© 2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Improved Diffie-Hellman Key Exchange Using Elliptic Curve (IDHECC) Scheme for Securing Wireless Sensor Networks Routing Data

Monika Kapoor Batra, Priyanka Bhatnagar

All Saint's College of Technology, Bhopal Madhya Pradesh, India

ABSTRACT

In Wireless Sensor Network (WSN) is a large scale network with thousands of tiny sensors moreover is of utmost importance as it is used in real time applications. Currently WSN is required for up-to-the-minute applications which include Internet of Things, Smart Card, Smart Grid, Smart Phone and Smart City. However the greatest issue in sensor network is secure communication for which key management is the primary objective. Existing key management techniques have many limitations such as prior deployment knowledge, transmission range, insecure communication and node captured by the adversary. The proposed novel ECC and diffie-hellman key exchange algorithm provides better transmission range and secure communication. The overall network is separated into circular tracks and triangular sectors. Energy conservation Routing Protocol was used for routing of data in WSN, which reduces the delay with increased packet delivery ratio. Further for secure routing Improved Diffie-Hellman key exchange using Elliptic Curve (IDHECC), which reduces the memory space and computational overhead than the existing Elliptic Curve Cryptography (ECC) key management scheme for Securing WSN (Wireless sensor Network).

Keywords : Wireless Sensor network; Routing Protocol; Key Management Scheme; Elliptic Curve Cryptography; IDHECC;

I. INTRODUCTION

Elliptic Curve Cryptography (ECC) was first proposed by victor Miller and independently by Neal Koblitz in the mid-1980s and has evolved into a mature public-key cryptosystem. Compared to its traditional counterparts, ECC offers the same level of security using much smaller keys. This result in faster computations and savings in memory, power and bandwidth those are especially important in constrained environments. More significantly, the advantage of ECC over its competitor's increases, as the security needs increase over time. Recently the National Institute of standards and Technology (NIST) approved ECC for use by the U.S.government. Several standards organizations, such as Institute of Electrical & Electronics Engineers (IEEE), American National Standards Institute (ANSI), Open Mobile Alliance (OMA) and Internet Engineering Task Force (IETF), have ongoing efforts to include ECC as a required or recommended security mechanism.Here we present our new algorithm using Diffie_hellman key exchange algorithm providing forward secrecy for web browsers Since the begin of the third Millennium, remote sensor systems (WSNs) produced an expanding enthusiasm from mechanical and examination points of view [1–7]. A WSN can be for the most part portrayed as a system of hubs that agreeably sense and may control nature empowering connection between persons or PCs and the encompassing environment [8]. On one hand, WSNs empower new applications and hence new conceivable markets; then again, the configuration is influenced by a few imperatives that call for new ideal models. Actually, the action of detecting, preparing, and correspondence under constrained measure of vitality, lights a cross-layer configuration approach normally obliging the joint thought of appropriated sign/information handling, medium access control, and correspondence conventions [9].

This is a review of WSNs innovations, primary applications and measures, highlights in WSNs plan with contextual analysis, and developments. Specifically case of execution in view of exploratory results will be accounted for. As for the writing [1] this paper bargains not just with applications and elements of WSNs, or just on configuration of WSNs, yet assembles every one of these perspectives, concentrating likewise the consideration on advances and measures.

WSNs have a few basic perspectives with remote specially appointed system [10] and much of the time they are basically considered as an extraordinary instance of them. This could be prompt wrong conclusions, particularly when conventions and calculations intended for specially appointed systems are utilized as a part of WSN.

Applications, on top of the stack, set prerequisites that drive the determination of conventions and transmission strategies; at the flip side, the remote channel postures imperatives to the correspondence capacities and execution. Taking into account the necessities set by applications and the limitations postured by the remote channel, the correspondence conventions and methods are chosen.

II. METHODS AND MATERIAL

A. Traditional Protocol In Elliptic Cryptography

An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \quad (1)$$

Where *a*, *b*, *c*, *d*, and *e*, are real numbers.

A special addition operation is defined over elliptic curves and this with the inclusion of a point *O*,called point at infinity.If three points are on a line intersecting an elliptic curve, then their sumis equal to this point at infinity *O*,which acts as the identity element for this addition operation.

Sometimes the general equation (1) can be referred as Weierstrass equation as shown in (2)

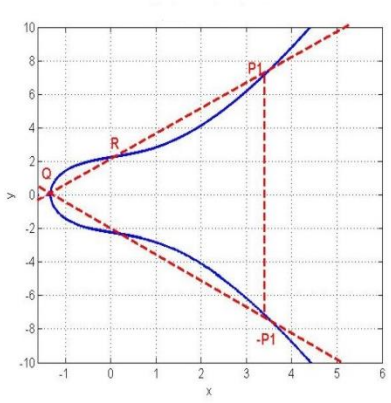


Figure 1. Elliptic curves

If we wanted use a elliptic curve to be used for cryptography the necessary condition is the curve is not singular, i.e. the discriminant of polynomial: $f(x) = x^3 + ax + b$:

$$4a^3 + 27b^2 \neq 0 \quad (3)$$

Figures 1 and 2 show the two elliptic curves are

$$y^2 = x^3 + 2x + 5 \quad (4) \text{ and}$$

$$y^2 = x^3 - 2x + 1 \quad (5)$$

We can see those two equations meet

An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \quad (1)$$

Where *a*, *b*, *c*, *d*, and *e*, are real numbers.

An elliptic group over the Galois Field $E_p(a,b)$ is obtained by computing $x^3 + ax + b \bmod p$ for $0 \leq x < p$. The constants *a* and *b* are non-negative integers smaller than the prime number *p* and as here we used "mod *p*", so equation (3) should be read as:

$$4a^3 + 27b^2 \bmod p \neq 0$$

For each value of *x* one needs to determine whether or not it is a quadratic residue. If it is the case, then there are two values in the elliptic group. If not, then

the point is not in the elliptic $E_p(a,b)$ group. When we fixed a prime number, p and then we can have the Galois Field E_p (a, b) group via the fixed constants a *and* b following the above conditions

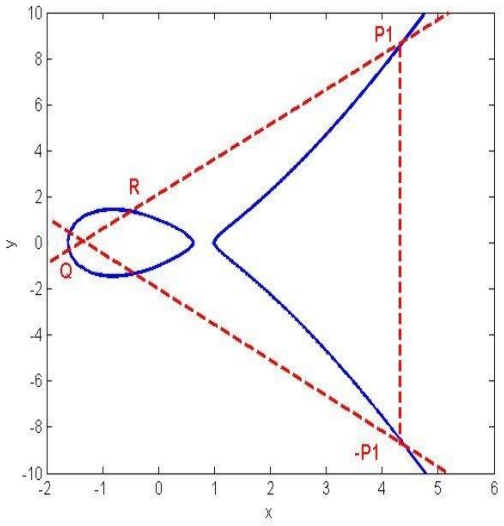


Figure 2. Elliptic curve

For example, let the points $P = (x_1, y_1)$ and Q $(x_2, y2)$ be in the elliptic group $E_p(a,b)$ group and O be the point at infinity. The rules for addition over the elliptic group $E_p(a,b)$ are :

(1) $P+O = O + P = P$

(2) If $x_2 = x_1$ and $y_2 = -y_1$,

that is $P(x_1, y_1)$ and $Q = (x_2, y2) = (x_1\text{-}y_1) = -P$,

that is the case: $P+Q = O$.

(3) If $Q \neq -P$, then their sum $P + Q = (x_3, y_3)$ is given by;

$$x_3 = \lambda - x_1 - x_2 \mod p$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \mod p$$

B. Diffie-Hellman Key Exchange

Diffie–Hellman establishes a shared secret that can be used for secret communications by exchanging data over a public network. The following diagram illustrates the general idea of the key exchange by using colours instead of a very large number. The key part of the process is that Alice And Bob exchange their secret colours in a mix only. Finally this generates an identical key that is mathematically difficult (impossible for modern supercomputers to do in a reasonable amount of time) to reverse for another party that might have been listening in on them. Alice and Bob now use this common secret to encrypt and decrypt their sent and received data. Note that the yellow paint is already agreed by Alice and Bob:

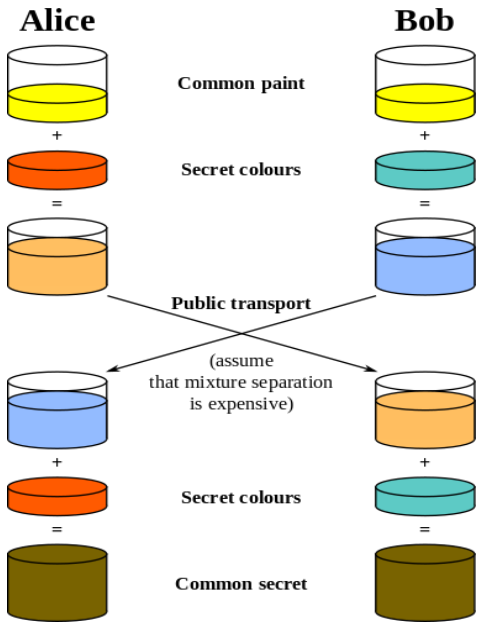


Figure 3. Diffie-helllman key exchange.

Forward secrecy for Google HTTPS

As announced on the Google Security Blog, Google HTTPS sites now support forward secrecy. What this means in practice is two things:

Firstly, the preferred cipher suite for most Google HTTPS servers is ECDHE-RSA-RC4-SHA. If you have a client that supports it, you'll be using that ciphersuite. Chrome and Firefox, at least, support it.

Previously we were using RSA-RC4-SHA, which means that the client (i.e. browser) picks a random key for the session, encrypts it with the server's public key and sends it to the server. Since only the holder of the private key can decrypt the session key, the connection is secure.

However, if an attacker obtains the private key in the future then they can decrypt recorded traffic. The encrypted session key can be decrypted just as well in ten years' time as it can be decrypted today and, in ten years' time, the attacker has much more computing power to break the server's public key. If an attacker obtains the private key, they can decrypt everything encrypted to it, which could be many months of traffic.

ECDHE-RSA-RC4-SHA means elliptic curve, ephemeral Diffie-Hellman, signed by an RSA key. You can see a previous post about elliptic curves for an introduction, but the use of elliptic curves is an implementation detail.

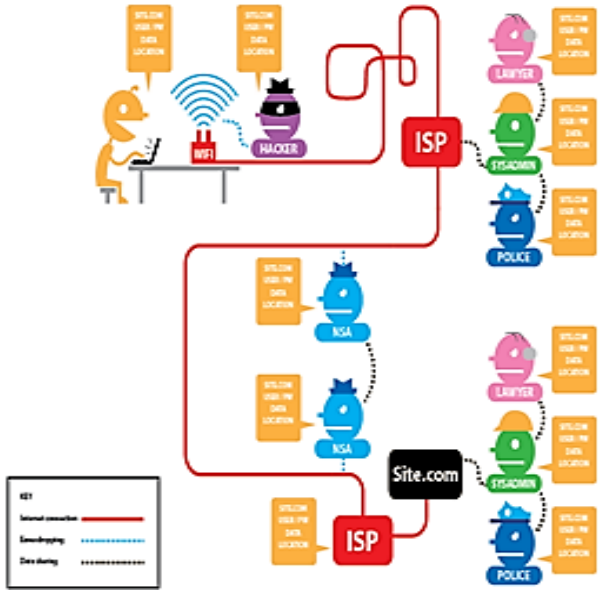


Figure 4. Man-in-middle attack.

Ephemeral Diffie-Hellman means that the server generates a new Diffie-Hellman public key for each session and signs the public key. The client also generates a public key and, thanks to the magic of Diffie-Hellman they both generate a mutual key that no eavesdropper can know.

The important part here is that there's a different public key for each session. If the attacker breaks a single public key then they can decrypt only a single session. Also, the elliptic curve that we're using (P-256) is estimated to be as strong as a 3248-bit RSA key (by ECRYPT II), so it's unlikely that the attacker will ever be able to break a single instance of it without a large, quantum computer.

While working on this, Bodo Möller, Emilia Kasper and I wrote fast, constant-time implementations of P-224, P-256 and P-521 for OpenSSL. This work has been open-sourced and submitted upstream to OpenSSL. We also fixed several bugs in OpenSSL's ECDHE handling during deployment and those bug fixes are in OpenSSL 1.0.0e.

Session Tickets

The second part of forward secrecy is dealing with TLS session tickets.

Session tickets allow a client to resume a previous session without requiring that the server maintain any state. When a new session is established the server encrypts the state of the session and sends it back to the client, in a session ticket. Later, the client can echo that encrypted session ticket back to the server to resume the session.

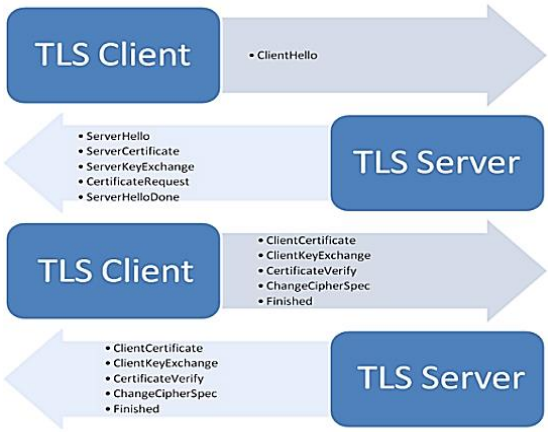


Figure 5. A typical flow of the TLS protocol

Since the session ticket contains the state of the session, and thus keys that can decrypt the session, it too must be protected by ephemeral keys. But, in order for session resumption to be effective, the keys protecting the session ticket have to be kept around for a certain amount of time: the idea of session resumption is that you can resume the session in the future, and you can't do that if the server can't decrypt the ticket!

So the ephemeral, session ticket keys have to be distributed to all the frontend machines, without being written to any kind of persistent storage, and frequently rotated.

III. RESULTS AND DISCUSSION

Proposed work and Performance

A. Diffie-Hellman key exchange using Elliptic Curve (DHECC)

An elliptic curve E over the finite field Fp is given through an equation of the form

$$Y^2 = X^3 + aX + b,$$
$$a, b \in Fp, \text{ and } -(4a^3 + 27b^2) \neq 0$$

Please note that as stated in the beginning of the section, the "=" should be replaced by a "≡" in the above definition. Another remark is that when we talk about partial derivatives we mean the "formal partial derivate" which can be defined (see beginning of this section) over an arbitrary field.

Suppose two communication parties, Alice and Bob, want to agree upon a key which will be later used for encrypted communication in conjunction with a private key cryptosystem.

They first fix a finite field F_q, an elliptic curve E defined over it and a base point $B \in E$ (with high order). To generate a key, first Alice chooses a random $a \in F_q$ (of high order) which she keeps secret. Next she calculates $aB \in E$ which is public and sends it to Bob. Bob does the same steps, i.e. he chooses a random integer b (secret) and calculates b_B which is sent to Alice. Their secret common key is then $P = ab_B \in E$.

Definition . An elliptic curve E over the field F is a smooth curve in the so called "long transform"
$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$, $ai \in F$

We let E(F) denote the set of points (x, y) $\in$ F2 that satisfy this equation, along with a "point at infinity" denoted O.

Remember that smooth means that there is no point in E(F) where both partial derivatives vanish. The definition given above is valid for any field. But in cryptography we are only interested in finite fields. Considering only finite fields we get an "easier" equation. Two finite fields are of particular interest. The finite field Fp with p $\in$ P elements, because of it's structure, and the finite field Fqm with q = pr Elements, since setting p = 2 the arithmetic in this field will be well suited for implementations in hardware.

For generation a shared secret between A and B using ECDH, both have to agree up on EC domain parameters. Both end have a key pair consisting of a private key d(a randomly selected integer less then n, where n is the order of the curve) and public key Q= d*G (G is the generator point). Let (d_A,Q_A) be the private-public key pair of A and (d_B,Q_B) be the private-public key of B.

1. The end A Computes $K_A= (X_A, Y_A) = d_A * Q_B$
2. The end B Computes $K_B= (X_B, Y_B) = d_B * Q_A$
3. Since $d_A * Q_B = d_Ad_BG= d_Bd_AG = d_B * Q_A$. Therefor $K_A= K_B$ and hence $X_A =X_B$
4. Hence the shared secret is K_A.

Since it is practically impossible to find the private key d_A or d_B from the public key K_A

B. Proposed Methods to tow level Encryption Decryption by Diffie – Helman and Elliptic Curve

Today, the scientific efforts are looking for a smaller and faster public key cryptosystem, a practical and secure technology, even for the most constrained environments. For any cryptographic, there is an analogue for Elliptic Curve. One of these systems is Diffie – Helman key exchange system. This paper proposed methods to encrypt and decrypt the message, by using the Diffie–Hellman Exchanging key which is a secrete point in the proposed methods (M1) and (M2)**.**

Diffie – Helman key exchange system

This system is merely a method for exchanging key; no massages are involved. The following algorithm illustrates this system. Suppose two communications Alice and Bob, want to agree upon a key.

They first fix a finite field F_q, an elliptic curve E defined over it and a base point B E (with high order). To generate a key, first Alice chooses a random a F_q (which is approximately the as the number N of point of E) which he keeps secret.

Next, he calculates aB E that is public and sends it to Bob. Bob does the same steps, i.e. she chooses a random integer b (secret) and calculates bB, which is sent to Alice. Their secret common key is then P = abB E. The following algorithm illustrates this system.

The Algorithm of Diffie–Helman key exchange system

- Alice and Bob first choose a finite field F_p and an elliptic curve E defined over it ($E(F_p)$).
- They publicly choose a random base point B E.
- Alice chooses a secret random integer e. He then computes eB E. In addition, send it to Bob.
- Bob chooses a secret random integer d. She then computes dB E. And send it to Alice.
- Then eB and dB are public and e and d are secret.
- Alice computes the secret key edB = e(dB).
- Bob computes the secret key edB = d(eB).

There is no fast way to compute edB if only knows B, eB and dB.

After these setups, Alice and Bob have the same point (only Alice and Bob know it). Then to start with (M1) and (M2), let us consider the following algorithms:

Algorithm of (M1)

Alice and Bob Compute edB = S = (s1, s2). (Using Diffie – Hellman Scheme)
Alice sends a message M $\in$ E to Bob as follows:
Compute (s1 * s2)modN = K.
Compute K*M = C, and send C to Bob.

Bob receives C and decrypts it as follows:
Compute (s1 * s2)modN = K.
Compute (K^{-1})modN.
(where N = #E)
$K^{-1}*C = K^{-1}*K*M = M$.

Algorithm of (M2)

Alice and Bob Compute edB = S = (s1, s2).
Using Diffie – Hellman Scheme)

Alice sends a message M to Bob as follows:
Compute (S_1^{s2})modN = K.
Compute K*M = C, and send C to Bob.

Bob receives C and decrypts it as follows:
Compute S_2^{s2} mod N = K.
Compute (K–1) mod N.
$K^{-1}*C = K^{-1}*K*M = M$.

System Test:

Let **E** be an elliptic curve define over **Fp**
where **p = 3023** with parameters **a = 1, b = 2547**
where **(4a3+27 b2)mod p = 2027 0**.
And **#E = 3083**.

Since **#E** is prime number then by theorem1, every point on **E** in base point, therefore let **B = (2237, 2480)**.

To apply this system test using (M1), at first we must apply Diffie Hellman Exchanging key

- Alice chooses a secret random integer **e = 2313**.
  eB = 2313 (2237 , 2480) = (934 , 29)
  And send **(934, 29)** to Bob .
-
- Bob chooses a secret random integer **d = 1236**.
  dB =1236 (2237 , 2480) = (1713, 1709)
  And send **(1713 , 1709)** to Alice

- Alice computes the secret key **e (dB) = 2313 (1713, 1709)**.
  edB = (2537 , 1632) = S
- Bob computes the secret key **d (eB) =1236 (934, 29)**.
  deB = (2537 , 1632) = S

Now, Alice and Bob have the same point **S = (2537, 1632)** If Alice send a message **M = (2284, 2430)** to Bob

- Compute **(s1 * s2) mod p = (2537 * 1632) mod 3083 = 2998 = K**.
- Compute **K*M = 2998 (2284, 2430)**
  = (2179, 1833)
  =C, and send it to Bob.
- Bob receives **C** and decrypts it as follows:
  - Compute **(s1 * s2) mod p = 2998 =K**
  - Compute **(K–1) mod N = (2998)–1 mod 3083= 1342**
  - **K–1 C =1342 (2179, 1833)**
    = (2284, 2430)

To apply this system test using the algorithm (M2), at first we must apply Diffie–Hellman Exchanging key.

By the same procedure to solve Diffie–Hellman scheme we have obtained

S = (2537, 1632)

If Alice sends a message **M = (2284, 2430)** to Bob using (M2), he does the following:

- Compute (S^{s2}) **mod N = (25371632) mod 3083= 323=K.**
- Compute **K* M = 323 (2284, 2430) = (2555, 1066) =C**, and send it to Bob.

- Bob receives **C** and decrypts it as follows:

- Compute (S_1^{s2}) **mod N = 323=K.**
- Compute **(K–1) mod N = (323)–1 mod 3083= 1594.**
- **K–1 C = 1594 (2555 , 1066) = (2284, 2430) = M.**

Performance Performance of Elliptical Cryptography with Deffie Hellman Key Exchange will depend on the hardware as well as the quality of the JavaScript execution environment. The following table shows the times taken for various public-key operations on a cross-section of browsers and hardware.

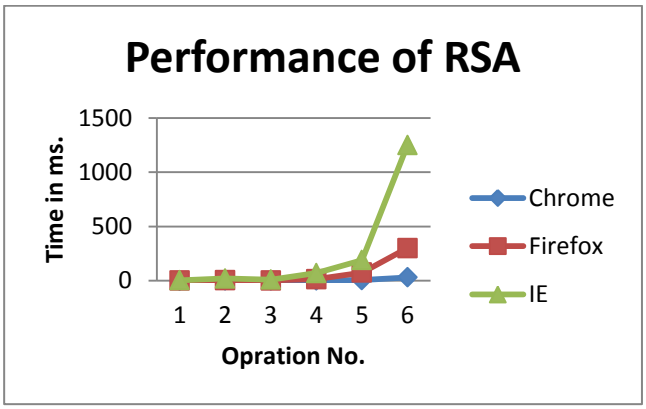


Figure 6 : Performance of RSA

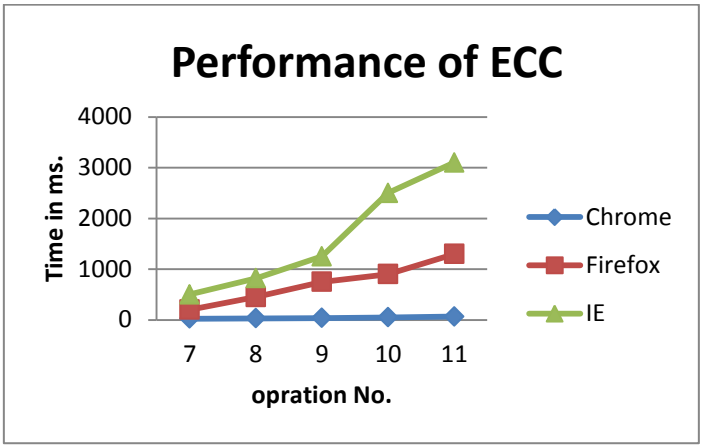
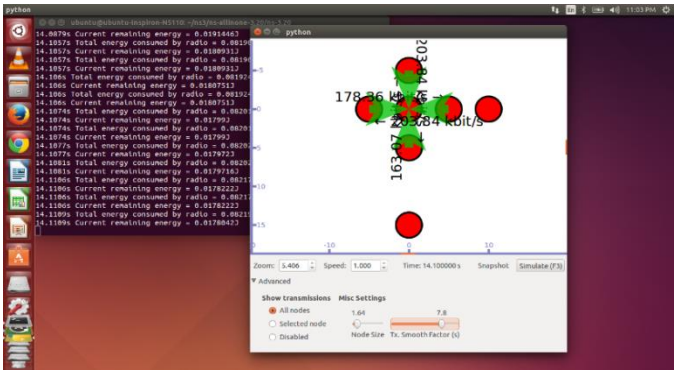


Figure 7 : Performance of ECC

Table 1 : Cross Section of Browsers and Hardware

	Operation	Chrome (ms)	Firefox (ms)	IE (ms)
1	RSA public, 512 bit, e=3	0	1	4
2	RSA public, 512 bit, e=F4	1	6	20
3	RSA public, 1024 bit, e=3	1	3	10
4	RSA public, 1024 bit, e=F4	2	15	70
5	RSA private, 512 bit	5	75	190
6	RSA private, 1024 bit	30	300	1250
7	EC multiply, 128 bit	25	200	500
8	EC multiply, 160 bit	30	450	820
9	EC multiply, 192 bit	35	750	1250
10	EC multiply, 224 bit	50	900	2500
11	EC multiply, 256 bit	65	1300	3100



(A)In simulation red dots shows mobile nodes and at center Sink node and Green lines indicate

communication links crested during simulation between mobile nodes and sink nodes.

IV. CONCLUSION

The The Diffie Helman scheme is one of the exchanging key cryptosystem, no massages are involved in this scheme, in this report, and we try to benefit from this scheme by use the key (which exchange it) as a secret key. (That is, we know now the one of the advantages of the Diffie–Helman key exchange system).

In our proposal work, we address the essentialness conservation issue to enable group in-framework get ready in broad scale WSNs. We consider WSNs made out of homogeneous remote sensors accumulated into gatherings, inside which applications are iteratively executed. Since imperativeness use capability is a champion amongst the most essential thoughts for any WSN game plan, our proposed plans intend to achieve essentialness adequacy from various viewpoints. To redesign information get ready utmost in WSNs, plan length streamlining is moreover bit of our framework objectives. The dedication of this investigation can be packed as takes after. Centers may be equipped with different sensors recognizing particular events. Dependent upon applications, the recognized events may happen in a discontinuous case. Therefore, a component intra-sensor arranging count should be proposed to handle these events and capably allocate sensor resources. We proposed two different methods to encrypt and decrypt the message for the WSN. In the second method, we support the system more security of the first method, because the sender compute the exponentiation function between the coordinates of the key in the encryption algorithm (use fast exponentiation method), and the receiver compute the inverse of the exponentiation function between the coordinates of the key in the decryption algorithm. While in the first method, the sender compute the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm.

We proposed two different methods to encrypt and decrypt the routing data of WSN. In the second method, we support the system more security of the first method, because the sender compute the exponentiation function between the coordinates of the key in the encryption algorithm (use fast exponentiation method), and the receiver compute the inverse of the exponentiation function between the coordinates of the key in the decryption algorithm. While in the first method, the sender compute the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm

V. REFERENCES

[1] V. S. Miller, "uses of elliptic curves in cryptography," in Advances in Cryptology, CRYPTO’85, ser . Lecture Notes in Computer Science, vol. 218, Springer, 2014. pp. 417-428.

[2] N. Koblitz, " Elliptic curve cryptosystems," Mathematics of Compution, vol. 48, no.177, pp.203-209, Jan 2013.

[3] D. Hakerson, A. Menezes, and S. Vanston , "Guide to Elliptic Curve Cryptography," Springer-Verlag, NY (2014).

[4] H. Cohen, A Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Lectures Notes in Computer Science, 1514, 51-65 (2012).

[5] V. Dimitrov V., L. Imbert, and P. K. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," Lectures Notes in Computer Science, 3788, 59-78 (2013).

[6] M. Ciet, M. Joye, K. Lauter and P.L. Montgomery,"Trading inversions for multiplications in elliptic curve cryptography," Designs, Codes, and Cryptography, 39, 189-206 (2011).

[7] D. Bernstein, "High-speed diffie-hellman, part 2," presented at the INDOCRYPT’06 tutorial session, Dec. 11-13, Kolkata, India (2006).

[8] K. Kaabneh and H. Al-Bdour, "Key exchange protocol in elliptic curve cryptography with no public point," American Journal of Applied Sciences 2 (8): 1232-1235, 2005.

[9] J. Adikari, V. Dimitrov, and L. Imbert, "Hybrid binary-ternary joint sparse from and its application in ellipic curve cryptography," Cryptology ePrint Archive, Report 2008/285, 2008.

[10] Bangju Wang, Huanguo Zhang and Yuhua Wang, "An efficient elliptic curves scalar multiplication for wireless network," 2007 IFIP