Elliptic curve cryptography-Diffie Hellman technique extended for multiple two party keys at a time

Vankamamidi S. Naresh [1], Nistala V.E.S.Murthy[2] and V.V.Narasimha Rao [3]

[1] Department of Computer Science, S.V.K.P&Dr.K.S.Raju Arts&Science college
Penugonda,AndhraPradesh,534320,India

[2] Department of Computer Science & Systems Engineering, Andhra University
Visakhapatnam, Andhra Pradesh,530003,India

[3] Department of Computer Science & Engineering, J.N.T University Kakinada
Kakinada, Andhra Pradesh, 533003, India

Abstract

In recent years elliptic curve cryptography (ECC) is emerging as an alternative to traditional public key cryptosystems(DSA, RSA, AES, etc).ECC offers equivalent security with smaller key sizes resulting in faster computations, lower power consumption, as well as memory and bandwidth saving. This work presents an Extension of Elliptic curve Diffie Hellman technique to generate multiple shared keys at a time with reduced Key exchange operations (KEO), for increasing security and widening of applicability. A comparative study between proposed protocol and other crypto systems was made and satisfactory results have been obtained. Also an upper bound for the number of shared keys in terms of the number of exchanged keys and for a given number of shared keys, the minimum required number of keys to be exchanged.

Keywords: *Diffie Hellman, elliptic curves, multiple shared keys, discrete log problem.*

1. Introduction

In recent years some cryptographic algorithms have gained popularity due to properties that make them suitable for use in constrained environment like mobile information appliances, where computing resources and power availability are limited. One of these cryptosystems is Elliptic curve which requires less computational power, memory and communication bandwidth compared to other cryptosystem. This make the elliptic curve cryptography to gain wide acceptance as an alternative to conventional cryptosystems (DSA, RSA, AES, etc). Much smaller key lengths are required with ECC to provide a desired level of security, which means faster key exchange, user authentication, signature generation and verification, in addition to smaller key storage needs. The terms elliptic curve cipher and elliptic curve cryptography refers to an existing generic cryptosystem which use numbers generated from an elliptic curve. Empirical evidence suggests that cryptosystems that utilize number derived from elliptic curve can be more secure. As with all cryptosystems and especially with public-key crypto systems, it takes years of public evaluation before a reasonable level of confidence in a new system is established. ECC seem to have reached that level now. In the last couple of years, the first commercial implementations are appearing, as tool kits but also in real-world applications, such as email security, web security, smart cards, etc. The security of ECC has not been proven but it is based on the difficulty of computing elliptic curve discrete logarithm in the elliptic curve group

Because of the rapid increase of popularity of remote communication through unsecured channels such as the Internet, the use of cryptographic protocols to secure them increases.Even though there are many efficient cryptosystems currently available, their reliability depends on the keys being used, because the messages are easily interpreted when opponents know the secret values. Although this problem is overcome by changing the cryptographic keys frequently, the question is how it can be done through public communication channels. In fact, the ability to dynamically and publicly establish a session key for secured communication is a big challenge in cryptography.

Although public key algorithms prior to communication may be used for establishing the session keys, it requires additional key exchange (KEOs) per session and increases overhead. In such conditions, if multiple shared keys are exchanged securely at a time with comparatively fewer KEOs and if a key or even multiple keys are used in the same session, it not only eases the establishment of session keys, but also reduces the key exchange overhead significantly. The same is implemented in the Biswas work, where the generation of multiple two-party shared keys and a multiparty key is proposed But there are known sub exponential algorithms for breaking RSA, Diffie Hellman based on modular arithmetic ECC is important because the mathematicians do not (yet?) have sub exponential algorithms for breaking it, Therefore, it is believed to be secure with much smaller key size which is important for performance.

ECC is a candidate replacement for public key cryptographic schemes like DH, RSA, DSS etc as EC groups are advantages because they offer more security than other groups with smaller key sizes and faster computation times.

So combining the advantages of ECC and Biswas work[2].We proposed ECC crypto systems Elliptic curve multiple key Exchange protocol (ECMKEP) with the addition operation in proposed cryptosystem as a counter part of modular multiplication and multiple addition as a counter part of modular exponentiation in Biswas work [2]. In this present work instead of exchanging one public key as in ECC-DH, exchange m public keys between two parties A and B we generates $2^{m^2} - 1 - m^2$ shared keys in which m^2 keys are called base keys and these keys are used to generates 2^{m^2}-1-m^2 keys which are called as extend keys .

1.1 The main advantages of present work:

1. ECC-DH offers more security than modular arithmetic DH with smaller key sizes, less processing overheads and faster computation times.

2. The proposed technique reduces not only the computational cost significantly but also the key exchange over heads.

3. Depending on the application and the security needed, we can generate sufficient number of shared keys N, by selecting

$$m = \left[|\sqrt{(log\ (N+1)/\ log 2)}| \right] \qquad (1)$$

4. The advantage of elliptic curve over the other public key systems such as RSA, DSA etc is the key length. The following table [3] summarizes the key length of ECC based systems in comparison to other public key schemes.

Table 1: Comparison of the key lengths of RSA/DSA and ECC

RSA/DSA Key length	**ECC Key length for Equivalent Security**
1024	160
2048	224
3072	256
7680	384
15360	512

From the table it is very clear that elliptic curves offer a comparable amount of security offered by the other popular public key for a much smaller key length. This property of ECC has made the scheme quite popular of late.

2. Cryptography with elliptic curves:

2.1 Review of ECC DH Technique:

First pick a prime number$p \approx 2^{180}$and elliptic curve parameters a and b for equation

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

This defines elliptic group of $p \approx 2^{180}$ .Next pick a generator point $G = (x_1, y_1) \in E_p(a,b)$, whose order is a very large value n.

Step 1: A selects an integer$n_A < n$. This is A's private key and then A generates a public key

$$\mathrm{P_A} = \mathrm{n_A} \times \mathrm{G} \in E_p(a,b) \qquad (3)$$

Step 2: B selects an integer $\mathrm{n_B} < n$. This is B 's private key and then B generates a public key

$$\mathrm{P_B} = \mathrm{n_B} \times \mathrm{G} \in E_p(a,b) \qquad (4)$$

Step 3: A and B exchanges their public keys.

Step 4: A generates the secret key

$$\mathrm{K} = \mathrm{n_A} \times \mathrm{P_B} \qquad (5)$$

and B generates the secret key

$$\mathrm{K} = \mathrm{n_B} \times \mathrm{P_A}\ . \qquad (6)$$

The two calculations in step 4 produce the same result as follows.

$$\begin{aligned} n_A \times P_B &= n_A \times (n_B \times G) \\ &= n_B \times (n_A \times G) \\ &= n_B \times P_A \end{aligned} \qquad (7)$$

3. Proposed Protocol:

3.1 Elliptic curve multiple shared key Exchange Protocol (ECMKEP):

1. A selects m private keys $n_{A_i} < n$ for $1 \leq i \leq m$ and then generates m public keys

$$P_{A_i} = n_{A_i} \times G \in E_p(a,b) \text{ , for } 1 \leq i \leq m \qquad (8)$$

2. B selects m private keys $n_{B_i} < n$, for $1 \leq i \leq m$ and Then generates m public keys.

$$P_{B_i} = n_{B_i} \times G \in E_p(a,b)\text{,for } 1 \leq i \leq m \qquad (9)$$

3 .A and B exchanges their public keys.

4. A and B generates their " m^2" base keys as follows

$$K_{ij} = n_{A_i} \times P_{B_j} \text{ , for } 1 \leq i \leq m1 \leq j \leq m \qquad (10)$$

$$K_{ij}{}' = n_{B_j} \times P_{A_i} \text{ , for } 1 \leq i \leq m1 \leq j \leq m \qquad (11)$$

Clearly $K_{ij} = K_{ij}{}'$,these m^2 shared keys are called as base keys.

5. Additional shared keys can be derived by adding these base keys in different combinations which are called as extended keys

i. Adding two base keys at a time out of m^2 base keys generates $C(m^2,2)$shared keys such as

$$K=K_{11}+K_{12} \qquad (12)$$

ii .Adding Three base keys at a time out of m^2 base keys generates $C(m^2,3))$ shared keys such as

$$K = K_{11} + K_{12} + K_{13} \qquad (13)$$

....

……………………...

……………………...

iii. Adding m^2-1 base keys at a time out of m^2 base keys generates $C(m^2, m^2-1)$shared keys such as

$$K = K_{11} + K_{12} + \cdots + K_{1m} + \cdots + K_{m1} + K_{m2} + \cdots + K_{mm-1} \qquad (14)$$

Finally adding all m^2 base keys generates one shared key.

$$K = K_{11} + K_{12} + \cdots + K_{1m} + \cdots + K_{m1} + K_{m2} + \cdots + K_{mm} \qquad (15)$$

4 Security of proposed technique:

The elliptic curve cryptosystems which are based on the EC-DLP (Elliptic curve Discrett log problem) over a finite field have some advantage s over other systems, the key sizes can be much smaller than those in other schemes since only exponential time attacks have been known so far.If the curve is carefully chosen and EC-DLP might still intractable even if factoring and the multiplicative group discrete log turn out to be tractable problem. The EC-DLP is defined as follows.

4.1 EC-DLP (Elliptic curve Discrett log problem):

Def: Let E be an elliptic curve defined over a finite field F_P and let p ∈ $E(F_P)$,be a point of order n. Given Q ∈ $E(F_P)$ The Elliptic curve DLP is to find the integer l, $1 \leq l \leq n-1, \ni Q=l.P$

1.Proposition: m^2 two party shared keys $K_{11}, K_{12}, \dots, K_{1m} \dots K_{m1}, K_{m2}, \dots K_{mm}$ (base keys) derived in the application of the ECCDH technique are indistinguishable in polynomial time from random numbers.

Proof: Since each of the m^2 shared keys is basically a ECC-DH style key and for ECC-DH shared key the proposition is true, our proposition follows.

2. Corollary: The extended 2^{m^2}-1-m^2 shared keys generated by adding the m^2 base keys in different combinations are also indistinguishable in polynomial time from random numbers.

5. Comparison between ECMKDP and ECC-DH repeated m-times and Polynomial Time Complexity:

5.1 Comparative Analysis between the Proposed Protocol and Elliptic Curve Diffie Hellman (ECDH) Repeated N Times

First observe that

i. To generate $2^{m^2}-1$ shared keys in ECC D-H technique it requires $2^{m^2}-1$ rounds and proposed technique (ECMKEP) requires single round

ii. To generate $2^{m^2}-1$ shared keys it requires interchange of 2($2^{m^2}-1$) messages in ECCD-H technique and 2m messages in ECMKEP.

iii. To generate $2^{m^2}-1$ shared keys it requires $2^2(2^{m^2}-1)$ multiple addition operations in ECC D-H technique and ECMKEP requires $2m^2$+$2m$

iv. To generate $2^{m^2}-1$ shared keys it requires no additions and ECMKDP requires

$$2(2^{m^2-1}(m^2-2)+1)$$

v. The time complexity of D-H is

$$T_{ECCDH}(m) = C_e\, 2^2\, (2^{m^2}-1) = O(2^{m^2}) \qquad (16)$$

Where C_e denotes time needed for execution of one multiple addition operation.

So, D-H possibly has non polynomial (exponential) time complexity. Hence, it requires more time for execution.
Since multiplications are very less expensive than exponentiation, for time complexity we consider exponential operations and hence the time complexity of MSK

$$T_{ECMKDP}\,(m) = C_e(2m^2\text{+}2m) = O\,(m^2) \qquad (17)$$

where C_e denotes time needed for execution of one multiple addition operation. Thus, ECMKEP has polynomial (quadratic) time complexity. So it requires less time for execution than ECC D-H

Table2: Comparative Analysis

Protocol	**Number of Rounds**	**Interchange of number of messages**	**Execution of Number of multiple additions**	**Number of additions**	**Time complexity.**
EC-DH	$2^{m^2}-1$	$2(2^{m^2}-1)$	$2^2(2^{m^2}-1)$	NIL	$O\ (2^{m^2})$ (exponential)
ECMKEP	1	2m	m^2 +2m.	$\mathbf{2(2^{m^2-1}(m^2-2)+1}$	O (m^2) (quadratic)

Table3: Complexity Comparison for Various Protocols:

protocol	**complexity**
AKAP	3
SAKAP	1
MQVKC	2.5
MQV	2.5
ECDH	2
MTI/AO	3
MTI/CO	2
JOUX	2
SMART	4
ECMKDP Generalized algorithm, complexity decreases as m increases	0.8, for m=2 0.047 for m=3… $2m^2$+$2m/2^{m^2}-1$

6. Selection of shared key:

Now that the proposed extension can generate multiple shared keys, it is necessary for us to be able to select a key for a session. Here we suggest that one can follow a method similar to that in the well knownMerkle's puzzle which is recalled below:

A party generates n messages each with having a different puzzle number and a secret key number and sends all the messages to the other party in encrypted form. Note that a different 20 bit key is used for encryption of each message. The other party chooses one message at random and performs brute-force attack to decrypt it -although it needs a large amount of work, it is still computable. It then encrypts its message with the key thus recovered and sends it to the first party along with the puzzle number. Since it knows the puzzle number, it thus identifies the key and decrypts the message.
Similarly, in order to select a key out of $2^{m^2}-1$ shared keys, one party generates a message

comprising a shared key and a puzzle number. After encrypting it either with the smallest or largest key of the shared keys generated, it is sent to the other party. The message is easily decrypted as the recipient knows all keys and the (shared key, puzzle number) pair is recovered. The party then either sends the puzzle number alone or an encrypted message along with the puzzle number to the first party, where the message is encrypted with the shared key found. Since the first party knows the puzzle number, it therefore identifies the session key and can decrypt the message. For subsequent changes, the present session key may be used to encrypt a shared key at one end, and it is decrypted at the other end to obtain the new session key.

7. Conclusion:

In this paper we proposed a secure protocol ECMKEP based on DH using EC as public key cryptosystems using EC group works with smaller key sizes less processing overheads and provides more security in competitive to DH, DSS, RSA based on modular arithmetic and proved that our protocol is secured under EDLP, Further using the lower and upper bounds for m and N , the security levels can be increased in SDT with relatively lesser operations over heads.

References:

[1] M. Bellare, P. Rogaway, Entity Authentication and Key Distribution, Proceedings of CRYPTO'93, Santa Barbara, US

[2] G.P.Biswas, Diffie Hellman Technique Extended To Multiple Two Party Keys And One Multi Party Key, IET inf. Sec., 2008, Vol.2(1), pp.12-18.
A, 1994, 341-358.

[3] L. Law, A. Menezes, M. Qu, J. Solinas, & S. Vanstone, An efficient Protocol for Authenticated Key Agreement, Designs, Codes and Cryptography,
28 (2), 2003, 119-134.

[4] A. Menezes, M. Qu, S. Vanstone, Key Agreement and the need for authentication, Proceedings of PKS'95, Toronto, Canada, 1995.

[5] A. Menezes, M. Qu, S. Vanstone, Some new key agreement protocols providingmutual implicit authentication, Proceedings Workshop on Selected Areas in Cryptography (SAC'95), Nashville, USA, 1995, 22-32.
[6] M. Burmester, On the risk of opening distributed keys, Proceedings of CRYPTO'94, Santa Barbara, USA, 1994, 308-317.

[7] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography (Boca Raton, FL: CRC press, 1997).

[8] S. Blake-Wilson, D. Johnson, A. Menezes, Key Agreement Protocols and Their Security Analysis, Proceedings of Sixth IMA International Conference
on Cryptography and Coding, Cirencester, UK, 1997, 30-45.

[9] B. Kaliski, An unknown key-share attack on the MQV key agreement protocol, ACM Transaction on Information and Systems Security, 4 (3), 2001,275-288.

[10] N.P. Smart, An identity based authenticated key agreement protocol based on the Weil pairing, Electronic Letters, 38 (13), 2002, 630-632.

[11] F. Zhang, S. Liu, K. Kwangjo, ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings, Proceedings of IEEE International Symposium on Information Theory, Yokohama, Japan, 2003,136-148.

[12] C. Popescu, I. Mang, An Authenticated Key Agreement Protocol Based on the Weil Pairing, Proceedings of International Conference on Applied
Informatics (AI 2003), Innsbruck, Austria, 2003, 797-800.

[13] K. Shim, Efficient ID-based authenticated key agreement protocol based on Weil pairing, Electronic Letters, 39 (8), 2003, 653-654.

[14] A. Joux, A one-round protocol for tripartite Diffie-Hellman, Proceedings of Algorithmic Number Theory Symposium, Leiden, The Netherlands, 2000,385-394.

[15] S.S. Al-Riyami, K.G. Paterson, Tripartite authenticated key agreement protocols from pairings, Proceedings of IMA Conference on Cryptography and Coding, Cirencester, UK, 2003.

[16] L. Law, A. Menezes, M. Qu, J. Solinas, S. Vanstone, An efficient Protocol for Authenticated Key Agreement, Technical Report CORR98-05, Department of CO, University of Waterloo, 1998.

About the Authors:

Vankamamidi Srinivasa Naresh is currently working as a Director, for the Post Graduate Department of Computer Science Courses in **S.V.K.P. and Dr. K.S.R. Arts and Science College**. He obtained an M.Sc. in Mathematics from Andhra University, an M.Phil. in Mathematics from Madurai Kamaraj University and an M.Tech in Computer Science and Engineering from J.N.T. University-Kakinada. He is also a recipient of U.G.C.-C.S.I.R.JUNIOR RESEARCH FELLOSHIP and cleared NET for Lectureship

Nistala V.E.S. Murthy is currently working as a Professor in the department of Computer Science and Systems Engineering of Andhra University, Visakhapatnam. He developed f-Set Theory –wherein f-maps exists between fuzzy sets with truth values in *different* complete lattices, generalizing L-fuzzy set Theory of Goguen which generalized the [0,1]-fuzzy set theory of Zadeh, the Father of Fuzzy Set Theories. He also published papers on Representation of various Fuzzy Mathematical (Sub) structures in terms of their appropriate crisp cousins.

V.V.NarasimhaRao is currently pursuing his M.Tech computer science from J.N.T University Kakinada. He has presented number of technical papers in National Conferences. His research interests are Mobile Computing, Information Security and data mining.