ECDH based Security Model for IoT using ESP8266

Ravi Kishore Kodali and Ashwitha Naikoti

Department of Electronics and Communication Engineering

National Institute of Technology,Warangal

WARANGAL 506004, INDIA

Abstract-Now a days, Internet of Things (IoT) is the emerging trend in technology that aims at making our life simpler using sensors and smart devices. All of them are connected to the Internet. Because of this it is possible to access and control them from any place and at any time. The most important aspect in this networking is communication between the smart devices. The correctness in the behavior of the devices mostly depends on its efficiency to send the data properly. Thus security is very important in IoT implementations. In this paper Elliptic Curve Diffie-Hellman(ECDH) key exchange on NIST P-192 curve for secured communication between ESP8266 modules has been discussed and implemented using NodeMCU.

I. Introduction

The idea of Internet of Things is spreading to be part and parcel of our lives. The devices used in domestic automation, industry, infrastructure and other smart applications are interconnected with the internet. Because of this there will be a variety of data that is read, assembled and transmitted in an efficient and secure manner.[2]

To achieve the task of connecting devices ESP8266 is used which provides embedded Wi-Fi capabilities at lowest cost with good functionality[8]. It is a system on chip(SOC) which is self contained and has an integrated TCP/IP protocol stack to give access to WiFi network. It is a low power technology. The widespread internet connectivity gives rise to high security problems such as unauthorized access to the devices, eavesdropping and threat to privacy. As these interconnected devices can be accessed and controlled at anytime and from anywhere, authentication and authorization becomes essential. The following are the criteria that must be taken into account while selecting the type of public-key exchange methods. [1]

1) Functionality
2) Security
3) Performance

The difficulty involved in solving the mathematical problem employed in the public key protocol is very important as it is the main factor that decides the strength of the cryptography method used. The hardness of this problem reflects the execution as it decides the main elements like the sizes of the domain and key parameters[1]. Elliptic Curve Cryptography has taken over RSA in many applicatons beacuse it is very efficient. The size of the parameters is very much compressed in Elliptic Curve Cryptograph(ECC) than with RSA and DL schemes when compared at the same security level. High speed computations, reduced key sizes and certificates are the advantages gained by using ECC. Comparison of key sizes of ECC and RSA is shown in Table 1 [5]. It is used in applications where power,storage and bandwidth are constrained.

TABLE I
KEY SIZE COMPARISON(IN BITS)

ECC	RSA	Symmetric Algorithms
163	1024	80
233	2240	112
283	3072	128
409	7680	192
512	15360	256

ECC becomes the best choice for authentication and authorization of IoT devices because of the storage capacity of ESP8266. The module has various variants. Each variant is an improvement over the previous one in terms of the hardware of the module. ESP8266-01 is the cheapest and it is has the least features. ESP8266-13 is the most expensive of them with most features. The various features include the types of pins, number of GPIO pins, presence of shield and antenna on the chip, type of chip packaging and storage capacity. The ESP8266 module requires 3.3V and upto 250mA power supply. It is power efficient and low in cost.[8].

TABLE II
COMPARISON OF PRICES

ESP8266-01	$ 5
Ethernet shield for Arduino	$ 60
Zigbee	$ 25
Wify shield sparkfun	$ 40
Wi-Fi shield for Arduino	$ 80
Huzzah Wi-Fi shield by Adafruit	$ 40
ESP8266-12	$ 7

978-1-5090-5240-0/16/$31.00 ©2016 IEEE

II. Overview of Elliptic Curve Cryptography

Victor Miller and Neal Koblitz selfstandingly proposed Elliptic Curve Cryptography[3][4]. In these cryptosystems an elliptic curve is considered on which the members of the group are defined. The operations between these members is also defined on the curve[9]. ECC has been standardized by organizations such as NIST(National Institute of Standards Technology)[14], IEEE(Institute of Electrical and Electronics Engineers)[12], ISO(International Standards Organization)[13], and ANSI(American National Standards Institute)[11]. This made it widely commercially accepted.

A. Prime Field

A prime field is variant of finite field or galois field GF(p) where p is a prime number. A prime field denoted by F_p comprises of integers modulo p with the results of all the operations such as addition and multiplication also belonging to the same field. It is a finite field of order p. All the elements belonging to this field are $0, 1, 2, ..., p-1$

B. Elliptic Curves

Choose p such that it is a prime number, and let F_p denote the prime field. An elliptic curve **E** is defined over F_p. All the points on this curve belong to the defined prime field. These points satisfy the equation given below. Here the pair (x,y) represent the cartesian coordinates of a point on the curve.

$$y^2 = x^3 + ax + b \qquad (1)$$

a,b are also the elements of the prime field F_p. They must satisfy the condition $4a^3 + 27b^2 \neq 0$ (mod p).

C. Domain Parameters

Other than the curve parameters the parties involved must agree on some other parameters for secure communication in ECC. These parameters are known as domain parameters. A suitable elliptic curve over a prime field is chosen. This gives rise to a set of parameters called domain parameters[10]. They are summarized as following

1) p is the prime number which is chosen for defining the prime field.
2) *a, b* are the two parameters in the curve equation on which the shape of the curve depends
3) G is the Base point or Generator point which is a point on the elliptic curve over F_p selected for performing the elliptic curve operations
4) n is called the order of a point D on the elliptic curve. It is the smallest integer such that $nD = 0$. The operation performed for calculating nD is called point operation and is discussed in the later parts of this paper.
5) h is the cofactor of the curve

D. NIST p-192

A set of elliptic curves were recommended for the benefit of the Federal Government. All the curves which are proposed over a prime field have unity cofactor and the curve equation is given by E: $y^2 = x^3 - 3x + b$ mod p. Curves were proposed for 192, 224, 256, 384, and 521 bits. In this paper NIST p-192 is used[6][14].

- p=6277101735386680763835789423207666416083908700390324961279
- a=6277101735386680763835789423207666416083908700390324961276
- b=0x64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1
- G_x=0x188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012
- G_y=0x07192b95ffc8da78631011ed6b24cdd573f977a11e794811
- n=0xffffffffffffffffffffffff99def836146bc9b1b4d22831

E. Finite Field Arithmetic

1) **Inversion:** Inverse of a nonzero element x which belongs the finite field F_p, denoted by x^{-1} mod p is an other unique element m which also belongs to the same finite field F_p such that mx=1 mod p. It is efficiently computed by using extended euclidean algorithm[7].

The modular inversion is calculated by first finding the greatest common divisor of x and p. The following algorithm is used for finding the greatest common divisor. The values returned by the algorithm is the gcd of the two numbers x,p and the also the values of c,y such that cx+py=gcd(x,p).

Algorithm 1 Extended Euclidean Algorithm

```
1:  procedure EUCLID(x, p)        ▷ g=gcd of x and p and c,y such that cx+py=gcd(x,p)
2:      a ← x
3:      b ← p
4:      u1 ← 1
5:      v1 ← 0
6:      u2 ← 0
7:      v2 ← 1
8:      while a ≠ 0 do
9:          q ← ⌊b/a⌋
10:         t ← b − qa
```

Suppose that p is a prime number and $x\epsilon$[1,p-1], and hence greatest common divisor of x and p is 1. If the above extended euclidean algorithm is used we get t=1 in this case. The integer u_1, and v_1 satisfy au_1+pv_1=1. When a modulus over p operation is performed on this equation we get pv_1 mod

```
        k ← u2 − qu1
        l ← v2 − qv1
        b ← a
        a ← t
        u2 ← u1
        u1 ← k
        v2 ← v1
        v1 ← l
    end while
    gcd ← b
    c ← u2
    y ← v2
    return gcd, c, y
end procedure
```

$p = 0$. Hence au_1=1 mod p and thus u_1 is the inverse of *a* in the prime field.

Algorithm 2 Inversion in F_p using Extended Euclidean Algorithm

```
procedure INVERSION(x, p)        ▷ Inversion of x∈[1,p-1] mod p
    a ← x
    b ← p
    u1 ← 1
    u2 ← 0
    while u ≠ 0 do
        q ← ⌊b/a⌋
        t ← b − qa
        k ← u2 − qu1
        b ← a
        a ← t
        u2 ← u1
        u1 ← k
    end while
    return u1                    ▷ x^-1 mod p
end procedure
```

2) **Point Addition and Point Doubling:** Point addition is the method in which two points M and N on an elliptic curve are added to get another point which is also on the same elliptic curve over a prime field F_p.

Fig.1 explains the concept of point addition on an elliptic curve as defined before. To find the sum of two points M and N draw a line joining M and N and also to cut the elliptic curve. A perpendicular is dropped from this point to meet the elliptic curve at S. This point S gives the sum M+N [15].

Point doubling is method of adding a point M to the same point to obtain a point S on the elliptic curve over a prime field F_p.

Let M be the point on which the point doubling has to be performed. Draw a tangent at this point M. This tangent meets the elliptic curve. Drop a perpendicular from this point.

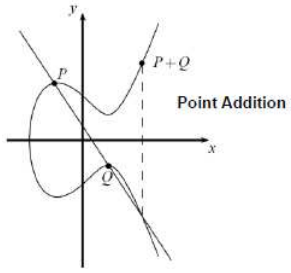


Fig. 1. Addition:M+N=S

This perpendicular meets the elliptic curve at S. This S is the desired point which is obtained by doubling of the point M [15]. Fig.2 shows how doubling is done.

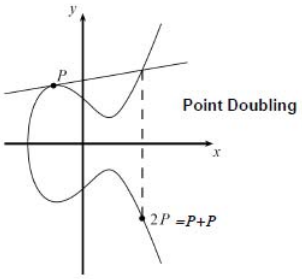


Fig. 2. Doubling:M+M=S

The concept of adding two points on the curve defines the following point operations on the elliptic curve.

1) *Identity:* M+∞ = ∞+M = M for a point M on the elliptic curve over the prime field
2) *Negative of point M:* If M = (x,y) then the negative of M is given by -M = (x,-y). This is also a point on the elliptic curve over the prime field.
3) *Point Addition:* Let the two points to be added on the curve be M=(x_m,y_m) and N=(x_n,y_n). Then the sum S=M+N is given by (x_s,y_s) such that

   x_s=$(\frac{y_n-y_m}{x_n-x_m})^n$ - x_m - x_n mod *p*

   y_s=$(\frac{y_n-y_m}{x_n-x_m})$($x_m$-$x_s$) - y_m mod *p*

   Here $(\frac{y_n-y_m}{x_n-x_m})$ is the slope of the line joining M and N on the curve.

4) *Point Doubling:* Let P=(x_1,y_1) be the point on the elliptic curve which has to be doubled. Then S=2M = (x_s,y_s), where

   x_s=$(\frac{3{x_m}^2+a}{2y_m})^2$ - 2x_m mod *p*

   y_s=$(\frac{3{x_m}^2+a}{2y_m})$($x_m$-$x_s$) - y_m mod *p*

3) **Point Multiplication:** This is also known as Scalar Multiplication. It involves calculation of bM, here b belongs to the prime field and M is a point on the elliptic curve defined over the same prime field F_p. This operation is done by repeated doubling and adding of the point on the curve till the point is multiplied for the required number of times. The following algorithm is followed for the implementation of scalar point multiplication on an elliptic curve. Binary Left to Right method is used.

Algorithm 3 Point Multiplication

procedure SCALAR MUL($b = (b_{k-1}, ...b_2, b_1)_2$, M ϵ E(F_p))
  $R \leftarrow \infty$
  for `i=k-1 to 0` **do**
    $R \leftarrow 2R$
    If $b_i = 1$ **then** $R \leftarrow R + M$
  end for
  return R
end procedure

F. Generation of key using Elliptic Curve

Let the defined elliptic curve be denoted by E. It is defined over a prime field F_p. Let T be a point on such an elliptic curve E(F_p), and T must have a prime order n [1].

Algorithm 4 Key Generation on an Elliptic Curve

procedure KEY(p, E, G, n)
  $Private \leftarrow [1, n-1]$ ▷ a value in the interval is chosen
  $Public \leftarrow kT$ ▷ Point Multiplication
  return $Public, Private$
end procedure

A number is selected randomly such that it belongs to the prime field. This number is called the private key. It is multiplied to the base point or the generator point. The result of this point multiplication is the public key.

Diffie-Hellman Key Exchange: Elliptic Curve Diffie-Hellman is a protocol used for key agreement that lets two entities to generate a secret key that will be used for operations involving the private key. The public key generated by one individual is shared with the other. Elliptic curve operations are used for generating the secret key. We consider that Alice and Bob agree on a common key exchange protocol for exchanging data. Assumption is made that they had no former contact and the mode of communication available between them through the channel is only public. Both of them exchange some public data or public key with each other. Each of them have a private key which is used to generate the shared key called the public key. Both the individuals agree on same domain parameters. The following steps are followed in this protocol.

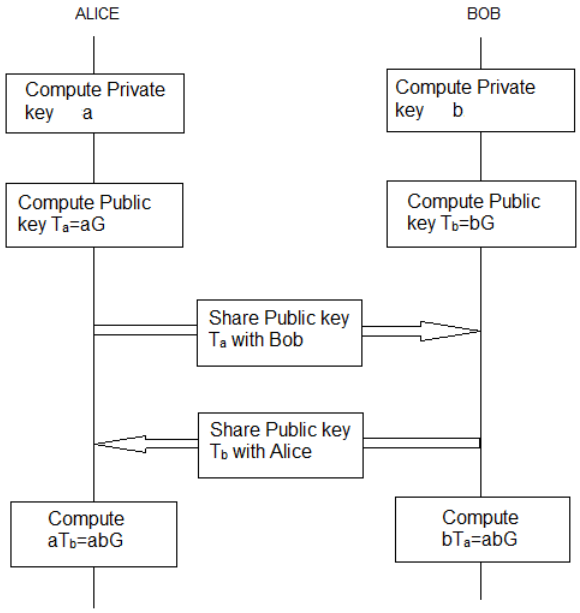


Fig. 3. ECDH Key Exchange

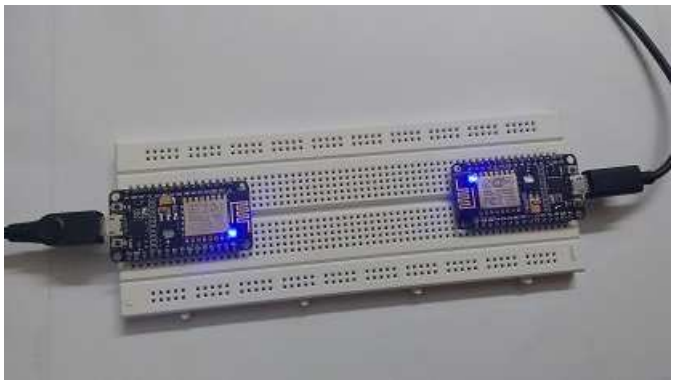
1) Alice and Bob choose a common elliptic curve E over a prime field F_p. They also comport on a base point G ϵ E(F_p) so that the subgroup generated by G has larger group cardinality. This decides the strength of the method involved.[16]
2) Alice chooses an integer a. This is a secret key and is not shared with anyone. This is the private key of Alice. It then performs point multiplication and calculates the public key T_a=aG, and sends T_a to Bob.
3) Bob also selects an integer b which becomes his private key, calculates T_b=bG by point multiplication, and sends T_b to Alice.
4) Alice computes aT_b=abT. This is done by point multiplication of the secret key of Alice with shared key of Bob.
5) Bob performs point multiplication between private key of Bob and public key of Alice and computes bT_a=abT.

The sole data that an eavesdropper can get is about the elliptic curve E, the finite field F_p and the points G, aG, bG. It is difficult to calculate the shared secret with only this information.

III. IMPLEMENTATION AND RESULT

In this paper C++ and BigNumber library in PlatformIO are used for implementing the Elliptic Curve Diffie Hellman key exchange. PlatformIO is a platform for IoT development. It is an open source environment. Many platforms are integrated into a single environment . It has many libraries for different platforms like Arduino and MBED. It supports more than 200

development boards along with more than 15 development platforms and 10 frameworks [19].
ESP8266 has a 32-bit Microcontroller running at 80Mhz. It has 64KB of instruction RAM and 96KB of data RAM. The program uses 234,729 bytes of program storage space out of 1,044,464 bytes available on the device.



Result:

Private key of Alice :
3414121761587732966806895359043733035535109630469222112985
Public key of Alice :
(51816790119107273625873865765880617692097015020142359699 24, 51333049748951334482010415493877324297543596965816350375 18)

Private key of Bob :
4805327472301777801070369516874236604158393374114880750942
Public key of Bob :
(45193790228990435009008901963536601048026242162644065252 37,50273334552610273548545950359029665473407341859870941 46271)

Verification at Alice (aT_b) :
(35413485630412598676400949136684953820106314253240116414 07, 55810722140998750193554349565706773225433024153575938 60274)

Verification at Bob (bT_a) :
(35413485630412598676400949136684953820106314253240116414 07, 55810722140998750193554349565706773225433024153575938 60274)

After the key exchange, verification is done by both Alice and Bob. The computations by both the individuals result in the same point on the curve **E**. This authenticates the connection established between them.

IV. Conclusion

Elliptic Curve Diffie Hellman(ECDH) key agreement scheme which is one of the variants of Elliptic Curve Cryptography forms the basis for security authentication. This has grabbed the attention of the industry in recent times because of its advantages over RSA and AES. Many improvements can be made in securing the communication using this agreement protocol. Implementing such an algorithm on a potential device like ESP8266 adds to the advantage. With the IoT going to be the next Industrial Revolution, secure and low-cost WiFi device plays a vital role. This can be used in many applications such as Mesh networks, Home automation, Smart power meters, Wearable devices, Security ID tags and Sensor networks.

References

[1] Darrel Hankerson, Alfred Menezes, Scott Vanstone *"Guide to Elliptic Curve Cryptography"*.
[2] Sye Loong Keoh, Sandeep S.Kumar, and Hannes Tschofenig *"Securing the Internet of Things/; A Standardization Perspective"* June 2014.
[3] V.S.Miller, Use of Elliptic Curves in Cryptography, *Advances in Cryptography*, 1985.
[4] N.Koblitz, Elliptic Curve Cryptosystems, *Mathematics of Computation* 1987.
[5] Rounak Sinha, Hemant Kumar Srivatsava, Sumita Gupta, *"Performance Based Comparison Study of RSA and Elliptic Curve Cryptography", IJSER, Vol:4, Issue:5* May 2013.
[6] *ECC Brainpool Standard Curves and Curve Generation* v 1.0, 19.10.2005.
[7] Alfred J.Menezes, Paul C.van Oorschot, Scott A. Vanstone, *"Handbook of Applied Cryptography"*.
[8] Manan Mehta *"ESP8266: A Breakthrough in wireless sensor networks and internet of things"*, IJECET, Vol:6, Issue:8, August 2015.
[9] Alfred J.Menezes, Paul C.van Oorschot, Scott A. Vanstone, *"Handbook of Applied Cryptography"*
[10] Aqeel Khalique, Kuldip Singh, Sandeep Sood *"Implementation of Ellipic Curve Digital Signature Algorithm, IJCA, Vol:2, No-2"* May,2010.
[11] ANSI X9.62,*Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 1999.
[12] IEEE 1363-2000,*Standard Specifications for Public-Key Cryptography*, 2000.
[13] ISO/IEC 15946,*Information Technology Security Techniques Cryptographic Techniques Based on Elliptic Curves*, Committee Draft (CD), 1999.
[14] NIST, *Digital Signature Standard, FIPS Publication 186-2*, February 2000.
[15] Christof Paar, Jan Pelzl*Understanding Cryptograpghy*,2015.
[16] Harald Baier, Johannes Buchmann, *"Generation methods of Elliptic Curves"*, August, 2002
[17] M. A. Strangio,*"Efficient Diffie-Hellmann two-party key agreement protocols based on elliptic curves"*, Proc. 20th ACM Symposium on Applied Computing (SAC), pp. 324-331, 2005.
[18] S. Wang, Z. Cao, A. Strangio, L. Wang,*"Cryptanalysis and improvement of an elliptic curve Diffie-Hellman key agreement protocol"*, IEEE Commun. Lett., vol. 12, no. 2, pp. 149-151, Feb. 2008.
[19] *"What is PlatformIO"* Retrieved from http://docs.platformio.org/en/stable/what-is-platformio.html
[20] *"ECC tutorial"* Retrieved from https://www.certicom.com/ecc-tutorial