ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 9, November 2012

Forward Secrecy For Google HTTPS using Elliptic Curve Diffie-Hellman Key Exchange Algorithm

Neha Jha
Deptt of Computer Science.
M.Tech (4th SEM),CTA
SRIT, Jabalpur(M.P) India

Brajesh Patel
Deptt of Computer Science.
HoD (Com. Sci.)
SRIT, Jabalpur(M.P) India

ABSTRACT :-
In today's era of the ubiquitous computing, the Internet has become the main mode of data communication. In such a environment, providing security to data becomes a complex task. In Elliptic curve cryptography (ECC) is an emerging favorite because requires less computational power, communication bandwidth, and memory when compared to other cryptosystems. In this paper we present Elliptic curve cryptography and Diffie–Hellman key agreement, itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide forward secrecy for web browsers application using HTTPS . In its popular deployment on the internet, HTTPS provides authentication of the web site and associated web server that one is communicating with, which protects against Man-in-the-middle attacks. Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with and/or forging the contents of the communication.

Keywords:- Elliptic curve, diffie hellman, attacks, SSL,HTTPS.

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 competitors increases, as the security needs increase over time.Here we present Diffie_hellman key exchange algorithm providing forward secrecy for web browsers application. DHKE is a specific method of exchanging cryptographic keys,allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

II. TRADITIONAL PROTOCOL IN ELLIPTIC CRYPTOGRAPHY

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

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

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)

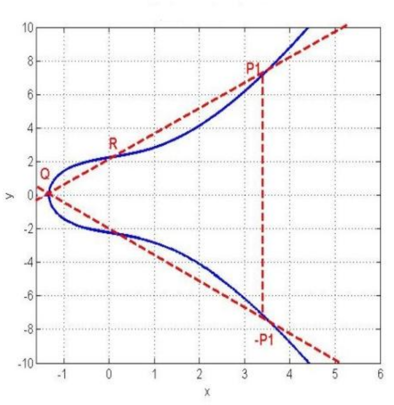

Fig.1 Elliptic curves

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: $f(x) = x^3 + ax + b$:

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

Figures 1 and 2 show the two elliptic curves are

$$y^2 = x^3 + 2x + 5 \qquad (4) \text{ and}$$
$$y^2 = x^3 - 2x + 1 \qquad (5)$$

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:

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

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

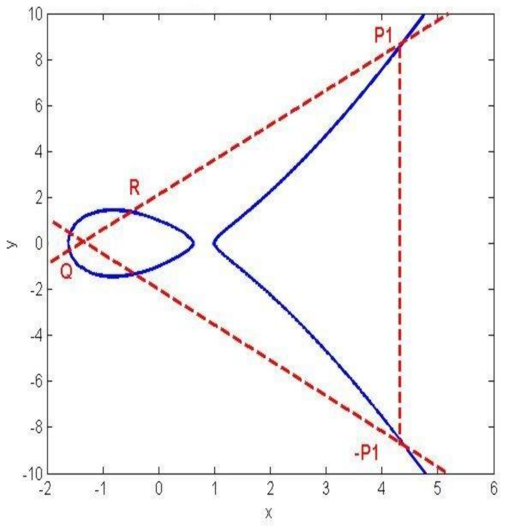

Fig.2 Elliptic curve

For example, let the points $P = (x_1, y_1)$ and Q (x_2, y_2) 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 $Q = (x_2, y_2) = (x_1 {-} y_1) = -P$, that is the case: $P + Q = O$.

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

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

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

III. 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:

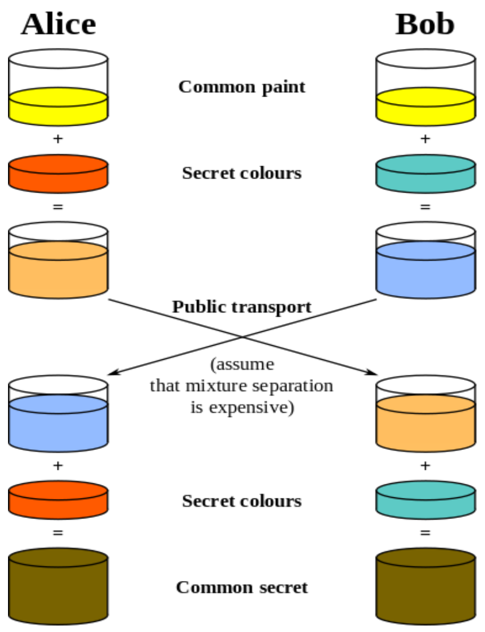

Fig 3: Diffie-hellman key exchange.

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.

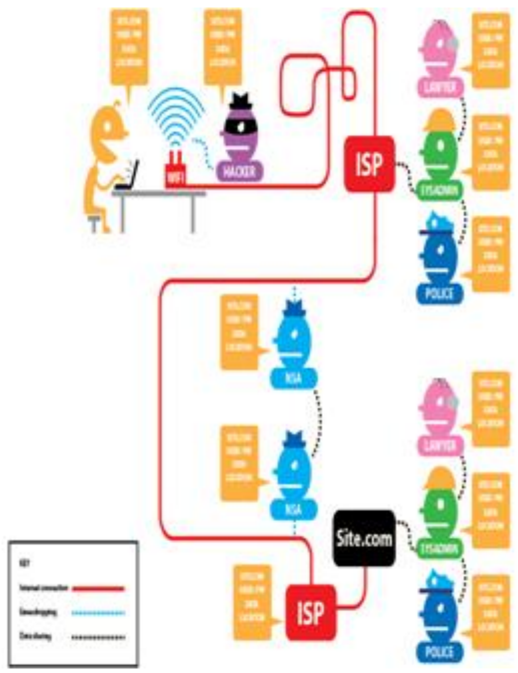

Fig 4: Man-in-middle attack.

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.

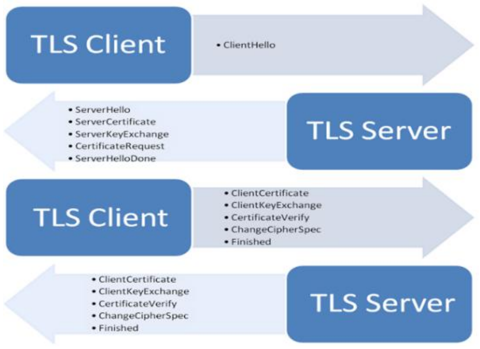

Fig 5: A typical flow of the TLS protocol

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.

IV. PROPOSED WORK AND PERFORMANCE

4.1 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

4.2 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 ∈ 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_1^{s2})mod N = K.
Compute (K–1) mod N.
$K^{-1}*C = K^{-1}*K*M = M$.

*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.

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 9, November 2012

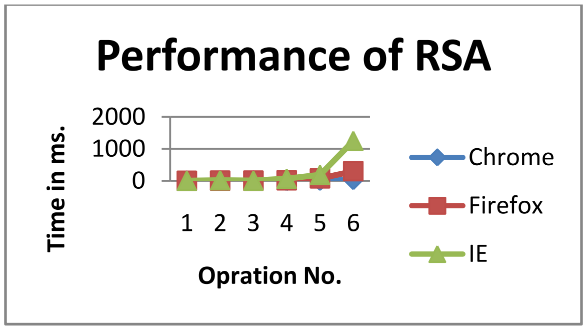

Fig 6: Performance of RSA

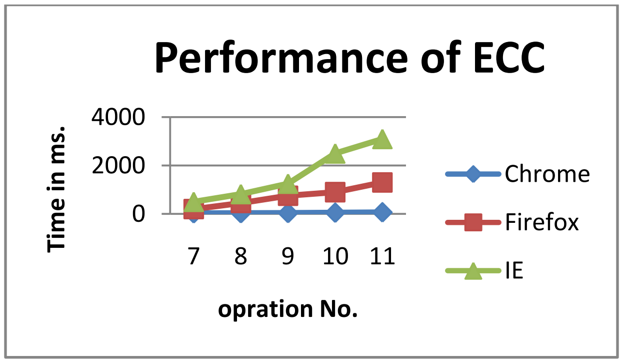

Fig 7: Performance of ECC

EC multiply = Elliptic curve point multiplication, bit size denotes both curve prime size and scalar multiplier size.
PC = Win7 64-bit, Intel Core i5 M520 (2.4GHz)

Browsers used:

Google Chrome version 10.0.648.151
Mozilla Firefox version 3.6.15
Microsoft Internet Explorer version 8.0.7601.17514

V. CONCLUSION

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) and we are using Elliptic curve cryptography for encryption and Decryption .

Table1: 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

We proposed two different methods to encrypt and decrypt the message. 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 and we can use our approach for forward secrecy in HTTPS protcol.

VI. 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, 1986. pp. 417-428.

[2] N. Koblitz, " Elliptic curve cryptosystems," Mathematics of Compution, vol. 48, no.177, pp.203-209, Jan 1987.

[3] D. Hakerson, A. Menezes, and S. Vanston , "Guide to Elliptic Curve Cryptography," Springer-Verlag, NY (2004).

[4] H. Cohen, A Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Lectures Notes in Computer Science, 1514, 51-65 (1998).

[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 (2005).

[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 (2006).

[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

NehaJha
Deptt of Computer Science
M.Tech (4th SEM),CTA,
SRIT,Jabalpur(M.P)India.

Brajesh Patel
Deptt of Computer Science.
HoD (Com. Sci.)
SRIT,Jabalpur(M.P)India