Elliptic Curve based Digital Envelope in WSN

Ravi Kishore Kodali
Department of Electronics and Communications Engineering
National Institute of Technology, Warangal
Andhra Pradesh, 506004, India

***Abstract*—Data privacy and integrity are critical factors in certain Wireless sensor network (WSN) application areas demanding security. Many cryptographic algorithms have been proposed to meet the security requirements of these. The digital envelope is one of the practices used to attain privacy, authentication, integrity, maintenance, and non-repudiation in e-commerce channels. However, the WSN nodes have various constrained resources, like limited battery energy, less computational power and small memory. Elliptic curve cryptography (ECC), one of the Public key cryptographic (PKC) security techniques, which can be implemented on the nodes in WSN. This work deals with an implementation of ECC based digital envelope based security model in WSN applications.**

I. Introduction

In general, the deployed physical area of a wireless sensor networks(WSN) is very large and certain WSN application fields, which include humidity and temperature, vehicular traffic monitoring and control, human body monitoring, etc. Authentication and access control are some of the challenges in a mobile network with dynamic topology having limited resources. In military and certain commercial WSN application areas, the transiting data need to be kept confidential and only authorized users to be permitted to gain access to the same. Many studies are being conducted involving security aspects in WSN's.

A digital envelope is a secure electronic data container that is used to protect a message through encryption and data authentication. A digital envelope allows users to encrypt the data with the speed of a symmetric key encryption scheme and the convenience and security of a public key encryption. A digital envelope can provide privacy and tamper resistant, whereas as a digital signature is similar to sealing an envelope. Due to this, the receiver shall have greater confidence that the message originated from the intended original sender and further, the receiver can ascertain that the message is received without any modification during transit, maintaining integrity of the data contained therein. Figure 1 shows generation of a digital signature. Figure 2 illustrates how an encrypted message is generated. A random number key generator providing encrypted random key is given in Figure 3. Both digital envelopes and digital signatures can be used in the context of WSN's for the improvement of their security.

As WSN nodes have various resource constraints in the form of reduced battery energy and limited computational resources, a novel implementation of the digital envelope needs to be implemented in WSN applications. Any signed

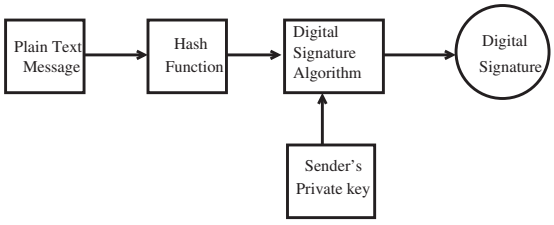

Fig. 1. Generation of digital signature

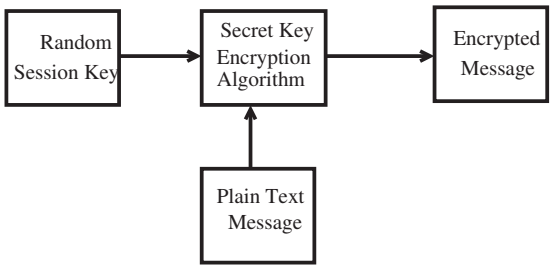

Fig. 2. Generation of Encrypted message

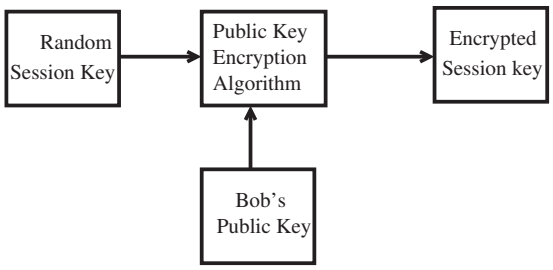

Fig. 3. Generated of Encrypted Session Key

digital envelope based application mainly comprises of the following two steps: 1. Generation of digital signature and 2. Generation of digital envelope. Both of these can be implemented using different algorithms. Table I gives a list of different attack types at each of the layers. The rest of the paper is organized as follows: Section II gives an implementation of a digital signature algorithm using ECC, section III provides the implementation of digital envelope algorithms and section IV presents various issues related to the key distribution and seed values in WSN.

II. IRIS node Description

IRIS mote designed and manufactured by MEMSIC, is used in this work. IRIS mote has Atmel's AT1281 micro-controller

978-1-4799-1441-8/13/$31.00 ©2013 IEEE

TABLE I
DIFFERENT THREATS IN WSNS

Layer	Type of attack
Physical	Jamming interference Physical Tampering
Data Link	Collisions Unfairness Exhaustion
Network	Spoofing and altering Selective Forwarding Sink hole Attacks The Sybil Attack Wormholes Hello world attacks Acknowledgement Spoofing
Transport	Flooding De-synchronization

with Atmel's AT86RF230 transceiver based on IEEE 802.15.4 protocol and Zigbee compliant radio [1]. Using the transceiver, a mote can be tuned to any of the frequency channels in the range, 2.405 GHz - 2.480 GHz, each separated by 5 MHz with 16 channels. Communication range of the mote can be set by varying transmission power in the range 3 dBm to -17.2 dBm. IRIS mote provides 4 KB RAM and 128 KB flash memory, which is sufficient for most of the WSN applications. Apart from flash memory on micro-controller, IRIS mote also contains an external 512 KB flash memory to support *On The Air Programming* (OTAP). Table II provides the specifications of an IRIS mote device.

TABLE II
IRIS MOTE SPECIFICATIONS

Parameter	Value
RAM	8 KB
ROM	128 KB
Clock Frequency	7.37 MHz
Supply voltage	2.7 to 3 V
TX Power	-17 to 3 dBm
TX current consumption	16 mA
RX current consumption	15 mA

III. DIGITAL SIGNATURE ALGORITHMS

A signed digital envelope involves generation of its digital signature. For the messages that are transmitted using a less secure channel, a proper implementation of a digital signature scheme infuses confidence to the recipient in trusting the message that it originated from the claimed sender. The following are some of the commonly used digital signature algorithms:

1) RSA-based signature schemes, such as RSA-PSS [2]
2) DSA and its elliptic curve variant ECDSA [3]
3) ElGamal signature scheme as the predecessor to DSA, and variants of Schnorr signature and Pointcheval Stern signature algorithm [4]
4) Rabin signature algorithm [5]
5) Pairing-based schemes, such as BLS
6) Undeniable signatures

All of these algorithms require more power as they are computationally complex, which make them infeasible for WSN applications in view of the resource constraints of the nodes. In the digital signature stage, the hash value of the message is encrypted along with the private key of the sender. This stage involves two different operations: Hash value computation of the message and its digital signature. In order to implement these operations, ECC can be employed.

A. Hash functions using Elliptic Curve Cryptography

Algorithm 1 HASH function using ECC [6]

1. Select a natural number N and a set of coefficients a,b∈F, where F represents finite field. All the elements in this finite field are represented by N+1 bits
2. Generate an elliptic curve y^2 = x^3 + a*x + b
3. Twist of the elliptic curve, namely, TEC is generated which satisfies α*y^2 = x^3 + a*x + b, here α is a non quadratic residue of the finite field, F. Define z_i = α*y_i^2 for the following steps of the algorithm.
4. Select a point on ECC (x,y)∈EC and another point $(x_t,\sqrt{z_t}) \in$ TEC
5. The set of coefficients and the two base points are given as public information
6. A pair of scalar random numbers k1 and k2 are selected as private information
7. Compute the scalar multiplications (x_{pu},y_{pu}) = k1*(x,y) and $(x_{tpu},\sqrt{z_{tpu}})$ = k2*$(x_t,\sqrt{z_t})$, these two points are established as public information
8. The input message is pre-processed into a message of N-bits
9. Assign a initial Hash point (x_{c1},y_{c1}) on the elliptic curve and another hash point $(x_{ct1},\sqrt{z_{ct1}})$ on the twist of the elliptic curve.
10. Assign an integer i, which represents each message data block and assign i = 2
11. Embed the ith block of N- bits onto the x-coordinate of the EC message point $(x_{mi}$,$\sqrt{z_{mi}}$)
12. Now compute the set of hash points (x_{ci},y_{ci})=$((x_{mi},y_{mi})$+$(x_{ci-1},y_{ci-1}))$ and $(x_{cti},\sqrt{z_{cti}})$=$((x_{cti-1},\sqrt{z_{cti-1}}))$ if αm*i =1 otherwise the equations are reversed
13. The steps 11 and 12 are iterated until all blocks of data get completed.
14. Concatenating bits of the x-coordinate xc, and the sign bit of the y-coordinate yc of a hash point, along with the bits of the x-coordinates xct and the sign bit of the y-coordinate yct of a hash point to form a hash bit string.

The hash values are irreversible. The hash functions, using ECC, employ elliptic curves and the security strength of these hash functions is dependent on the difficulty level of the elliptic curve discrete logarithm problem (ECDLP) in terms of the computational complexity, thereby ensuring that the hash functions are optimally secure.

The algorithm uses two curves, namely, elliptic curve and the twist of the same elliptic curve (TEC) within the same encryption scheme [6]. Here, it may be noted that this method remains valid, even if the elliptic curve and its twist are not isomorphic to each other. Algorithm 1 shows the steps for generating the hash functions using these elliptic curves. This algorithm is implemented using IRIS WSN nodes. The WSN implementation of the hash algorithm using a prime field of less than 255 has been tried with a message size of 4 bytes. The message is divided into 4 blocks, each block of one byte size. All the four iterated functions are run to compute the hash value as per the algorithm. The number of iterations depends on the number of blocks in the input message. The difficulty level is dependent on the number of blocks of the input message. Table III presents the results of

TABLE III
ANALYSIS FOR HASH ALGORITHM COMPUTATION USING ECC

Parameter	Computational overhead	Communication + Computational overheads
ROM	5.858 KB	58.66 KB
RAM	1.16 KB	3.86 KB
Clock cycles needed	320	320
Time needed	1.25 S	1.25 S

the experimental analysis on the IRIS node.

B. Generation of Digital Signatures using ECC

To obtain the digital signature, we need to sign the hash value of the message and the same can be achieved by using elliptic curves (EC). Each string in the hash value is mapped onto a point on the elliptic curve and then scalar point multiplication operations are carried out. This EC point is mapped onto a string using the same EC parameters. The Elliptic Curve Digital Signature Algorithm (ECDSA) is one of the algorithms used for signing the generated hash value. However, in ECDSA, SHA-1 algorithm is used to compute the hash values. In the proposed scheme, the hashing is obtained by making use of EC's. By using this approach, hash value computational complexity is comparable with those of SHA or any other hashing algorithm providing a similar security level. The signed hash value is called the digital signature. The proposed digital signature algorithm is implemented using IRIS WSN nodes. Table IV provides experimental analysis of this implementation.

TABLE IV
ANALYSIS FOR DIGITAL SIGNATURE ALGORITHM USING ECC

Parameter	Value
RAM	3.86 KB
Clock cycles	540
Time required	2.1 S

IV. DIGITAL ENVELOPE ALGORITHMS

The development of a digital envelope requires a random key generator and public key encryptor. The message is encrypted using a random key and the key is also encrypted.

A. Random key generation using EC

EC's can be used while generating random numbers. In the work [7], a random number generator based on the addition of the points on an EC over a finite field has been developed. This generator is used along with EC point generator. The block diagram of the proposed generator is shown in Figure 4.

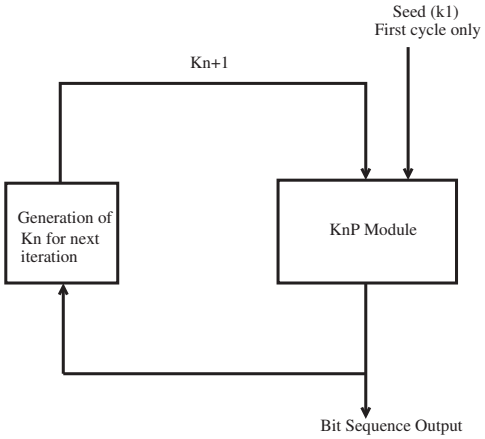

Fig. 4. Random Number Generator

In this process, scalar multiplication of a point, P, with an integer on an EC is same as that of adding the point, P, with itself (k-1) times. Here, this scalar multiplication is referred as kP operation. The integer, r, for which rP = O, where 'O' is the point of infinity of the EC and r, is the order of P. We can convert k into binary form and using Koblitz's method [8] and by employing the addition and doubling operations of EC, the kP operation can be computed with less number of computations.
When k_n is fed into the module, the k_nP point is generated. In order to select the seed for the next $(n+1)^{th}$ iteration, the seed value is chosen from the x- co-ordinate, x_n of k_nP, and current value n by using equation (1).

$$k_{n+1} = x_n + n(modp) \tag{1}$$

Using this random number generator, different bit sequence is generated by taking another initial seed value, k1, and an alternate initial point, P. Both these values need to be kept secret as the next random values generated depend on these values. Generally, the points which are used on the EC are treated as the parameters of public keys and hence, P cannot be a point that is used in the ECC. The length of the random values simply depends on the initial seed values.

B. Public key encryption using ECC

Public key encryption is also termed as asymmetric encryption as two different keys are chosen, one public key of the receiver during encryption of data and the other private key of the receiver during decryption of data. The most important advantage of asymmetric encryption is that it allows

a secured communication between two parties without the actual exchange of their respective private keys, only the corresponding public keys are published. Figure 5 shows Public key encryption block diagram [9].

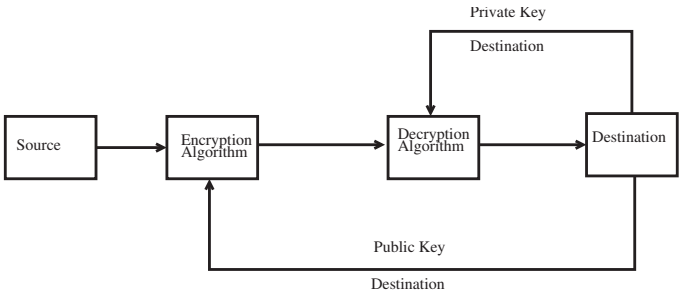

Fig. 5. Block diagram of public encryption algorithm

The scenario of implementing public key cryptographic algorithms is still dynamic since in terms of public key encryption algorithms, many use the RSA algorithm. Nowadays the number of applications using EC based algorithms has been increasing rapidly. Table V shows the comparison of key sizes for different crypto-algorithms.

TABLE V
COMPARISON OF KEY SIZES FOR DIFFERENT SECURITY ALGORITHMS

Symmetric	ECC	DH/DSA/RSA
80	160	1024
128	283	3072
192	409	7680
256	571	15360

TABLE VI
REQUIREMENTS FOR DIFFERENT CRYPTOGRAPHIC PRIMITIVES

Cryptographic primitive	sender requirements	Receiver requirements	Is Key exchange necessary?
Hash functions	ECC variables	ECC variables	Yes
Signature Algorithms	secret key	No	No
Random number generator	seed values and ECC variables	not necessary	No
Asymmetric Cryptography	Public keys	Private key	No

In view of various resource constraints of WSN nodes, ECC can be employed as it demands for lesser key sizes. The procedure of EC based encryption and decryption is given in Algorithm 2.

As discussed previously, all the cryptographic primitives that are needed for a digital envelope, can be implemented by using EC's alone. Historically , EC's are computationally efficient and also have satisfactory security standards. Table VI gives a brief summary of all the cryptographic primitives that are generated by using EC's

Algorithm 2 Encryption and Decryption using ECC

1. Represent the message as a point M in the prime field of ECC
2. Select a scalar k in the prime field and calculate the scalar multiplication value C = k*P
3. Represent the public key as Q and now compute D = M + k*Q
4. The pair (C,D) is the cipher text pair
5. In decryption procedure, let pr be the private key of the receiver and compute M = D - pr*C
6. Decode the message point to extract the original message, m

V. DISTRIBUTION OF PUBLIC KEYS AND THE SEED VALUES IN WSN

The major problem is the distribution of public keys and various seed values for all the nodes in the WSN. One approach can be dividing the WSN into a number of clusters or groups of nodes and a cluster head monitors the key exchange with other cluster heads and also in its own cluster. The problem with this method is that the cluster head need to expend more amount of its energy and hence it dies sooner. This can be circumvented with random assignment of the cluster head role to a node. Another method can be on-demand exchange of keys. A node, which wants to send its data, communicates with the destination node, the key exchange takes place and then the data transfer is initiated. Table VI shows the requirements for different cryptographic primitives to be implemented. The computational and energy requirements are dependent on the number of nodes as well as the network security requirements.

Table VII gives various commonly used key management approaches that can be employed in WSN's [10].

TABLE VII
KEY MANAGEMENT PROCEDURES IN WSN NETWORKS

Model	Description	Benefits	Problems
Network	A secret key is shared by the entire network	Simplicity Data Aggregatabilty Scalability self-organizability Flexibility	No robustness
Pairwise	Unique key is shared between a specific pair of nodes	High robust scheme Node authentication is possible	Non-scalability No self-organizabilty Less flexible while scaling
Group	Unique key for each group of nodes	Multicasting Collaboration High robustness Scalability	Storage problem in IEEE 802.15.4 Secure set-up is not possible Application dependent formation of clusters

VI. CONCLUSION

The proposed signed digital envelope model in WSN requires cryptographic primitives with less computational and

memory requirements. All the cryptographic primitives can be implemented by using the same elliptic curve parameters. Since the range of WSN is generally small, EC's can be used for various security aspects. This can be a good choice for the implementation as all the algorithms needed in both the digital envelope and digital signature can be implemented by using EC's demanding fewer of the scarce resources of a WSN node. The work also included experimental implementation analysis on a resource constrained node.

REFERENCES

[1] Z. Alliance, "Zigbee specification," *ZigBee Document 053474r13*, pp. 344–346, 2006.

[2] J. Malone-Lee and W. Mao, "Two birds one stone: signcryption using rsa," in *Topics in CryptologyCT-RSA 2003*. Springer, 2003, pp. 211–226.

[3] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[4] D. R. L. Brown, "Elgamal signature schemes," Feb. 26 2010, uS Patent App. 12/714,116.

[5] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold dss signatures," in *Advances in CryptologyEUROCRYPT96*. Springer, 1996, pp. 354–371.

[6] L. Ghouti, M. K. Ibrahim, and A. J. Al-Najjar, "Hash functions using elliptic curve cryptography," May 22 2012, uS Patent 8,184,803.

[7] L.-p. Lee and K.-w. Wong, "Elliptic curve random number generation," in *Electrical and Electronic Technology, 2001. TENCON. Proceedings of IEEE Region 10 International Conference on*, vol. 1. IEEE, 2001, pp. 239–241.

[8] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," in *Towards a Quarter-Century of Public Key Cryptography*. Springer, 2000, pp. 103–123.

[9] G. S. Quirino, A. R. Ribeiro, and E. D. Moreno, "Asymmetric encryption in wireless sensor networks," pp. 217–232, InTech, 2012.

[10] J. C. Lee, V. C. Leung, K. H. Wong, J. Cao, and H. C. Chan, "Key management issues in wireless sensor networks: current proposals and future developments," *Wireless Communications, IEEE*, vol. 14, no. 5, pp. 76–84, 2007.