Volume 2, Issue 9, September 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Routing-Driven elliptic Curve cryptography Based Key Management Scheme for Heterogeneous Sensor Networks

K. Naga Divya
Dept. of CSE., P.V.P.S.I.T,
India

D. Sree Lakshmi
Dept. of CSE., P.V.P.S.I.T,
India

K.Sri Vijaya
Dept. of IT., P.V.P.S.I.T,
India

Abstract- sensor networks are deployed in a hostile environment, security becomes extremely important. Abstract- sensor networks are deployed in a hostile environment, security becomes extremely important. An efficient Key Management Scheme to provide security in HSN. In HSN, Clusters are formed as shown in below figure. Routing is done in two phases: 1) Intra-cluster routing each L-sensor sends data to its cluster head(H-Sensor) via multi hops of other L-sensors ; 2)Inter-cluster routing –a cluster head aggregates data from multiple L-sensors and then sends the data to the sink via the H-sensor backbone. This Project focuses on intra cluster routing using MST (minimum spanning tree) algorithm to approximate the least energy consumption case. After constructing SPT(Spanning tree), every L-sensor node sends sensor information to H-sensor(Cluster head) with in a cluster. In this presents a preventive technique to overcome non-differential side channel attack in HSN by enhancing Elliptic Curve Cryptography and it minimizes storage space requirement, communication overhead and energy consumption in HSN.

***Keywords:** Sensor Networks, cryptography, key management, clustering.*

I. INTRODUCTION

Wide-spread deployment of sensor networks is on the horizon. Networks of thousands of sensors may present an economical solution to some of our challenging problems: real-time traffic monitoring, building safety monitoring (structural, fire, and physical security monitoring), military sensing and tracking, distributed measurement of seismic activity, real-time pollution monitoring, wildlife monitoring, wildfire tracking, etc. Many applications are dependent on the secure operation of a sensor network and have serious consequences if the network is compromised or disrupted. Research has shown that Heterogeneous Sensor Network (HSN) model has better performance and security. In this model sensor nodes have different capabilities in terms of communication, computation, energy supply, storage space, reliability and other aspects.

Security is critical to sensor networks deployed in hostile environments, such as military battlefield and security monitoring. Key management is an essential cryptographic primitive upon which other security primitives are built. Due to resource constraints, achieving such key agreement in wireless sensor networks is non-trivial. Eschenauer and Gligor first presented a key management scheme for sensor networks based on probabilistic key predistribution. Several other key pre-distribution schemes (e.g., [2]) have been proposed. Those schemes require a large storage space for key pre distribution and are not suitable for small sensor nodes.Thus Xiaojiang Du, Mohsen Guizani, Yang Xiao and Hsiao-Hwa Chen have proposed a routing driven ECC based key management scheme [1] based on the fact that most sensor nodes only communicate with a small portion of their neighbours. In this paper,the above scheme has been improved by including efficient ECC to overcome non-differetial side channel attack.

II. BACKGROUND WORK

2.1. WIRELESS SENSOR NETWORKS:

A W**ireless Sensor Network** (WSN) consists of spatially distributed autonomous *sensors* to cooperatively monitor physical or environmental conditions, such as *temperature*, *sound*, *vibration*, *pressure*, *motion* or *pollutants*. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance.

A typical sensor network has hundreds to several thousand sensor nodes. Each sensor node is typically low-cost limited in computation and information storage capacity, highly power constrained, and communicates over a short range wireless network interface. Most sensor networks have a base station that acts as a gateway to associated infrastructure such as data processing computers. Individual sensor nodes communicate locally with neighboring sensors, and send their sensor readings over the peer-to-peer sensor network to the base station. Generally, sensor nodes communicate over a wireless network. A typical sensor network forms around one or more *base stations*, which connect the sensor network to the outside network.

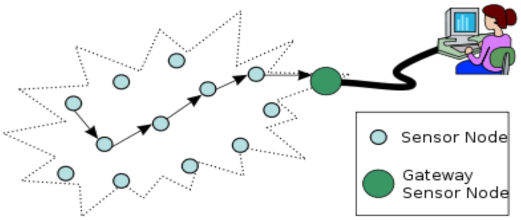


Fig.2.1 Wireless Sensor Network Architecture

The communication patterns within a sensor network fall into three categories: node to node communication (e.g., aggregation of sensor readings), node to base station communication (e.g., sensor readings), base station to node communication (e.g., specific requests).

2.1.1. Heterogeneous Wireless sensor networks

Heterogeneous Sensor Networks (HSNs), where sensor nodes have different capabilities in terms of communication, computation, energy supply, storage space, reliability and other aspects. Wireless Sensor Network systems (WSNs) are increasingly following heterogeneous designs, incorporating a mixture of elements with widely varying capabilities. The development and deployment of WSNs rides heavily on the availability of simulation, emulation, visualization and analysis support. The presence of heterogeneous nodes in a sensor network is known to increase network reliability and lifetime.

Let us consider an HSN consisting of two types of sensors: a small number of high-end sensors (H-sensors) and a large number of low-end sensors (L-sensors). Both H-sensors and L-sensors are powered by batteries and have limited energy supply. Clusters are formed in an HSN. For an HSN, it is natural to let powerful
H-sensors serve as cluster heads and form clusters around them.

The Cluster Formation

After sensor deployment, clusters are formed in an HSN. An illustration of the cluster formation is shown in Fig. 1, where the small squares are L-Sensors, large rectangular nodes are H-sensors and the large square at the bottom-left corner is the sink which acts as base station to connect the sensor network to outside network. In cluster formation of HSN, H-sensor serve as the cluster head in each cluster and all H-sensors form a backbone connecting to sink. From each cluster each L-sensor sends sensor information to its cluster head (H-sensor) ,then that Sensor information is forwarded to other cluster head in order to reach the base station.

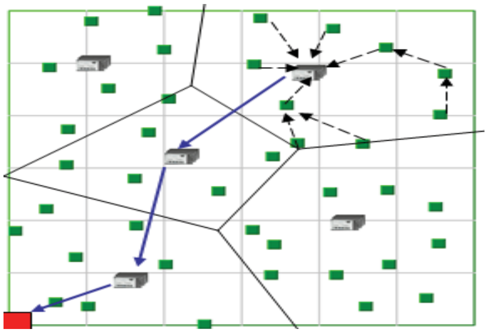


Fig2.2.Cluster Formation

To achieve security in wireless sensor networks, it is important to be able to encrypt messages sent among sensor nodes. Keys must be established in each sensor node and they must be agreed upon by communicating nodes for encryption purpose. Key establishment in sensor networks is a challenging problem because asymmetric key cryptosystems are unsuitable for use in resource constrained sensor nodes, and also because the nodes could be physically compromised by an adversary. Many applications such as real-time traffic monitoring and military sensing and tracking are dependent on the secure operation of a sensor network, and have serious consequences if the network is compromised.

2.1.2. Importance of key management Scheme in Heterogeneous Sensor Networks

To provide security, communication should be encrypted and authenticated. An open research problem is how to bootstrap secure communications among sensor nodes, i.e., how to set up secret keys among communicating nodes.

This key agreement problem is a part of the *key management* problem. There are three types of general key agreement schemes: trusted-server scheme, self-enforcing scheme, and key pre-distribution scheme. The *trusted-server* scheme depends on a trusted server for key agreement between nodes, e.g., Kerberos [4]. This type of scheme is not suitable for sensor networks because there is usually no trusted `infrastructure in sensor networks. The *self-enforcing* scheme depends on asymmetric cryptography, such as key agreement using public key certificates. However, limited computation and energy resources of sensor nodes often make it undesirable to use public key algorithms, such as Diffie-Hellman key agreement or RSA. The third type of key agreement scheme is key *pre-distribution*, where key information is distributed among all sensor nodes prior to deployment. If we know which nodes are more likely to be in the same neighborhood before deployment, keys can be decided *a priori*. However, because of the randomness of deployment, it might be infeasible to learn the set of neighbors *a priori*.

In the research of Key management scheme for HSN, an efficient key management scheme that only needs small storage space has been presented. This scheme achieves significant storage saving by utilizing 1) the fact that most sensor nodes only communicate with a small portion of their neighbors; 2) Elliptic Curve cryptography.

Elliptic Curve cryptography(ECC) takes limited power consumption for Sensor network . One main advantage of ECC is its small size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA.

III. ECC-BASED KEY MANAGEMENT SCHEME

3.1. INTRODUCTION:

Most existing key management schemes try to establish shared keys for all pairs of neighbor sensors, no matter whether these nodes communicate with each other or not, and this causes large overhead. This scheme provides significant reduction in communication overhead , storage space and energy consumption than other key management scheme. It achieves significant storage saving by utilizing 1) the fact that most sensor nodes only communicate with a small portion of their neighbors; 2) an efficient public-key cryptography.

3.2. KEY GENERATION USING ELLIPTIC CURVE:

Choose an Elliptic Curve E over finite field .Choose a Prime field F_P ,which consists of finite number of elements between 0 to P-1. The Elliptic Curve over a finite field F_P is the set of all (x,y)(x,y $\in F_p$) that satisfies the following equation:

$$\underline{Y^2 \bmod P=X^3+aX+b \bmod P} \quad \text{where a,b,}F_p \text{ and } 4a^3+27b^2 \bmod P\neq 0.$$

If a point is on Elliptic Curve $G(X_G ,Y_G)$,then there is minimum positive integer 'n' such that nG=O, where "O" is Point of infinity and Integer 'n' is called the order of Point G. A scalar K is chosen for Point multiplication in between 0 and n-1.

The Public Key "Q" is formed by following equation:

$$Q=KG,$$

where 'K' is Discrete Logarithm of Q to the base P.

The *elliptic curve discrete logarithm problem (ECDLP) is the following:*

Given an elliptic curve E defined over F_q, a point P∈ E (Fq) of order n, and a point $Q=E(F_q)$, determine the integer l, $0 \leq l \leq n - 1$, such that Q = lP, provided that such an integer exists.

Point Multiplication:

In point multiplication a point 'G' on the elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on the same elliptic curve.

i.e. Q= kG

Point multiplication is achieved by two basic elliptic curve operations:

Point addition, adding two points J and K to obtain another point L

i.e., L = J + K.

Point doubling, adding a point J to itself to obtain another point L

i.e. L = 2J

An example of point multiplication:

Let P be a point on an elliptic curve. Let k be a scalar that is multiplied with the point P to obtain another point Q on the curve. i.e. Q = kP.

If k = 23 then

$Q= kP = 23.P = 2(2(2(2P) + P) + P) + P$

Thus point multiplication uses point addition and point doubling repeatedly to find the result.

Procedure to find Point Addition and Point Doubling:

Let P1 and P2 be two points on E , it is possible to find a closed formula that gives the coordinates (x_s,y_s) of the sum Ps of two points P1 and P2 as
a function of their coordinates (x_1, y_1) and (x_2, y_2).

$$Xs=\lambda 2-X_1-X_2$$
$$Ys=\lambda(X_1-X_s)-Y_1$$

$\lambda=(Y_2-Y_1)/(X_2-X_1)$ if $P1\neq P2$

$\lambda=(3X_1^2+a)/(2Y_1)$ if $P1=P2$

Example: Let us take an Elliptic Curve E over F_p as E: $Y^2=X^3+aX+b$
where (a,b)=(1,14) and point P on curve as (X,Y)=(-2,2). Now let us choose scalar K=3 to evaluate the public Key 'Q' using above formula.

$$Q=3P$$
$$=2P+P$$

Evaluation of Point doubling(2P):

$$\lambda= \frac{3*(-2)^2 +1}{2*2}$$
$$=13/4$$
$$X_S=10.5625$$
$$Y_S=-42.8281$$

Evaluation of addition:(2P+P):

$$\lambda=-3.5684$$
$$X_S=-12.7334$$
$$Y_S=-40.3010$$

This resultant point will also satisfy the Curve equation .

3.3. BROADCASTING ROUTING TREE STRUCTURE TO ALL L-NODES IN A CLUSTER:

Each L-sensor node(denoted as u)sends its location information to its cluster head "H". U computes Message Authentication Code(MAC) over the message by using u's Private Key, and MAC is appended to message. H-Node canverfy the MAC and then authenticate u's identify ,by using u's Public key and H generates a certificate for u's Public key by using H's Private key.

H-Node determines the routing tree structure(i.e Parent-Child relationship) in cluster and Sends to all L-nodes with the corresponding Public key certificate to each L-Sensor.The public key certificates are signed by H's private key and can be verified by every L-sensor, since each L-sensor is preloaded with H's public key. A public key certificate proves the authenticity of a public key and further proves the identity of one L-sensor to another L-sensor.

3.4. KEY EXCHANGE BY EACH SENSOR NODE WITH ITS C-NEIGHBOUR:

If two L-sensors are parent and child in the routing tree, then they are *c*-neighbours of each other, and they will set up a shared key by themselves. For each pair of *c*-neighbours, the sensor with smaller node ID initiates the key establishment process.

For example, suppose that L-sensor *u* and *v* are *c*-neighbors and *u* has asmaller ID than *v*. The process is presented below:

1) Node u sends its public key $K^U_u = I_uP$ to v.
2) Node v sends its public key $K^U_v = I_vP$ to u.
3) Node u generates the shared key by multiplying its private key I_u with
v's public key K^U_v,
i.e., $K_{u,v} =K_u^R K^U_v = I_uI_vP$; similarly, v generates the shared key –
$K_{u,v} =K_u^R K^U_v = I_uI_vP$;

After the above process, nodes u and v share a common key and they can start secure communication.

IV. ECC- ADVANCED (ECC-A)

The proposed Key Management Scheme gives better performance by reducing communication overhead and energy consumption and also provides better security against non-differential side channel attack by including unified addition formula in key generation. It also shows reduction in energy consumption of point multiplication operation , if that operation includes more number of doublings than additions. Since this scheme gives high security, it can be applied to military applications.

4.1. KEY GENERATION PHASE:

Choose an We irstrass form of an Elliptic Curve E over a Prime finite field F_p and select a random value as private key K from the Prime finite field F_p such that the value is in between 0 and n-1,where n is order of point.

The order of a point P on an elliptic Curve is the smallest positive integer 'r' such that rP=O,where O is the point at infinity. The Elliptic Curve over a finite field F_P is the set of all (x,y)(x,y $\in F_P$) that satisfies following equation:

$$\underline{Y^2 \bmod P=X^3+aX+b \bmod P}$$

The Public Key "Q" is formed by following equation: Q=KG , Where 'K' is Discrete Logarithm of Q to the base P.

Hence the Point multiplication has to be performed to get public key. The x co-ordinate of resultant point of multiplication will be taken as public key. The Point multiplication is performed using double-and-add method.This method is as shown below.

```
---------------------------------------------------
Input: P, $k= (1,k_{t-2},\ldots\ldots k_0)_2$
Output: Q=kP
---------------------------------------------------
    $R_0 \leftarrow P$
for j=t-2 down 0 do
    $R_0 \leftarrow 2R_0$
    if ($k_j$==1) then $R_0 \leftarrow R_0+P$
    end for
    return $R_0$
---------------------------------------------------
```

(a). Double-and-add method

The above method gives resultant point performing Point multiplication using doubling and addition operations. This Point doubling and Point addition is performed using below procedure.

<u>Procedure to find Point Doubling and Point Addition</u>:

An unified addition formulae is presented to perform Point doubling and addition by taking weierstrass-form of an elliptic . Let K be a field of characteristic Char K≠ 2,3 and let E be the elliptic curve given by equation E: $y^2=x^3+ax+b$. Then for any P = (x_1, y_1) and Q=(x_2, y_2) E (K)\{O} with $y_1 \neq -y_2$. we have P+Q = (x_3, y_3) then

$x_3= \lambda^2 -x_1-x_2$ $\qquad$ $y_3= \lambda (x_1-x_3)-y_1$

Where, $\lambda=(x_1 (x_1+x_2)+x_2^2+a) / (y_1+y_2)$

4.3. BROADCASTING ROUTING TREE STRUCTURE TO ALL L-NODES IN A CLUSTER:

Each L-sensor node(denoted as u)sends its location information to its cluster head "H". U computes Message Authentication Code(MAC) over the message by using u's Private Key, and MAC is appended to message. H-Node canverfy the MAC and then authenticate u's identify ,by using u's Public key and H generates a certificate for u's Public key by using H's Private key. H-Node determines the routing tree structure(i.e Parent-Child relationship) in cluster and Sends to all L-nodes with the corresponding Public key certificate to each L-Sensor.The public key certificates are signed by H's private key and can be verified by every L-sensor, since each L-sensor is preloaded with H's public key. A public key certificate proves the authenticity of a public key and further proves the identity of one L-sensor to another L-sensor.

4.4. KEY EXCHANGE BY EACH SENSOR NODE WITH ITS C-NEIGHBOURS:

If two L-sensors are parent and child in the routing tree, then they are *c*-neighbours of each other, and they will set up a shared key by themselves. For each pair of *c*-neighbours, the sensor with smaller node ID initiates the key establishment process.

For example, suppose that L-sensor *u* and *v* are *c*-neighbors and *u* has asmaller ID than *v*. The process is presented below:

1) Node u sends its public key $K^U_u = I_uP$ to v.
2) Node v sends its public key $K^U_v = I_vP$ to u.

3) Node u generates the shared key by multiplying its private key I_u with
v's public key $K^U{}_v$,
i.e., $K_{u,v} = K_u^R\, K^U{}_v = I_u I_v P$; similarly, v generates the shared key –
$K_{u,v} = K_u^R\, K^U{}_v = I_u I_v P$;

After the above process, nodes u and v share a common key and they can start secure communication.

4.5. PERFORMANCE EVALUATION AND SECURITY ANALYSIS:

This ECC-Advanced key management scheme provides better performance by reducing storage space requirement and energy consumption. It also reduces Communication overhead by establishing communication with only C-neighbours (Communication-neighbours).As it uses Elliptic Curve Cryptography (ECC),the computations are less compare to other public key algorithm like RSA. Thus ECC is more suitable to small sensor nodes.In this scheme,if one node is compromised then the probability of compromising other node is zero since each node has its own private key and public key only.

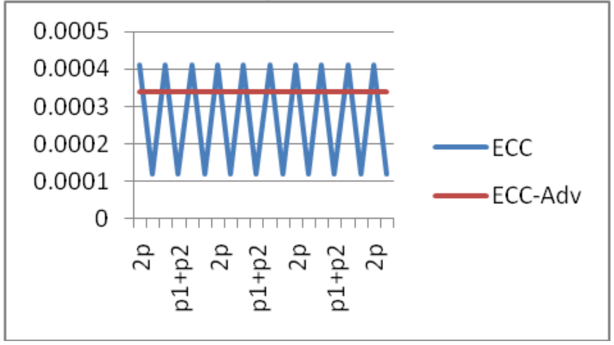
This Scheme gives high security against non differential side channel attack using unified addition formula(same to doubling and adding) by taking weirstrass form of an elliptic curve. In this attack the attacker captures secret data using simple power analysis which includes power consumption as side channel. In Point multiplication ,as we use unified addition formulae the attacker can not identify the difference between energy consumption of doubling and adding operation. Hence he can not find private key by tracing double-and-add –method.

The total energy consumption needed for the Point multiplication is less compare to the previous method, if the numbers of doubling operations are more than three times to that of number of addition operations. This results are showed in below table.

Number of Point Multiplications (point is represented as 'P')	Energy consumption to evaluate λ value for Point multiplication in ECC-based scheme	Energy consumption to evaluate λ value for Point multiplication in ECC-Advanced scheme	Calculations
2P	0.000410	0.00068	2P
5P	0.00094	0.00102	2(2P)+P
10P	0.001350	0.00136	2(2(2P)+P)
*20P	0.001760	0.00170	2(2(2(2P)+P)))
25P	0.001880	0.00204	2(2(2((2P)+P))+P
30P	0.002100	0.00238	2(2(2((2P)+P)+P)+P
35P	0.002290	0.00238	2(2(2(2(2P))+P)+P
*40P	0.002170	0.00204	2(2(2(2(2P)+P))))
*50P	0.002290	0.00204	(2(2(2(2(2P+P)))+2P)

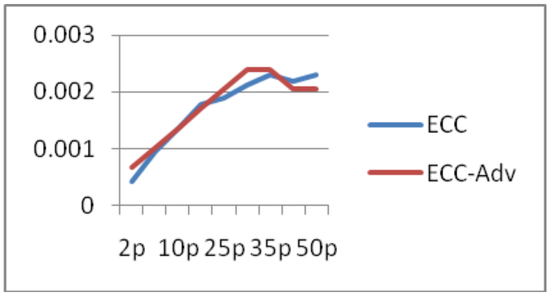
Table1.calculation of energy consumption of λ for Point multiplication

Y-axis:Energy consumption
Xaxis: Point multiplication(Double and add)



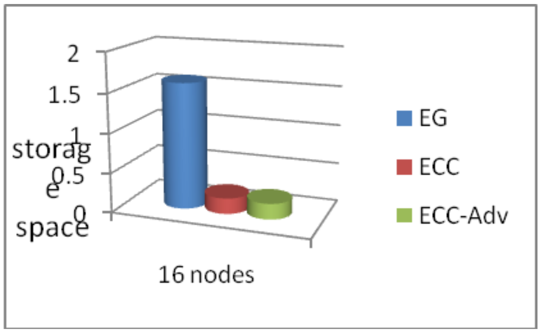
4.1. Energy consumption of λ for doubling and adding in ECC and ECC-Advanced method

Y-axis:Energy consumption
Xaxis: Point multiplication



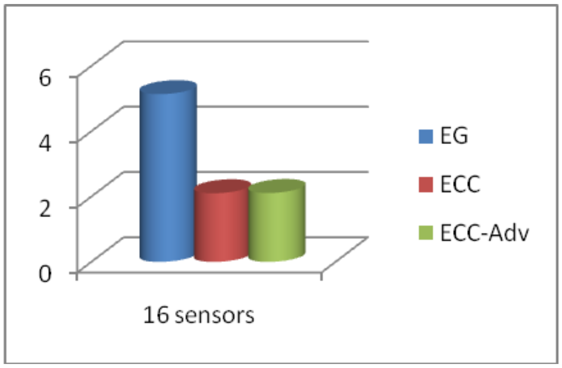
4.2. Comparison of Total Energy consumption of λ calculation methods for various Point multiplications in ECC and ECC-Adv

Y-axis: Storage space(in bytes)
X-axis: Number of nodes



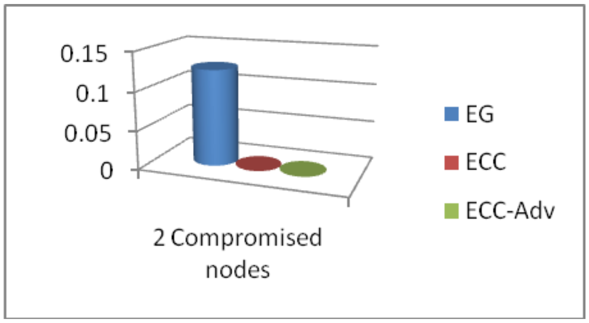
4.3.Comparison of storage space requirement for EG,ECC and ECC-Advanced schemes

Y-axis: Energy consumption
X-axis: Number of Sensors



4.4. Comparison of total energy consumptions

Y-axis: Compromising Probability
X-axis: Number of compromised nodes



4.5.The probability of an independent secure link being compromised

V. CONCLUSION AND FUTURE WORK

In this thesis, an Efficient ECC based Key management scheme against non-differential side channel attack has been presented. This scheme reduces storage space Requirement, Communication overhead and provides security using unified addition formulae for Point multiplication in Elliptic Curve Cryptography. This approach also ensures saving of energy consumption for point multiplication, if that number of doubling operations are more than three times to that of addition operations in point multiplication. If one node is compromised, then the probability of compromising other node with captured information is zero since the keys are independent to each node. This can be extended by applying to all types of elliptic curves with some modification.

REFERENCES

[1] Xiaojiang Du, Member, IEEE, Mohesen Guizani,Fellow,IEEE,Yang Xiao, Senior Member, IEEE,and Hsiao-Hwa Chen,Senior Member,IEEE "*A Routing- Driven elliptic Curve cryptography Based Key Management Scheme for Heterogeneous Sensor Networks*" IEEE Transaction on Wireless Communications, VOL.8, NO. 3 MARCH 2009, pp.1223- 1229.

[2] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes For sensor etworks," in *Proc. 2003 IEEE Symposium on Security and Privacy*, May 2003, pp. 197-213.

[3] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Proc. 6th Interna tional on Cryptographic Hardware and Embedded Systems*, Boston, MA, Aug. 2004.

[4] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society, Lecture Note Series 265, Cambridge University Press