Securing Wireless Sensor Networks Using Elliptic Curve Cryptography

Najmus Saqib, Sayar Singh Shekhawat

M.Tech, Computer Science And Engineering,
RTU, Kota, Rajasthan, India.

Abstract
Wireless Sensor Networks have been an active area of research owing to its myriad range of applications. Traditional security protocols are not feasible for such networks due to their resource constraint nature. However, ECC has been considered as a viable cryptographic technique due to its low computational overhead.in this paper study and application of ECC on a popular wsn operating system, TinyOS has been carried out. Practical implementation of the ECC operations have been performed using TinyECC library. TinyECC has been used to develop custom security protocols on TinyOS. Performance benchmarking of the proposed protocols has also been carried out. The developed protocol has been fused on the Micaz Motes

Keywords— WSN; TinySec; ECC; TinyOS, Hash Chains

I. Introduction

Wireless sensor systems involve low power sensor nodes. These nodes have limited assets in terms of processing, power and Memory.Wirleess Sensor networks are playing critical role in the development of smart city applications. As these applications are data centric, there is a dire need to provide security primitives like authentication, integrity, confidentiality. However due to their resource contained nature conventional security protocols cannot be directly employed.

Public key cryptography along with symmetric key cryptography has been prudent in providing security primitive for traditional networks. However, this methodology has not been leveraged in case of WSN due to its resource constraint nature. Traditional PKC is not feeble as they involve heavy computational operations. However, ECC based variant of PKC has emerged as a viable option for providing PKC platform.

N. Gura et al and W. Du at al demonstrates that the ECC can be effectively implemented in the resource constraints nodes. The advantage of using ECC instead of traditional RSA is in the fact that 160 bits key in ECC provides equivalent security to that of 1024 bits of RSA as shown in Table 1.

Table 1: Key comparison between RSA and ECC in terms of security equivalence

Key length of RSA	Key length OF ECC	Ratio of RSA/ECC
512	106	5:1
768	132	6:1
1024	160	7:1
2048	210	10:1

This paper provides the study an application of ECC in WSN on a popular WSN operating system, TinyOS. Rest of the paper is organized as: Section 2 provides the context knowledge pertaining to tinyos platform. Section 3 discusses an ECC library, TinyECC, which can be used for practical applications. Section 4 designs and implements a security protocol based ECC and TinyOS.

II. TinyOS Platform

A. *TinyOS*

TinyOS is an embedded operating system designed for wireless sensor networks. The operating system has been developed by university of California, Berkeley. The architecture of TinyOS is a component based. thus, providing high degree of usability. The execution methodologies of tinyOS employees split phase operations. The split phase operations are similar to Asynchronous method calls. The operating system provides a single shared stack thus there is no kernel /User space differentiation. The design of the TinyOS emphasis on having Low duty cycle.

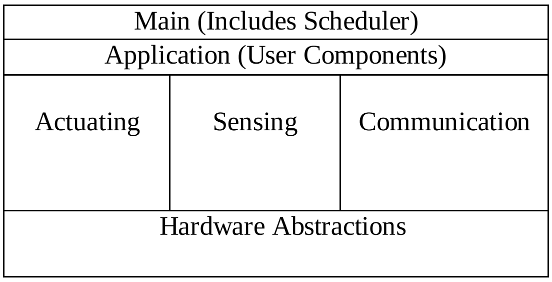


Figure1: Architecture of Tiny OS

B. *NesC Language*

NesC language is programming language design to embody the structuring concepts and execution model of TinyOS. The Building blocks of NesC language:

Interface – provides an abstract definition of the interaction between two components. The Interfaces are implemented in split phase manner.

Application – it is graph of one or more components wired along with the TinyOS scheduler together to form an executable

Module – it provides the implementation of one or more interfaces. if the module is using an interface it must implement the events generated by it.

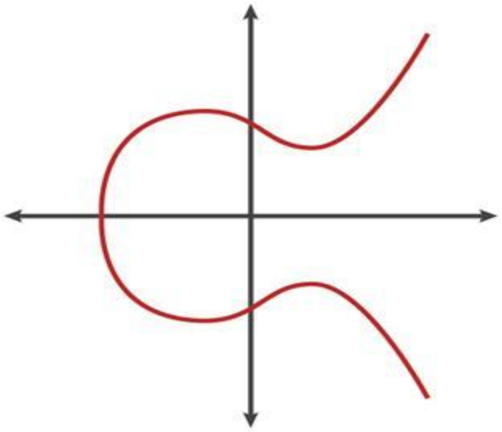
Configuration – it is component that wires other components together to form an application.

C. Tossim

TOSSIM[16] is a discrete event simulator for TinyOS sensor networks. Instead of compiling a TinyOS application for a mote, users can compile it into the TOSSIM framework, which runs on a PC. This allows users to debug, test, and analyse algorithms in a controlled and repeatable environment. As TOSSIM runs on a PC, users can examine their TinyOS code using debuggers and other development tools. TinyViz is a Java visualization and actuation environment for TOSSIM.

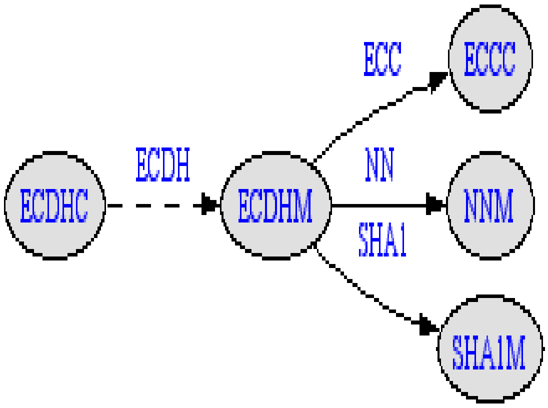
III.TinyECC

A curve E over a finite field FP is an Elliptical Curve if it is of the form: y2 = x3 + Ax +B, where A & B ∈FP. The discriminant (∆) = 4A3 + 27B2 ≠ 0, which ensures no degenerate case occurs. The Elliptical Curve E may also be defined as a set such that, E = {(x, y) | y2 = x3 + Ax +B} ∪ {O}, where 'O' is the point at infinity. The Elliptical Curve E has the following domain parameters (FP, a, b, G, n, h) where 'FP' is the finite field over which the elliptical curve is defined, 'a' & 'b' are elements of the elliptical curve equation. 'G' is the generator point that can generate all other points of the same elliptical curve, 'n' is the Order of the Generator point 'G', and 'h' is the co-factor. The strength of the Elliptical Curve cryptosystem is directly proportional to the difficulty of the Elliptical Curve Discrete Log Problem (ECDLP) which is explained in the next section. The more difficult the ECDLP, the better the security for the Cryptosystem.



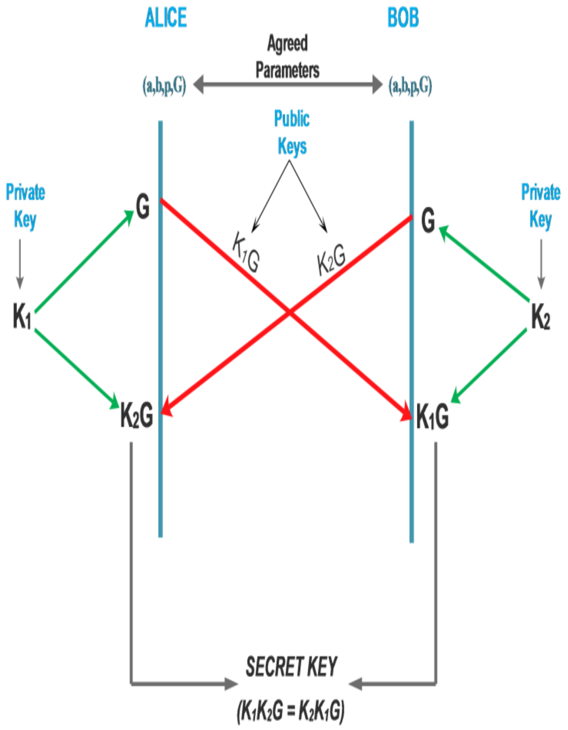
Elliptical Curve of the form y2=x3-x+1

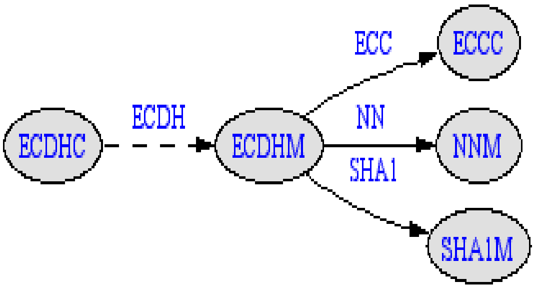
TinyECC [13] provides simple, configurable, flexible and ready-to-use software for developing WSN based applications with ECC at its core. All the ECC operations including point addition, point doubling and point multiplications are supported by TinyECC. The major modules of TinyECC include ECDSA, ECIES and ECDH.



A. ECDH

ECDH is a key agreement/exchange scheme, that enables two or more entities to communicate through an insecure channel and generate a secret key. This key helps in providing the necessary data confidentiality and security. In this scheme the participants agree upon a value called a key, from which many more keys can be derived to be later used in a symmetric encryption or data authentication scheme. The participants also agree to certain domain parameters such as (a, b, p, G) {where terms have their usual meanings }. Each of the participant contributes to the final key and cannot know it on its own.





B. *Elliptical Curve Digital Signature Algorithm(ECDSA)*

ECDSA is an asymmetric authentication scheme depending on a pair of private and public key. The private key is used by the authenticator to digitally sign the message, while the public key is used by the recipient of the message to verify the authenticator. In case the message is modified on its way, the signature changes and the authentication process fail. ECDSA is an analogous form of the Digital Signature Algorithm (DSA) wherein the Algorithm is used on Elliptical Curves.

ECDSA is an asymmetric authentication scheme depending on a pair of private and public key. The private key is used by the authenticator to digitally sign the message, while the public key is used by the recipient of the message to verify the authenticator. In case the message is modified on its way, the signature changes and the authentication process fail. ECDSA is an analogous form of the Digital Signature Algorithm (DSA) wherein the Algorithm is used on Elliptical Curves.

ECDSA incorporates the following Steps:

STEP-1: Key Pair Generation

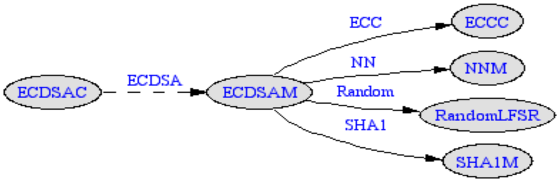
The Private key known to the authenticator is used to generate the private key. In addition to this the Elliptical Curve Domain Parameters (FP, a, b, G, n, h) are also used. A random or a pseudo-random integer ‘d’ in the interval [1, n-1] is chosen which becomes the authenticator’s private key. The public key is computed as: Q = dG, where Q is the public key and G is the Generator point.

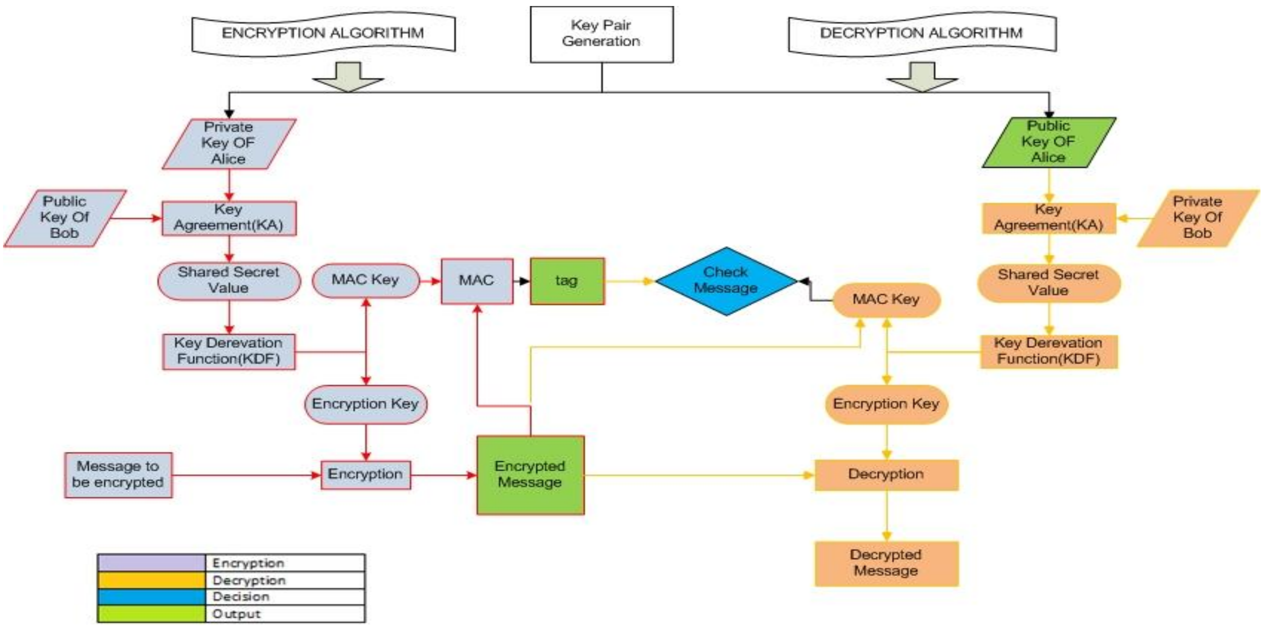
Step-2: Signature Generation

The recipient of the message uses a digital signature to verify that the received message is from the authenticator. For generating a signature, an authenticator having a key pair (d, Q), first chooses a random or a pseudo-random number (k) in the range of [1, n-1]. The signature consists of two integers ‘r’ & ‘s’. For computing r, the following steps are employed:
(x1, y1) = kG mod p where p is a prime no. ∈FP
r = x1 mod n, r must be in the range [1, n-1]
If this r comes out to be equal to 0, a new r is computed again. To compute s, a secure hash algorithm such as SHA-1 converted in the form of an integer ‘e’ is derived. ‘s’ is computed as:
s = k-1(e + dr) mod n
If this ‘s’ comes out to be ‘0’, a new random number ‘k’ is selected and both ‘r’ & ‘s’ are re-computed.
(r, s) is the signature of the authenticator.

Step 3: Signature Verification

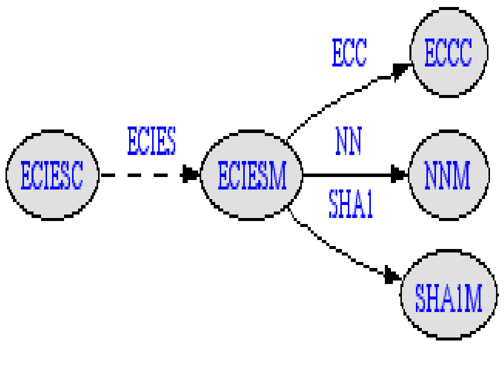
The received message needs to be authenticated, and for this purpose the receiver uses the authenticator’s domain parameters and public key ‘Q’. The authenticator’s signature (r, s) is also known. The receiver uses the same secure hash algorithm (SHA-1 in this case) to get ‘e’. For the verification process the following values are computed:
w = s-1 mod n
u1 = ew mod n
u2 = rw mod n
(x2, y2) = u1G + u2Q mod n
If (x2, y2) comes out to be equal to the ‘point at infinity’ (O), the signature is rejected & if x2 comes out to be equal to ‘r’, the verification is successful.





C. Elliptical Curve Integrated Encryption Algorithm(ECIES):

Integrated Encryption Scheme (IES) is a hybrid encryption scheme which provides semantic security against an adversary who is allowed to use chosen-plaintext and chosen-ciphertext attacks. The security of the scheme is based on the Diffie–Hellman problem. The elliptic curve integrated encryption system (ECIES) is the standard elliptic curve based on encryption algorithm. It is called integrated, since it is a hybrid scheme that uses a public key system to transport a session key for use by a symmetric cipher. ECIES is a public-key encryption algorithm.



IV. Designing Protocols using TinyECC

TinyECC can be used to design and implement new security protocols .in this section a key exchange protocol is designed. The new protocol is better than the ECDH in terms of authentication. ECDH suffers from Man in the Middle Attack.

A. Proposed Authentication Protocol.

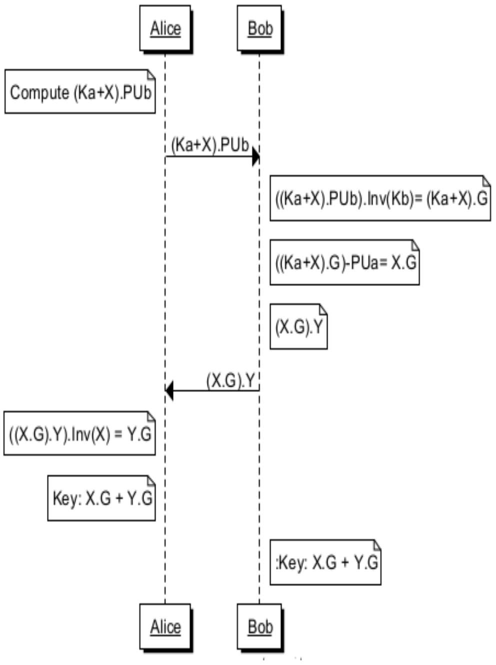
The major problem in ECDH is that no authentication happens prior to exchange. Based on this need, an improved protocol has been suggested. in the Figure 3. The algorithm developed performs a sequence of steps to establish a shared key between two parties while in co-operating a mechanism to authenticate.

In this protocol, Alice and BOB choose the Radom numbers x and y respectively. The secret chosen by Alice is X.G and by BOB is Y.G. The purpose of the key exchange algorithm is that Alice and BOB should be able exchange their secrets with each other in an authenticated manner. BOB receives (Ka+x). Pub. BOB multiplies (Ka+x).Pub with inv(Kb) i.e. inverse of the private key. A nesC program was written to measure the time taken by various critical operations. Time taken by critical operations in the proposed protocols are tabulated in figure 3.

Protocol	**Point Addition**	**Point Multiplication**	**Inverse Operation**
ECDH	NIL	4	NIL
Proposed Protocol	3	4	2

Fig.3.ROM Requirement of ECDH and Proposed Protocol

A sequence of steps to establish a shared key between two parties while in co-operating a mechanism to authenticate.



V. Conclusion

In this paper, Re-keying of TinySec using hash chains and ECC has been addressed. It was mentioned that ECDH has been suggested a technique for Re-keying. However, ECDH for pair wise key establishment is expensive. Moreover, ECDH does not offer authentication primitive. A protocol using hash chain has been proposed for re-keying the entire the network. This protocol with minimal computational overhead can update the existing network wide key of TinySec.A protocol for establishing pair wise authenticated key has also been proposed. The protocol establishes a pair wise key in an authenticated manner. The protocols were in TinyOS and Simulated in TinyOS. Performance benchmarking of the developed against ECDH has also been carried out. From performance benchmarking matrix, it is evident that with little computational overhead the protocols are able to Re-key the network.

References

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

[2] 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.

[3] Karlof, C., Sastry. Wagner,D. 2004 . TinySec: Link Layer Security Architecture for Wireless Sensor Networks," Sensys., Baltimore, MD.

[4] 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.

[5] N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz. 2004.Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. CHES2004.

[6] Malan, D.J., Welsh. Smith,D.M,2008, Implementing Public Key Infrastructure for Sensor Networks," Transactions on Sensor Networks, vol.4

[7] Malan, D.J., Welsh. Smith, 2004 A Public Key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography. First Annual IEEE Communications Society Conference on Sensor and Adhoc Communications and Networks, pp. 71-80.

[8] D.Hankerson et al. 2004. Guide to Elliptic Curve Cryptography. Springer;

[9] Bernard Menzes. Network Security and Cryptography. Cengage Learning

[10] Lampard L.1981. password. Authentication with in Secure Communication. COMM ACM ;1981; 24:770-772

[11] Levis, P., and Gay.D., 2009 TinyOS Programming. Cambridge University Press.

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

[13] A. Liu and P. Ning et al.2008. Tiny ECC: A Configurable Library for Elliptical Curve Cryptography in Wireless Sensor Networks," in proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track

[14] Moon A.H and Iqbal Ummer .2015 .Light Weight Secure Key Generation protocol with Hidden Generator point using ECC"Transactions in Network and Communication, Society for Science and Education, United Kingdom