A SECURE DIFFIE-HELLMAN SCHEMES OVER ELLIPTIC CURVES

Malek Jakob Kakish
Amman Arab University,
Department of Computer Information Systems, P.O.Box 2234, Amman 11953, Jordan
Email: doctor_malek@yahoo.com; malek@aau.edu.jo

ABSTRACT

The protection of information technologies is very essential because information technologies play a major role in our information society, such protection includes data and system protection against many kinds of threats or attacks which may lead to lose of money, or lose of reputation and thus destroy businesses.

The Diffie-Hellman public key cryptosystem over elliptic curves is often used in modern communications and system technologies; it is one of the firstly defined public key cryptosystem that enable secure communicating over public unsecure communication channels.

This paper introduce a security Diffie-Hellman cryptosystem based on elliptic curves, it suggests the use of randomization in the encryption process to become immune against many attacks described in literature, this proposed security enhancement describe both the diffie-Hellman key exchange process and the Diffie-Hellman cryptosystem, this enhancement makes the Diffie-Hellman semantically secure, because an attacker will not be able to distinguish between two encryptions even if the attacker knows the corresponding plaintexts. Other important benefit is that the Diffie-Hellamn cryptosystem described here can easily be implemented and is very suitable on small and limited devices (e.g. smart cards) due the use of elliptic curves.

This paper also briefly investigate some attacks on the Diffie-Hellman scheme and the suitable choice of Diffie-Hellman parameter to avoid such attacks.

Keywords: *Diffie-Hellman cryptosystem, Diffie-Hellman key exchange protocol, elliptic curves, public key cryptosystems, cryptosystem analysis, finite fields*.

1. INTRODUCTION

The term information security is a major term that includes securing many components which process, transmit, store or modify information, to fulfill such requirements we often need to have suitable security technologies and security systems that meet the security needs and requirements.

Many security systems (e.g. security protocols, algorithms or applications) have been developed that are based on standards; such standards are mostly specified from well known standard organizations (e.g. Internet Architecture Board (IAB), Internet Engineering Task Force (IETF), etc.) to ensure compatibility and coordinated work in the development of such systems and to support our security needs in information technology.

Security needs and wishes cannot be achieved in one single mechanism; however, we can note that a major science called Cryptography (the science of data encryption and decryption) underlies many of the security mechanisms. Cryptography [1] enables users to securely store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient. By using such a powerful tool as encryption we gain privacy, authenticity, integrity, and limited access to data.

In Cryptography we differentiate between private key cryptographic systems (also known as conventional cryptography systems) and public key cryptographic systems. Private key cryptography, also known as secret-key or symmetric-key encryption, has an old history [1], and is based on using one key for encryption and decryption. In the 1960s many modern private key cryptographic systems where developed that are based on Feistel cipher, e.g. Data Encryption Standard (DES), Triple Data Encryption standards (3DES), Advanced Encryption Standard (AES), The International Data Encryption Algorithm (IDEA), Blowfish, RC5, CAST, etc.

In 1976 Diffie and Hellman [2] published a paper that describes a new concept which was called public-key cryptography and is based on using two keys (public and private key). This new concept solved many weaknesses and problems (e.g. key exchange problem) in private key cryptography, since then many public key cryptographic systems were invented (e.g. RSA [3], ElGamal [4], Diffie-Hellman key exchange [2], elliptic curves [5] [6], etc.). The security of such Public key cryptosystems is based on apparently difficulties of some mathematical number theory problems ("also called one way functions") e.g. the discrete logarithm problem over finite fields and over elliptic curves, the integer factorization problem or the Diffie-Hellman Problem, etc. For more information about Cryptography histories see [1].

One of the firstly defined public key cryptosystem and often used to secure keys exchange over public networks is the Diffie-Hellman key agreement protocol, such protocol is often an essential part of authentication protocols or part of a whole security system, e.g. Diffie-Hellman is used in Internet security standard protocols IPSEC [8] to secure transmitted data through public networks and is mostly used in web and email communication systems today. The Diffie-Hellman protocol is also called key exchange protocol and has been patented in 29 of April 1980, in the USA under Patent 4,200,770 (in 6 Sep. 1997 expired), it is assigned to Stanford University and covers the Diffie Hellman key agreement and mention three persons, Hellman, Diffie, and Merkle as inventors [2].
In praxis there are many security standards specifications which define the implementation and the use of Diffie-Hellman encryption/decryption or key agreement protocol [9] [10]. Due to the widely use of the Diffie-Hellman schemes is it critical to ensure a high level of security, in this paper I introduce a Diffie-Hellman cryptosystem over elliptic curves that is more secure compared with the basic version of the Diffie-Hellman, this is achieved by using randomization in the key agreement protocol and the encryption process, this will make it more difficult for an attacker or cryptanalysis people to break the Diffie-Hellman schemes.

2. PROBLEM FORMULATION

The security of many in the praxis used cryptosystems and protocols are based on the Diffie-Hellman problem, this means that if in the future the Diffie-Hellman problem is efficiently solved then many of these cryptosystems will no longer be secure, e.g. the well known El-Gamal cryptosystem over elliptic curve, his security depends on the intractability of the Discrete Logarithm Problem as well as the Diffie Hellman problem.
The Diffie-Hellman schemes considered in this paper use Z_p^* groups of prime order p over elliptic curves, but in literature the Diffie-Hellman scheme is also considered on groups of composite integers n [2].
Let p be a prime number, then Z_p denotes the set of integers $\{0, 1, 2, \ldots, p-1\}$, where addition and multiplication are performed modulo p.

Definition: A *Field* is a non empty set F of elements with two operations "+" (called addition) and " ·" (called multiplication) satisfying the following axioms: for all $a, b, c \in F$,

i. F is closed under + and ·, i.e., $a + b$ and $a \cdot b$ are in F;
ii. Commutative laws: $a + b = b + a$, $a \cdot b = b \cdot a$;
iii. Associative laws: $(a + b) + c = a + (b + c)$, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;
iv. Distributive law: $a \cdot (b + c) = a \cdot b + a \cdot c$.

Furthermore, two distinct identity elements 0 and 1 (called the additive and multiplicative identities, respectively) must exist in F satisfying:

v. $a + 0 = a$ for all $a \in F$;
vi. $a \cdot 1 = a$ and $a \cdot 0 = 0$ for all $a \in F$;
vii. For any a in F, there exists an additive inverse element $(-a)$ in F such that $a + (-a) = 0$;
viii. For any $a \neq 0$ in F, there exists a multiplicative inverse element a^{-1} in F such that $a \cdot a^{-1} = 1$

Definition: A finite field of prime order p or prime power $q = p^f$ $(f >= 1)$ is commonly denoted F_q or $GF(q)$ (Galois field) and because Z_m is a field if and only if m is a prime, we denote the field Z_m by F_m. This is called a *prime field*.

Definition: For $n \geq 1$, let $\varphi(n)$ denote the number of integers in the interval $[1,n]$ which are relatively prime to n. The function φ is called the Euler phi function (or the *Euler totient function*)

Definition: Let $\alpha \in Z_p^*$ .If the order of α is $\varphi(n)$, then α is said to be a *generator* or a *primitive element* of Z_p^*. If Z_p^* has a generator, then Z_p^* is said to be *cyclic*.

Definition: An *Elliptic Curve* E consists of the set of points (X, Y, Z) that satisfy the following homogeneous Weierstrass equation:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

Where a_i $(i=1, 2, 3, 4, 5, 6)$ are elements of a field F and with the exception that the triple $(0, 0, 0)$ is not a point on E.

F can be set C (complex numbers), R (real) or Q (rational) or any other finite field F_q we want. The advantage of using R over E is that we can analyse arithmetic calculation in that field geometrically, this will help use to understand the arithmetic of elliptic curves and why they call them elliptic curves.
If we set $Z= 0$ and substitute $x = X/Z$, $y = Y/Z$ then we gets the equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

The above equation is called the affine Weierstrass equation. If a point *P* satisfy the homogeneous Weierstrass equation and the equation:

$$\frac{\partial F}{\partial X} = \frac{\partial F}{\partial Y} = \frac{\partial F}{\partial Z} = 0$$

Then we call that point singular and we call the Weierstrass equation also singular, note that singular Weierstrass equations are not of interest in the Cryptography.

We need now criteria that can help us to determine if a given affine Weierstrass equation singular is or not. The discreminante Δ (field element) is such a tool, which can be defined as follow:

$$\begin{aligned}
d_2 &= a_1^2 + 4a_2 \\
d_4 &= 2a_4 + a_1a_3 \\
d_6 &= a_3^2 \\
d_8 &= a_1^2a_5 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\
c_4 &= d_2^2 - 24d_4 \\
\Delta &= -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \\
j(E) &= c_4^3 / \Delta
\end{aligned}$$

If Δ = 0, then affine Weierstrass equation is singular, otherwise not singular [5] [6]. We call *j(E)* the *j-invariant* of the elliptic curve *E*. Note that only elliptic curves *E* over finite fields are of interest in cryptography.

Definition: The *characteristic* of a field *F*, often denoted *char(F)*, is the smallest positive number *n* such that:

$$\underbrace{1 + 1 + \ldots + 1}_{n} = 0$$

The field is said to have the characteristic zero if this repeated sum never reaches the additive identity.

The Arithmetic of $E(F_q)$ groups

Now we must differentiate between the characteristics of the underlying field *F* that we are using to build the group of points. The following table shows different finite fields and the corresponding elliptic curve equation classified by the characteristic.

Characteristic on *F*	**Elliptic curve equation**
1.Char(F) = 2, j(E) ≠ 0	$y^2 + xy = x^3 + a_2x^2 + a_6$
2.Char(F) = 2, j(E) = 0	$y^2 + a_3y = x^3 + a_4x + a_6$
3.Char(F) = 3, j(E) ≠ 0	$y^2 = x^3 + a_2x^2 + a_6$
4.Char(F) = 3, j(E) = 0	$y^2 = x^3 + a_4x + a_6$
5.Char(F) > 3	$y^2 = x^3 + a_4x + a_6$

Such an elliptic curve over a finite field *F* is a plane curve which consists of points that satisfy the elliptic curve equation *E* along with a distinguished point at infinity, denoted *O*. This set together with the "addition" rules as group operation form an *Abelian group*, with the identity element *O*.

Definition: The *discrete logarithm problem* over the elliptic curve *E* is the following: given two points *P* and *Q* in a group that satisfy E, find a number *x* such that $xP = Q$; *x* is called the discrete logarithm of *Q* to the base *P*.

Definition: The *base point C* is also referred to as the generator (often denoted |*E*|, #*E*, and $\#E(F_p)$ in the literature) or subgroup generator (Subgroup order is #*E*/*h*).

Definition: The Diffie-Hellman problem over the elliptic curve *E* is the following: given a base point *C*, and the two points $P_A = x_aC$ and $P_B = x_BC$ in a group that satisfy *E*, where x_A and x_B are unknown, compute the point:

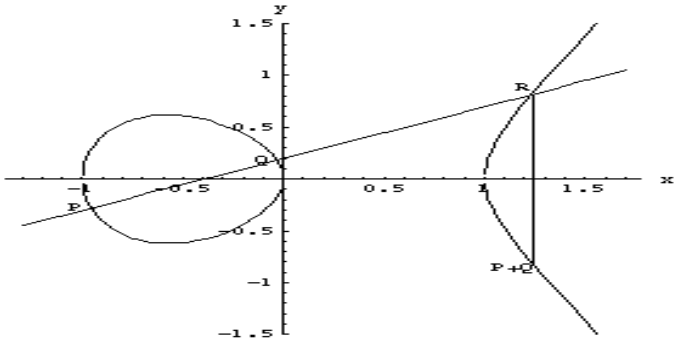
$$P_{AB} = x_Ax_BC$$

From the above definition we can easily conclude that if the discrete logarithm problem over elliptic curves is efficiently solved then the Diffie-Hellman problem over elliptic curves is broken because if we calculate x_B or x_B from publicly known P_A or P_B then we can easily compute the shared secret P_{AB}, for more information see [7]

The definition of group of points on elliptic curve *E*:

1. There is a point $O \in E$, such that for all $P \in E$, $P + O = O + P = P$.
2. $-O = O$ (the identity of the group).
3. If $P \neq O$ and $P=(x_1,y_1)$ then $-P$ is $(x_1,-y_1-a_1x_1-a_3)$.
4. If two points on *E* have same *x*-coordinate then either $P=Q$ or $P=-Q$.
5. If $Q = -P$, then $P + Q = O$.
6. For two points $P \neq O$ and $Q \neq O$ on *E*, the addition is defined as follows. Draw the line through *P* and *Q* to intersect the curve in a third point; then reflect that point in the *x*-axis.
7. For two points $P \neq O$ and $Q \neq O$ on *E*, when $P = Q$, use the tangent line at *P*. The identity of the group is *O*, the "point at infinity", which conceptually lies at the top and bottom of every vertical line.

The following sketch shows the addition of two points on the elliptic curve E:



For more information about elliptic curves in cryptography see [5] [6].

Before we encrypt a text message, we must decode that message into series of number that contains the message; here we could use some of the well known codes e.g. ASCII, Unicode.

The process of coding begins with substituting each character in the message with its corresponding numerical code z:

$$0 \leq z \leq a\text{-}1$$

Where *a* is the number of distinct characters in the selected code.

Next we have to select *s* such that

$$a^s < p < a^{s+1}$$

Now we divide our message in blocks of *s* length, and for each of these blocks we build the following sum:

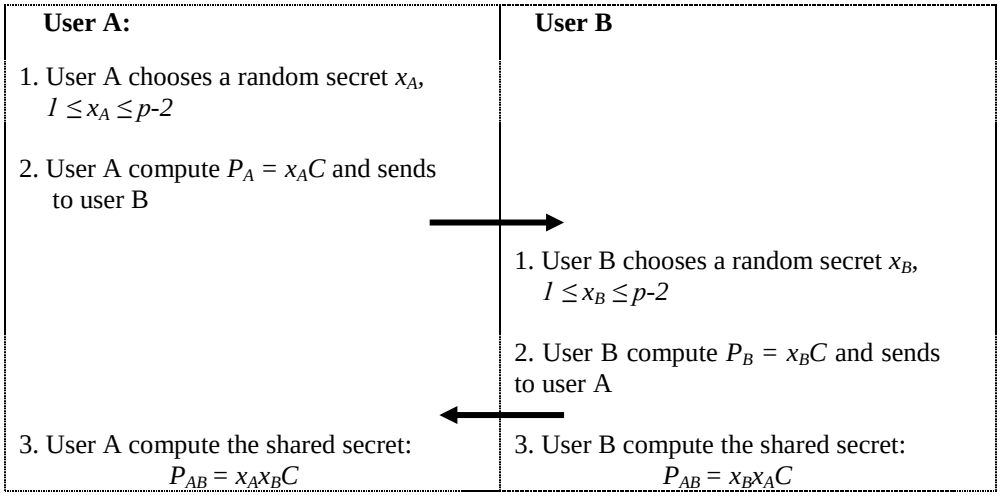
$$M = \sum_{i=0}^{z-1} z_i a^i$$

It is simple to check that $M < p$, we call *M* the coding block, it is also an element in the F_q.

Next we describe the basic version of the Diffie-Hellman key agreement protocol which is widely used in security protocols and which allows communicating parties that have never met before to establish a shared secret key over an open channel, this shared key can be used to encrypt data between the two communicating parties.

Protocol: Diffie-Hellman key agreement (basic version) over elliptic curves
Result: a shared secret $P_{AB} = x_A x_B C$ known to both communicating parties *A* and *B*.

1. One-time setup: a finite field F_q (where $q = p^f, f \geq 1$, p an appropriate big prime number) and a base point C over a given elliptic curve E are selected and published.
2. Protocol messages: Each time a shared key P_{AB} is required, A and B do the following:



Here we should also note that the Diffie-Hellman key agreement protocol provide the secrecy of the shared key because only the communicating parties knows x_A and x_B, thus only they can compute the shared secret key, on the other hand the problem rise that neither one of the communicating parties is assured of the identity of the other (man-in-the-middle attack), this problem can be solved if both parties have access to a trusted third party that issues certifications which binds their identity with the corresponding public key or if they use a public distribution of parameter over trusted channels. A list of attacks on Diffie-Hellman protocol can be found in [7].

An authenticated Diffie-Hellman key agreement protocol, also called Station-to–Station (STS) protocol, was described by Diffie, van Oorschot, and Wiener in 1992 [11] that provide protection against man-in-the-middle attack this is achieved by using digital signatures and public key certificates.

Algorithm: The Diffie-Hellman encryption/decryption scheme (basic version):
User B encrypts a message m for user A, which A decrypts.

1. Encryption. User B should do the following:
- (a) Obtain user A authentic public key $P_A = x_A C$.
- (b) Represent the message as an integer m in the interval $[0, p-2]$, m is a point on E.
- (c) Use private key x_B and compute shared secret $P_{AB} = x_B x_A C$
- (d) Compute $c = m + x_B x_A C$
- (e) Send the encrypted message c to user A.

2. Decryption. To recover plaintext m from c, user A should do the following:
- (a) Obtain user B authentic public key $P_B = x_B C$.
- (b) Use private key x_A and compute shared secret $P_{AB} = x_A x_B C$
- (c) Compute inverse element of $(P_{AB})^{-1} = - x_A x_B C$
- (d) Recover $m = c + (P_{AB})^{-1} = m + x_A x_B C - x_A x_B C$

The above described basic version of the Diffie-Hellman encryption, decryption and key agreement protocol does not contain any randomized parameter, thus a repeated message block will always lead to the same encryption block, and furthermore a chosen plain text attack can be performed.

The Diffie-Hellman problem has been studied for many years but still an efficient solution was not found thus it is considered as being difficult if the parameters are suitably chosen, but if the factors of $p-1$ are known or are small integers then the Diffie-Hellman problem can be easily solved using Integer factorization algorithms e.g. Index Calculus method [12], Pohlig-Hellman algorithm [13], etc., for more information see [7].

This implies that when generating Diffie-Hellman keys, it is important that the prime factors of $p-1$ should be selected in sufficient size such that factoring $p-1$ should be computationally infeasible.

3. THE SECURE DIFFIE-HELLMAN SCHEME OVER ELLIPTIC CURVES

The following protocol describes the modified Diffie-Hellman key agreement protocol over unsecure communication channel.

Protocol: Diffie-Hellman key agreement (secure version)
RESULT: a shared secret P_{AB} known to both communicating parties A and B.

1. One-time setup: a finite field F_q (where $q = p^f, f \geq 1$, p an appropriate big prime number) and a base point C over a given elliptic curve E are selected and published.
2. Protocol messages: Each time a shared key P_{AB} is required, A and B do the following:

User A	**User B**
1. User A chooses a random secret x_A, $1 \leq x_A \leq p-2$	
2. User A chooses a random integer k, $1 \leq k \leq p-2$	
3. User A compute $P_A = kx_AC$ and sends P_A to user B →	
	1. User B chooses a random secret x_B, $1 \leq x_B \leq p-2$
	2. User B compute $P_B = x_BC$ and sends to user A ←
4. User A compute the shared secret $P_{AB} = kx_Ax_BC$	3. User B compute the shared secret $P_{AB} = x_Bkx_AC$

The above modification of the Diffie-Hellman key agreement protocol ensure that the shared secret key P_{AB} will always look different even if the users A and B use the same public keys P_A, P_B, this is due to the randomized parameter k. Using this randomized parameter will provide more security.

For the below described examples we shall assume the following:

1. Arithmetic operation over F_q, where $char(F_q) > 3$

- Addition of two equal points ($P(x_1,y_1)+P(x_1,y_1) = 2P$): the (x_3, y_3) coordination of the $2P$ point are

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1$$

$$y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3)$$

- Addition of two distinct points ($P(x_1,y_1)+Q(x_2,y_2)=Z(x_3,y_3)$, where $P \neq Q$): the (x_3, y_3) coordination of the Z point are

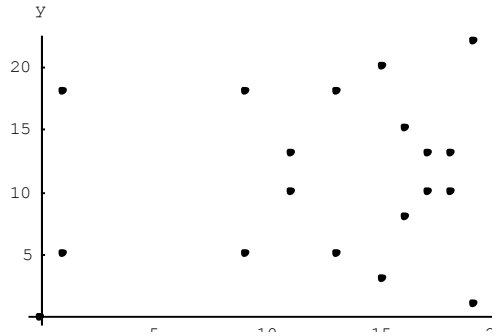
$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$

$$y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3)$$

2. We now consider an elliptic curve over the field F_{23}, where the elliptic curve equation $E: y^2 = x^3+ax+ b$, if we set $a = 1$ and $b = 0$, then we get the elliptic curve $E: y^2 = x^3 + x$. This equation must satisfy the equation $4a^3+27b^2 \neq 0 \ mod\ p$ to form a group, this is verified. The following 23 points over E that satisfies this equation are:

(0,0) (1,5) (1,18) (9,5) (9,18) (11,10) (11,13) (13,5) (13,18) (15,3) (15,20) (16,8) (16,15) (17,10) (17,13) (18,10) (18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17)

Using Mathematica, these points may be graphed as below:



This graph shows that E over F_q does not build a smooth elliptic function as E over R (Rational numbers).

Example: Diffie-Hellman key agreement (secure version)
RESULT: a shared secret P_{AB} known to both communicating parties A and B,
Assuming: C = (9, 5).

User A	**User B**
1. User A chooses a random secret $x_A=2$	
2. User A chooses a random integer $k=20$	
3. User A compute $P_A=kx_AC =$ $20*2\ C= 40(9,5) = (18,13)$ and sends P_A to user B →	
	1. User B chooses a random secret x_B,=5
	2. User B compute $P_B = x_BC = 5(9,5) =$ $= (9,18)$ and sends to user A ←
4. User A compute the shared secret $P_{AB} = kx_Ax_BC=20*2*(9,18)=$ $= (18,10)$	3. User B compute the shared secret $P_{AB} = x_Bkx_AC= 5*(18,13)=$ $= (18,10)$

The following algorithm describes the secure Diffie-Hellman cryptosystem.

Algorithm: The Diffie-Hellman cryptosystem (secure version):
User B encrypts a message m for user A, which A decrypts.

1. Encryption. User B should do the following:
- (1.1) Obtain user A authentic public key $P_A=x_AC$
- (1.2) Represent the message as a point m on the elliptic curve E
- (1.3) Choose a random integer k, where $1 \le k \le p-2$.
- (1.4) Use B private key x_B and Compute $P_{AB}=x_Bx_AC$
- (1.5) Compute $k*x_Bx_AC$
- (1.6) Compute $c = m + k * P_{AB} = m + k\, x_Bx_AC$
- (1.7) Represent k as a point on E, where k is the x-coordinate of that point
- (1.8) Compute $k+ x_Bx_AC$
- (1.9) Send the encrypted text message $(c, k+ x_Bx_AC)$ to user A.

2. Decryption. To recover plaintext m from c, user A should do the following:
- (2.1) Obtain user B authentic public key $P_B = x_BC$.
- (2.2) Use A private key x_A and compute $P_{AB} = x_Ax_BC$
- (2.3) Compute the inverse $(P_{AB})^{-1} = -x_Ax_BC$

(2.4) Recover *k*, compute $-x_Ax_BC + k + x_Bx_AC = k$
(2.5) Compute kx_Ax_BC
(2.6) Compute the inverse $-kx_Ax_BC$
(2.7) Recover message *m*, compute $-kx_Ax_BC + c = -kx_Ax_BC + m + kx_Bx_AC = m$

In the following example we assume that the message $m = 21$, and that one can use a simple coding algorithm to transform *m* into a point on the elliptic curve *E*, for simplicity we will set $m = (21, 6)$.

Example: Diffie-Hellman Cryptosystem (secure version)
Assuming: C = (9, 5), $P_{AB} = (18, 13)$.
User B encrypts a message m= (21, 6) for user A, which A decrypts.

1. Encryption. User B should do the following:
(1.1) Obtain user A authentic public key $P_A=x_AC=2(9, 5) = (18, 10)$
(1.2) Represent the message as a point *m* on the elliptic curve $m= (21, 6)$
(1.3) Choose a random integer *k*, where $1 \leq k \leq p-2$. $k = 20$
(1.4) Use B private key $x_B=5$ and Compute $P_{AB}=x_Bx_AC= 5(18, 10) = (18, 13)$
(1.5) Compute $k*x_Bx_AC = 20(18, 13) =(18,10)$
(1.6) Compute $c = m + k * P_{AB} = m + k\, x_Bx_AC = (21, 6)+(18, 10)= (19, 22)$
(1.7) Represent *k* as a point on *E*, where *k* is the *x*-coordinate of that point $(20, 4)$
(1.8) Compute $k+ x_Bx_AC = ((20,4)+(18,13))=(11,13)$
(1.9) Send the encrypted text message $(c, k+ x_Bx_AC) = ((19,22),(11,13))$ to user A.

2. Decryption. To recover plaintext *m* from *c*, user A should do the following:
(2.1) Obtain user B authentic public key $P_B = x_BC=5(9, 5) = (9, 18)$.
(2.2) Use A private key $x_A= 2$ and compute $P_{AB} = x_Ax_BC = 2(9, 18) = (18, 13)$
(2.3) Compute the inverse $(P_{AB})^{-1} = -x_Ax_BC = (18, 10)$
(2.4) Recover *k*, compute $-x_Ax_BC + k + x_Bx_AC = (18, 10)+(11,13)=(20,4)$
(2.5) Compute $kx_Ax_BC = 20(18, 13) = (18, 10)$
(2.6) Compute the inverse $-kx_Ax_BC = (18, 13)$
(2.7) Recover message *m*, compute $-kx_Ax_BC + c = -kx_Ax_BC + m + kx_Bx_AC =$
$(18, 13) + (19, 22) = (21, 6)$

Diffie-Hellman basic version is vulnerable against known plain-text attack; a known-plaintext attack is one where the adversary has a quantity of plaintext and corresponding ciphertext [7].

A known-plaintext attack is the following: given a sorted set $S = \{\{p_1,c_1\},\{p_2,c_2\},..., \{p_n,c_n\}\}$ (where $p_i \in P$ plaintext set, $c_i \in C$ ciphertext set, $n <= p$, p is the order of Z_p^*) an adversary can determine the plaintext p_x if the corresponding c_x is in S. But unlike the basic Diffie-Hellman where,

$$m = c + (P_{AB})^{-1}$$

The secure version of Diffie Hellman encryption algorithm requires the calculation of *k* in order to get the message *m* in the equation:

$$m = c + (k\, P_{AB})^{-1}$$

Important for Diffie-Hellman key agreement and encryption schemes regarding security consideration, is the size of the modulus *p* where *p-1* prime factors should be so selected sufficiently large such that factoring is computationally infeasible.

Here it is important to mention that the Diffie-Hellman problem over elliptic curve with small keys is much harder to solve than the discrete logarithm over finite fields, one of the reasons is that not all attacks on finite fields can be applied on elliptic curves, the following table shows cryptographic key length recommendation according to NIST (National Institute of Standards and Technology):

Date	**Finite fields**	**Elliptic curves**	**Security**
2007-2010	1024	160	short term
2011-2030	2048	224	middle term
>2030	3072	256	
>>2030	7680	384	long term
>>>2030	15360	512	

Some of the attacks of the Diffie-Hellman schemes are attack on the implementation [14] other powerful attacks on the Diffie-Hellman schemes are attacks on the integer factorization problem; e.g. the elliptic curve factoring algorithm [15], quadratic sieve [16] and number field sieve [17].
In 2010, the largest number factored by a general-purpose factoring algorithm was *768* bits long [18] [19] using distributed implementation thus some experts believe that *1024*-bit keys may become breakable in the near future so it is currently recommended to use *4096*-bit keys for long term security this requires about 270 bits using elliptic curve encryption.

4. CONCLUSION

In this paper I briefly discussed a secure version of the Diffie-Hellman key agreement and encryption decryption schemes, by using randomization to secure every shared secret key and every encrypted message block so that even if the same message is sent many times the encrypted message block will look different.
The major advantage gained here is that the security improvement described in this paper protects Diffie-Hellman schemes from a known plaintext attack, thus making the Diffie-Hellman schemes semantically secure, this is important because as mention in the introduction above, the Diffie-Hellman is implemented in many internet security standards and protocol and a weak Diffie-Hellman can make the whole system compromised. One solution that is used in praxis to overcome this problem is the use of padding bits in the generation process of the keys or in the encryption decryption process, but this may not always works if the adversary knows the padding bits.
It should also be mentioned that Diffie-Hellman over elliptic curve is also implemented on many small devices (e.g. smart card) where limited processing power and limited memory capacity exist, this is due to the small number of bits required to perform the encryption and decryption process, elliptic curves are considered newly in cryptography and is one of the most researched topic in cryptography.

5. REFERENCES

[1]. D. Kahn, The Code breakers: The comprehensive History of Secret Communication from Ancient to the Internet, Published 1967
[2]. W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, 22 (1976) 644-654.
[3]. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Commun. of the ACM, 21:120-126, 1978.
[4]. T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, volume 31, pages 469-472, 1985.
[5]. N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48:203-209, 1987
[6]. Menezes Alfred, "Elliptic Curve Public Key Cryptosystem", 1993, 4.eddition, 1997. Kluwer Academic Publishers.
[7]. A. Menezes, P. van Oorscot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7, 1999
[8]. Thayer, R.; Doraswamy, N.; Glenn, R. (November 1998). IP Security Document Roadmap. IETF. RFC 2411. http://tools.ietf.org/html/rfc2411
[9]. ANSI press specified ANSI X9.42 is a draft standard for the Diffie-Hellman Key Agreement Method
[10]. IEEE press, 2000. Specified the IEEE P1363 working group is developing standards for public-key cryptography based on RSA and Diffie-Hellman algorithm families and on elliptic curve systems
[11]. Diffie, W.; van Oorschot, P. C.; Wiener, M. J. (1992), "Authentication and Authenticated Key Exchanges", Designs, Codes and Cryptography (Kluwer Academic Publishers) 2 (2): 107–125
[12]. M. Kraitchik, *Théorie des nombres*, Gauthier--Villards, 1922
[13]. S. Pohlig and M. Hellman, "An improved algorithm for computing logarithm over GF(p) and its cryptographic significance", IEEE Transaction on Information Theory, volume 1462, Springer-Verlag, pages 458-471, 1998
[14]. Kocher, P., 1996. Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. Advances in Cryptology, 1109: 104-113.
[15]. B. Dixon, A.K. Lenstra, Massively parallel elliptic curve factoring, 183–193.
[16]. C. Pomerance, The quadratic sieve factoring algorithm, 169–182.
[17]. J. Buchmann, J. Loho, J. Zayer, An implementation of the general number field sieve, 159–165
[18]. RSA Laboratories, the RSA Factoring Challenge http://www.rsa.com/rsalabs/node.asp?id=2092
[19]. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (1997), 1484–1509. Available at http://www.research.att.com/shor.