X.
Step 2	Node Computes $Pn= X*G_s$
Step 3	Base selects a Private Key :S
Step 4	Base Computes $Pb=S*G_s$
Step 5	Node Computes IDA*Pn and sends it to the Base
Step 6	Base Computes A= S* Ida*Pn and Sends it to the Node
Step 7	Node Computes B=IDA*X*Pb
Step 8	Node Checks if (A==B) the Base is authenticated.

5.3 Node to Base Authentication

In this authentication algorithm, Base authenticates node. This algorithm utilizes the Hidden Generator point along with public key of node available with Base to authenticate the node. The Sequence of steps is shown in Table 5.

Table 5 Node to Base Authentication

Step No	Operation
Step 1	Node Selects a Private Key X.
Step 2	Node Computes $Pn=X*G_s$
Step 3	Base selects a Private Key :S
Step 4	Base Computes $Pb=S*G_s$
Step 5	Node Computes H(IDA)
Step 6	Node Computes $[H(IDA)+X] * G_s$ and Sends $[H(IDA)+X] * G_s$, IDA to the Base.
Step 7	Base Stores $A=[H(IDA)+ X] * G_s$ and Computes $B=[H(IDa)*G_s +Pn]$
Step 8	Base Checks if (A==B) the Node is authenticated

6. IMPLEMENTATION AND ANALYSIS

The protocols were implemented in TinyOS[13] operating system using NesC Language and simulated on TOSSIM[14]. TinyOS is a open-source lightweight operating system specifically designed for low-power wireless sensors. NesC is a dialect with features to reduce RAM and code size to enable significant optimizations, and help prevent low-level bugs like race conditions. The NesC programs developed were enabled with a highly optimized ECC implementation, TinyECC[15]. TinyECC is a code packet provided by North Carolina State University. It provides a base arithmetic operation of ECC on TinyOS. It provides all ECC operations on domain , including the point add, double and scalar multiplication on F_p.The component graph of TinyECC is shown in the figure 2.TinyECC provides implementation of various curves like Secp 160 ,Secp 128,Secp 190.

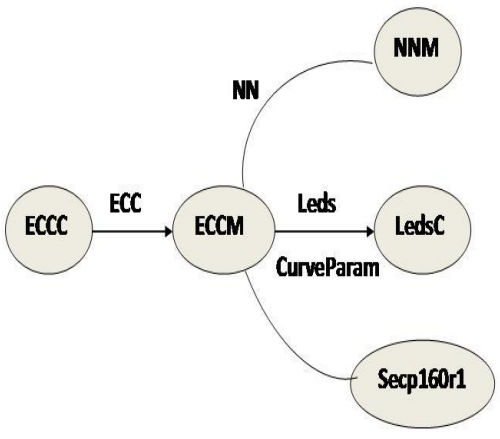

Fig 2: Component graph of TinyECC

The NesC Code of the 3 protocols developed was deployed on MicaZ Motes. A MIB 520[1] Programmer was used to program the Motes. While Fusing the Motes with actual code, RAM and ROM Consumption as calculated by TinyOS were captured and is shown in the figure 3 and figure 4.For the purpose of capturing computational time of various key ECC operations like Point addition, Scalar Multiplication, a nesC program was developed for sending the time message to a TinyOS Serial Forwarder through MIB520[1]. The time in the time message comprised of the time taken in the execution of the specific operation. This was achieved by starting and stopping the timer interface and recording the difference of the two. The difference was sent as a payload to the serial forwarder.

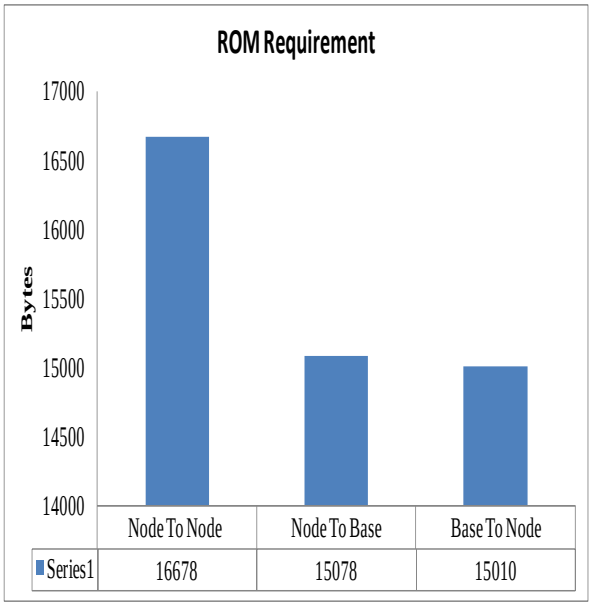

Fig 3: ROM Requirements

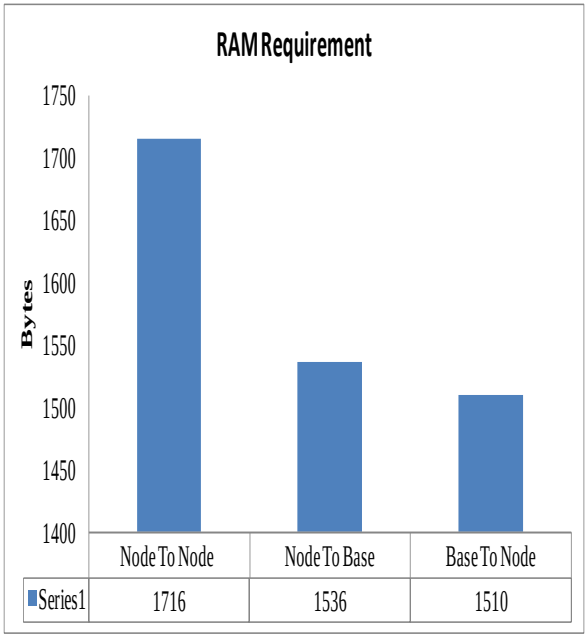

Fig 4: RAM Requirements

The time taken by various critical operations and no of critical operations in the developed protocols are tabulated in Table 6 and Table 7.

Table 6: Computational time of Key ECC operations

S.NO	Operation	Time Taken (Secs)
1	Scalar Multiplication	1.78
2	Point Addition	1.7
3	Multiplicative Inverse	0.11

Table 7: No of Critical ECC operations in Developed Protocols

Protocol	No of Scalar Multiplication	No of Point Addition	No of Multiplicative Inverse
Node to Node	4	5	Nil
Base to Node	5	Nil	Nil
Node to Base	4	Nil	Nil

The execution time of the developed protocols is shown in the figure 5.

Energy Calculations would primarily depend on computational time taken for core ECC operations like Point Addition, Scalar Multiplication in addition to the voltage and current requirements. For calculating energy, E= V*i*t (joules) is used, where V and i stand for voltage and current drawn respectively, t is the execution time for each operation. MicaZ node using Atmel AT Mega 128 L is powered by 02 AA batteries. With a voltage of 3 V form 02 AA batteries, and a maximum load current of 19.7 mA, the energy consumption is shown in figure 6.

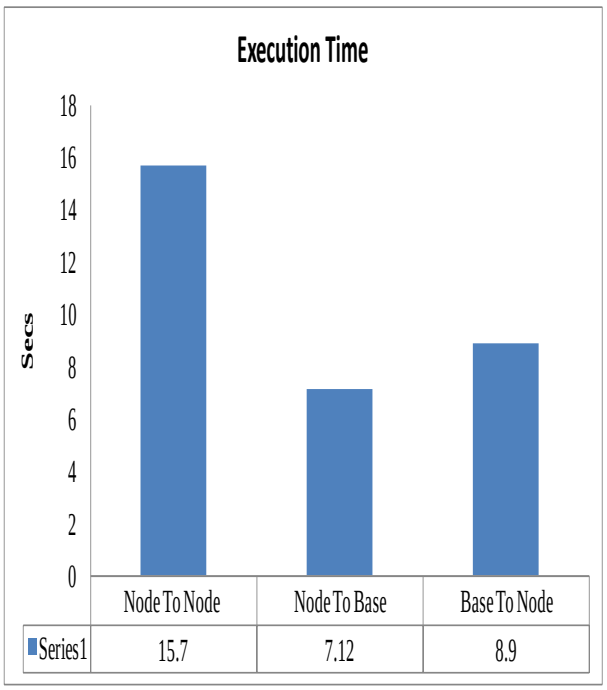

Fig 5: Execution Time of the Developed Protocols

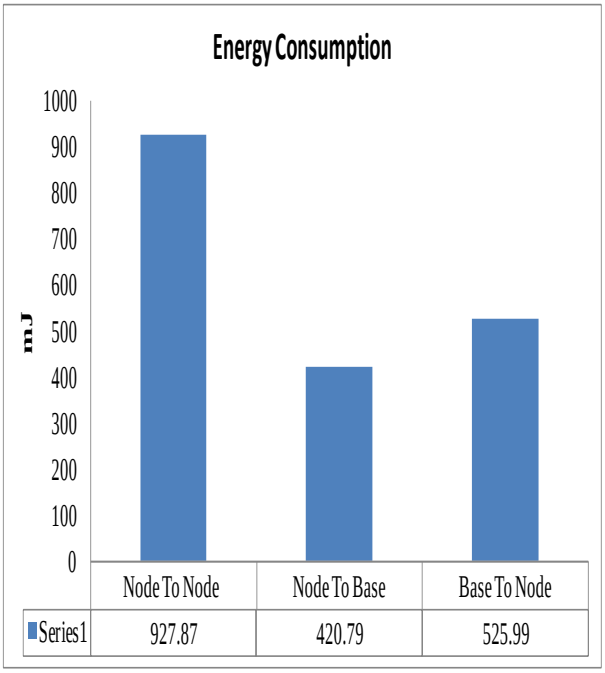

Fig 6: Energy Consumption of the Developed Protocols

7. CONCLUSION

In this paper authentication protocols based on ECC have been proposed and implemented in TinyOS platform. Existing authentication mechanism in WSN were discussed .These techniques mainly suffer from the issues like node capture,scalablity,communication overhead, Man-in-the Middle Attack. By leveraging the concept of hidden generator, threats like man in the middle attack are mitigated. Hash Chains which are computationally light have been used to generate common hidden generator point between a pair of nodes. The same concept can be extended to derive pair wise keys between two nodes.

The paper also brings out the memory and energy analysis of authentication schemes discussed. The RAM and ROM consumption of the 3 protocols developed does not significantly vary as the required Library components for all the 3 protocols remain the same. The ROM consumption of the 3 protocols lie between 14 to 16 KB and the RAM requirement lies between 1.2 KB to 1.6 KB. As the number of ECC operations differ significantly in the developed protocols therefore, the execution time and energy consumption varies considerably. Node to Node authentication protocol exhibits the highest execution time and energy consumption among the developed protocols where as that of Node to Base is lowest.

The work presented in this paper can be further extended for developing a comprehensive entity and message authentication framework for WSN. The energy efficient operations of Tiny ECC and computationally light one way Hash functions can be leveraged to develop authentication schemes for low power devices intended for smart city applications. Such schemes would also help in achieving broadcast authentication without using public crypto schemes like expensive RSA based digital signatures or even ECDSA.

8. REFERENCES

[1] Mote Works, Getting Started Guide, March 2013 ,PN: 7430-0102-02

[2] Francisco Sanchez-Rosario et al. 2015 A Low Consumption Real Time Environmental Monitoring System for Smart Cities based on ZigBee Wireless Sensor Network IEEE,978-1-4799-5344-8, 2015.

[3] Adrian Perrig, John Stankovic, David Wagner June 2004 Security in wireless sensor networks Communications of the ACM, vol 47,no. 6, pp 53-57,.

[4] Crtien etal 2009 Energy-Efficient Implementation of ECDH Key Exchange for Wireless Sensor Networks IFIP International Federation for Information Processing

[5] Bernard Menzes "Network Security and Cryptography", Cengage Learning

[6] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victor Wen and David E.Culler 2001 SPINS: Security protocol for sensor networks in proceedings of 7th International conference on mobile networking and computing, 2001, vol 8, no.5, pp 189-199,.

[7] Karlof, C., Sastry, N., Wagner, D. 2004 TinySec: a link layer security architecture for wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys, Baltimore, MD, USA, November 3-5, 2004, pp. 162–175. ACM

[8] S.,Setia,S., and Jajochia, S 2003 LEAP: Energy efficient security mechanism for large-scale distributed sensor networks In the proceedings of the conference on computer and communications security ,03,ACM Press, Washigton DC, pp 62-72.76.

[9] Ping Guo et al 2013 Authentication Mechanism on Wireless Sensor Networks: A Survey ITCS

[10] R. Watro, D. Kong, S.Cuti, C.Gardiner, C.Lynn and P. Kruus 2004 TinyPK: Securing sensor networks with public key technology. in the proceedings of 2nd ACM workshop on security of adhoc sensor networks (SASN 04), pp 59-64, New York, ACM press.

[11] D. Hankerson et al. 2004 Guide to Elliptic Curve Cryptography" Springer

[12] www.certicom.com

[13] TinyOS. http:// www.tinyos.net

[14] P. Levis, N. Lee, M. Welsh and D. E. Culler. et al 2003 TOSSIM : Accurate and Stable Simulation of Entire TinyOS Applications. SenSys

[15] A. Liu and P. Ning et al. 2008 Tiny ECC: A Configurable Library for Elliptical Curve Cryptography in Wireless Sensor Networks IPSN

[16] Moon A.H and Iqbal Ummer .2015 Light Weight Secure Key Generation Protocol with Hidden Generator Point using ECC in Transactions on Networks and Communications

[17] Moon A.H, Shah NA, Iqbal Ummer, Ayub Adil 2013 Simulating and Analyzing Security Attacks in WSN Using Qualnet, published in IEEE Conference on ICMIRA,pp. 68-76,2013