# An Efficient Password Security of Multi-Party Key Exchange Protocol based on ECDLP

**Jayaprakash Kar** jayaprakashkar@yahoo.com
*Department of Information Technology*
*Al Musanna College of Technology*
*Sultanate of Oman*

**Banshidhar Majhi** bmajhi@nitrkl.ac.in
*Department of Computer Science & Engineering*
*National Institute of Technology*
*Rourkela, INDIA*

## Abstract

In this paper we have proposed an efficient password security of multiparty Key Exchange Protocol based on Elliptic Curve Discrete Logarithm Problem. Key exchange protocols allow a group of parties communicating over a public network to establish a common secret key called session key. Due to their significance by in building a secure communication channel, a number of key exchange protocols have been suggested over the years for a variety of settings. Our Protocol is password authentication model, where group member are assumed to hold an individual password rather than a common password. Here we have taken two one-way hash functions to build the level of security high.

**Keywords:** Key exchange protocol, Password based, secure communication, off-line dictionary attack, ECDLP.

## 1. Introduction

Group key exchange protocol is an important cryptographic technique in public network, by which a group shares a human-memorable password with a trusted server, can agree a secure session key. Over the past years, many group key exchange protocols have been proposed. However, to our best knowledge, not all of them can meet the requirements of security and efficiency simultaneously. Therefore, in this paper, we would like to propose a new simple multi-party password based authenticated key exchange protocol. Compared with other existing protocols, our proposed protocol does not require any server's public key, but can resist against various known attacks. Therefore, we believe it is suitable for some practical scenarios.

With the proliferation of the hand held wireless information appliances, the ability to perform security functions with limited computing resources has become increasingly important. In mobile devices such as personal digital assistants (PDAs) and multimedia cell phones, the processing resources, memory and power are all very limited, but he need for secure transmission of information may increase due to the vulnerability to attackers of the publicly accessible wireless transmission channel [1]. New smaller and faster security algorithms provide part of the solution, the elliptic curve cryptography ECC provide a faster alternative for public key cryptography. 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 [2]. As with all cryptosystems and especially with public-key cryptosystems, 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 toolkits 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 [3].

## 2. Backgrounds

In this section we brief overview of Elliptic Curve over finite field, Elliptic Curve Discrete Logarithm Problem, Key exchange, Elliptic Curve Diffe-Helman (ECDH) and about three-party key exchange protocol.

### 2.1 The finite field $F_P$

Let p be a prime number. The finite field $F_P$ is comprised of the set of integers $0,1,2.......p-1$ with the following arithmetic operations [5] [6] [7]:

1. Addition: If $a,b \in F_p$ then $a+b=r$, where $r$ is the remainder when $a+b$ is divided by $p$ and $0 \le r \le p-1$. This is known as addition modulo $p$.
2. Multiplication: If $a,b \in F_p$ then $a.b=s$, where $s$ is the remainder when $a.b$ is divided by $p$ and $0 \le s \le p-1.$. This is known as multiplication modulo $p$.
3. Inversion: If $a$ is a non-zero element in $F_P$, the inverse of $a$ modulo $p$, denoted $a^{-1}$, is the unique integer $c \in F_p$ for which $a.c=1$.

### 2.2 Elliptic Curve over $F_P$

Let $p \ge 3$ be a prime number. Let $a,b \in F_p$ be such that $4a^3+27b^2 \ne 0$ in $F_P$. An elliptic curve $E$ over $F_P$ defined by the parameters $a$ and $b$ is the set of all solutions $(x,y), x,y \in F_p$, to the equation $y^2=x^3+ax+b$, together with an extra point $O$, the point at infinity. The set of points $E(F_p)$ forms an abelian group with the following addition rules [9]:

1. Identity: $P+O=O+P=P$, for all $P \in E(F_p)$.
2. Negative : if $P(x,y) \in E(F_p)$ then $(x,y)+(x,-y)=O$, The point $(x,-y)$ is dented as $-P$ called negative of $P$.

3. Point addition: Let $P(x_1, y_1), Q(x_2, y_2) \in E(F_p)$, then $P + Q = R \in E(F_p)$ and coordinate $(x_{3,} y_3)$ of $R$ is given by $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$ where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$
4. Point doubling : Let $P(x_1, y_1) \in E(F_p)$ where $P \neq -P$ then $2P = (x_3, y_3)$ where $x_3 = (\frac{3x_1^{\ 2} + a}{2y_1})^2 - 2x_1$ and $y_3 = (\frac{3x_1^{\ 2} + a}{2y_1})(x_1 - x_3) - y_1$

### 2.3 Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given an elliptic curve E defined over a finite field $F_p$ ,a point $P \in E(F_p)$ of order n, and a point $Q \in (P)$, find the integer $l \in [0, n-1]$ such that $Q = l.P$. The integer $l$ is called discrete logarithm of $Q$ to base $P$, denoted $l = \log_p Q$ [9].

### 2.4 Key exchange

Key exchange protocols allow two parties to agree on a secret shared secret key that they can use to do further encryption for a long message. One of these protocols is the Diffie-Hellman, which is the most used one. The Elliptic curve Diffie- Helman is considered as an extension to the standard Diffie- Hellman.

### 2.5 Elliptic Curve Diffie-Helman

Elliptic curve Diffie-Helman protocol (ECDH) is one of the key exchange protocols used to establishes a shared key between two parties. ECDH protocol is based on the additive elliptic curve group. ECDH begin by selecting the underlying field $F_p$ or $GF(2^k)$, the curve $E$ with parameters $a, b$ and the base point $P$ . The order of the base point $P$ is equal to $n$. The standards often suggest that we select an elliptic curve with prime order and therefore any element of the group would be selected and their order will be the prime number $n$ [5]. At the end of the protocol, the communicating parties end up with the same value $K$ which is a point on the curve.

## Key exchange

Key exchange protocols allow two parties to agree on a secret shared secret key that they can use to do further encryption for a long message. One of these protocols is the Diffie-Hellman, which is the most used one. The Elliptic curve Diffie-Helman is considered as an extension to the standard Diffie- Hellman. Another direction of research on key agreement is to generalize the two party key agreements to multi party setting.

## Group Key Exchange Protocol

Consider the dynamic scenario where participants may join or leave a multicast group at any given time. As a result of the increased popularity of group oriented applications, the design of an efficient authenticated group key agreement protocol has recently received much attention in the literature. A comprehensive treatment have been made to extend the two party (and three party) key agreement protocols to multi party setting. Notable solutions have been suggested by Ingemerson et al. [13], Burmester and Desmedt [10], Steiner et al. [12] and Becker and Willie [11]. All these works assume a passive (eavesdropping) adversary, and the last three provide rigorous proofs of security. For practical applications, efficiency is a critical concern in designing group key agreement in addition to provable security. In particular, number of rounds may be crucial in an environment where numbers of group members are quite large and the group is dynamic. Handling dynamic membership changes get much attention to the current research community. A group key agreement scheme in a dynamic group must ensure that the session key is updated upon every membership change so that subsequent communication sessions are protected from leaving members and previous communication sessions are protected from joining members. Although this can be achieved by running any authenticated group key agreement protocol from scratch whenever group membership changes, alternative approaches to handle this dynamic membership more effectively would be clearly preferable in order to minimize cost of the re-keying operations associated with group updates. The problems of key agreement in Dynamic Peer Groups (DPG) were studied by Steiner et al. [12]. They proposed a class of generic n-party Diffie-Hellman protocols". Atenise et al. [14] [15] introduced authentication into the class of protocols and heuristically analyze their security against active adversary. Steiner et al. [16] consider a number of different scenarios of group membership changes and introduced a complete key management suite CLIQUES studied specially for DPGs which enable addition and exclusion of group members as well as refreshing of the keys. The security analyses of these schemes are heuristic against active adversaries. However, Pereira and Quisquater [20] have described a number of potential attacks, highlighting the need for ways to obtain greater assurance in the security of these protocols. Bresson et al. [17] [18] have recently given a formal security model for group authenticated key agreement. They provided the first provably secure protocols based on the protocols of Steiner et al. [12] for this setting which requires O(n) rounds to establish a key among a group of n users. The initial works [18] [?] respectively consider the static and dynamic case, the security of both of which are in random oracle model following the formalized security model introduced by themselves under the computational Diffie-Hellman (CDH) assumption. They further refine in [18] the existing security model to incorporate major missing details, (e.g. strong corruption and concurrent sessions) and proposed an authenticated dynamic group Diffie-Hellman key agreement proven secure under the DDH assumption within this model. Their security result holds in the standard model instead of random oracle model.

## 3. Proposed Protocol

Our protocol is designed for use of multi-cast network. The protocol participants consists of a single authenticated server $S$ and multi clients $C_1, C_2 .......C_m$ who wish to establish a session key. All clients have registered their respective password $pw_1, pw_2 ....pw_m$ . Then the multiparty protocol runs among all the clients with the following parameters established:

- Let the elliptic curve $E$ defined over a finite field $F_P$ two field elements $a, b \in F_p$, which defined the equation of the elliptic curve $E$ over $F_P$ i.e. $y^2 = x^3 + ax + b$ in the case $p \geq 3$, where $4a^3 + 27b^2 \neq 0$.
- Let $M_1, M_2 ......M_m$ be $m$ number of group elements in $E(F_p)$.
- Two one-way hash functions $G$ and $H$ , where the output are the elements of $F_P$
- Iteration Count is the number to be randomly choosed and both the hash function will be executed that numbers of times. Let the number be $c \in [1, n-1]$ $[20]$. So we have to compute both the hash $G$ and $H$ for $c$ no of times.

The proposed protocol follows the follows the following steps.

- **Step -I**: Let each client $C_i$ for $i = 1.2.....m$ selects random numbers $t_i \in [1, n-1]$ and computes the point $P_i = t_i.Q$ and $P_i' = P_i + pw_i.M_i$ and broad cast $P_i'$ to rest of the group.
- **Step -II**: Clients send $(C_1 \| P_1') \| (C_2 \| P_2')......(C_m \| P'_m)$ to $S$ .
- **Step-III**: Upon receiving, $S$ first recovers $P_i$ by computing $P_i = P_i' - pw_i.M_i$ . Next $S$ and $R$ by computing $P = P' - M.pw_A$ and $R = R' - N.pw_B$ . Next $S$ select random number $u$ from $[1, n-1]$ and computes $\tilde{P}_i = u.P_i$ for all $i = 1,2.....m$ and then compute the following

$$pw_i'(1) = pw_i.G(C_i \| S \| P_i) \text{ for all } i = 1.2.....m$$

$$pw_i'(2) = G(pw_i'(1))$$

$$...$$

$$pw_i'(c) = G(pw_i'(c-1))$$

Finally we get $pw_i' = G(pw_i'(c))$

Then computes $\tilde{P}'_i = pw_j'.P_i', j = 1,2......m$ and $i \neq j$ and sends $\tilde{P}'_1 \| \tilde{P}'_2 \| .....\tilde{P}'_m$ to rest of the group.

- **Step -IV** : After having received $\tilde{P}'_1 \| \tilde{P}'_2 \| .....\tilde{P}'_m$ , $C_i$ computes the pair wise key as $K_j = t_j.p\tilde{w}'_j{}^{-1}.(\tilde{P}'_i)$ , where $i, j = 1,2.......m$ and $i \neq j$

$\alpha_1 = G(C_1 \| C_2 \| ...... \| C_m \| K)$, where $K = K_i = K_j$ for $i, j = 1,2......m$ and $i \neq j$ .

$$\alpha_2 = G(\alpha_1)$$

$$\alpha_3 = G(\alpha_2)$$

$$\vdots$$

$$\alpha = \alpha_c .$$

.

Client $C_j$ sends $\tilde{P}'_i \| \alpha$ to $C_i$ for $i, j = 1,2.....m$ and $i \neq j$ .

- **Step-V:** With $\tilde{P}'_i \| \alpha$ from $C_j, C_i$ computes $pw'_i = pw_i.G(C_i \| S \| P_i)$, $K_i = t_i.(pw'_i)^{-1}.\tilde{P}'_j$ and verifies that $\alpha = \alpha_c$ by computing $\alpha_1, \alpha_2 .......\alpha_c$ and $\alpha_1 = G(C_1 \| C_2 \| ...... \| C_m \| K)$ if the verification fails, then $C_i$ aborts the protocol. Otherwise $C_i$ computes the session key $SK$ as

$$SK(1) = H(C_1 \| C_2 \| ........ \| C_m \| K)$$

$$SK(2) = H(SK(1))$$
$$\vdots$$
$$SK(c) = H(SK(c-1))$$
$$SK = SK(c)$$

and sends $\beta = \beta_c$, where $\beta_1 = G(C_1 \| C_2 \| ........C_m \| K)$ and $G(\beta_{c-1}) = \beta_c$

- **Step-VI**: Each client $C_i$ verifies the correctness of $\beta$ is equal to $\beta_c$ by checking the equation $\beta_1 = G(C_1 \| C_2 \| ....... \| C_m \| K)$, $\beta_2 = G(\beta_1) \ldots \beta_c = G(\beta_{c-1})$. If it holds, then each client $C_i$ computes the session key $SK = H(C_1 \| C_2 \| \ldots \| C_m \| \ldots \| K)$, otherwise, $C_i$ abort the protocol.

### 3.1 Verification of Correctness of 3PAKE

The correctness of the protocol can be verified for each client $C_1, C_2 \cdots C_m$ . Let for the client $C_1$, the key $K_1 = \tilde{P}'_2.(pw_1')^{-1}.t_1$ can be verified with the client $C_2$ having the key $K_2 = P_1'.(pw_2')^{-1}.t_2$ by computing as

$$K_1 = \tilde{P}_2'.(pw_1')^{-1}.t_1 = (pw_1')^{-1}.(pw_1').\tilde{P}_2.t_1 = u.P_2.t_1 = u.t_1.t_2.Q$$
$$K_2 = \tilde{P}_1'.(pw_2')^{-1}.t_2 = (pw_2')^{-1}.(pw_2').\tilde{P}_1.t_2 = u.P_1.t_2 = u.t_2.t_1.Q$$

Similarly for each client $C_3, C_4....C_m$ the correctness of the protocol can be verified.

## 4. Security discussions

**Theorem-1:** The protocol does not leak any information that allows verifying the correctness of password guesses.

Proof: Since $G$ is a one-way hash function is executed $c$ times and $s, u$ and $t$ are all random numbers, so the protocol does not leak any information that allow the adversary to verify the correctness of password guesses.

**Theorem-2:** The protocol is secure against off-line password guessing attacks.

Proof: If the hacker intends to tract out the password, first he has to find out the iteration count $c$ which is a random number and process that number of times. Further he has to solve Elliptic Curve Discrete Logarithm problem (ECDLP) which is computationally infeasible takes fully exponential time. So we can say it is secured against off-line password guessing attacks.

## 5. Off-Line Dictionary Attack

The proposed protocol is secure against off-line dictionary attacks. This does not leak any information that allows to verify the correctness of password guesses, because $G$ is a one-way function and $s, u$ and $t$ all are random numbers to be taken from $[1, n-1]$. Further the vulnerability of the protocol to the off-line attack can be avoided as

- Consider for the client $C_i$, let $\overline{pw_i} = G(pw_i)$ and $\overline{pw_j} = G(pw_j)$ for $i \neq j$ and for all $i, j = 1,2 \cdots m$. Then $C_i$ computes $P' = P + \overline{pw_i}.M$ in stead of $P' = P + pw_i.M$ , and $C_j$ compute as $R' = R + \overline{pw_j}.N$ instead of as $R' = R + pw_j.N$ .
- Accordingly, the Server S recovers $P$ and $R$ is modified to $P = P' - \overline{pw_i}.M$ and $R = R' - \overline{pw_j}.N$ .

## 5. Conclusion

In this research a new protocol for exchanging key between a numbers of parties with a trusted Server has been defined. This new protocol has two major advantages over all previous key exchange protocol, first this protocol does not leak any information that allow the adversary to verify the correctness of password guesses. The second one is that this protocol does not leak any information that allows verifying the correctness of password guesses. The proposed protocol is also easy to implement. The security of our system is based on Elliptic Curve Discrete Logarithm Problem (ECDLP). The primary reason for the attractiveness of ECC over systems such as RSA and DSA is that the best algorithm known for solving the underlying mathematical problem (namely, the ECDLP) takes fully exponential time. In contrast, sub-exponential time algorithms are known for underlying mathematical problems on which RSA and DSA are based, namely the integer factorization (IFP) and the discrete logarithm (DLP) problems. This means that the algorithms for solving the ECDLP become infeasible much more rapidly as the problem size increases than those algorithms for the IFP and DLP. For this reason, ECC offers security equivalent to RSA and DSA while using far smaller key sizes. he attractiveness of ECC will increase relative to other public-key cryptosystems as computing power improvements force a general increase in the key size. The benefits of this higher-strength per-bit include higher speeds, lower power consumption, bandwidth savings, storage efficiencies, and smaller certificates.

## References

1. Murat Fiskiran A and B Ruby Lee *“Workload characterization of elliptic curve cryptography and other network security algorithms for constrained environments*”. Proc. IEEE Intl.Workshop on Workload Characterization, pp:127-137, 2002.

2. De Win E. and B Preneel “*Elliptic curve public-key cryptosystems - an introduction.State of the Art in Applied Cryptography*”, LNCS 1528, pp: 131-141, 1998.

3. Aydos M., E Savas and C .K .KoV 1999. “*Implementing network security protocols based on elliptic curve cryptography*”. Proc. fourth Symposium. Computer Networks, pp: 130-139, 1999.

4. Y.F. Chang “*A Practical Three-party Key Exchange Protocol with Round Efficiency”*. International Journal of Innovative Computing, Information and Control,Vol.4, No.4, April 2008, 953960.

5. N. Koblitz. “*A course in Number Theory and Cryptography*”, 2nd edition Springer-Verlag-1994.

6. K. H Rosen ”*Elementary Number Theory in Science and Communication*”,2nd ed., Springer-Verlag, Berlin, 1986.

7. A. Menezes, P. C Van Oorschot and S. A Vanstone “*Handbook of applied cryptography”*. CRC Press, 1997.

8. D. Hankerson, A .Menezes and S.Vanstone. “*Guide to Elliptic Curve Cryptography* “Springer Verlag, 2004.

9. ”*Certicom ECC Challenge and The Elliptic Curve Cryptosystem*” available: http://www.certicom.com/index.php.

10. M. Burmester and Y. Desmedt ”*A Secure and Efficient Conference Key Distribution System*”. In proceedings of Eurocrypt 1994, LNCS 950, pp. 275-286, Springer-Verlag, 1995.

11. K. Becker and U.Wille ”*Communication Complexity of Group Key Distribution*”. In proceedings of ACM CCS 1998, pp. 1-6, ACM Press, 1998.

12. M. Steiner, G. Tsudik, M. Waidner ”*Diffie-Hellman Key Distribution Extended to Group Communication*”. In proceedings of ACM CCS 1996, pp.31-37, ACM Press, 1996.

13. I. Ingemarsson, D. T. Tang, and C. K. Wong ”*A Conference Key Distribution System*”. In IEEE Transactions on Information Theory 28(5), pp. 714-720, 1982.

14. G. Ateniese, M. Steiner, and G. Tsudik ”*Authenticated Group Key Agreement and Friends*”. In proceedings of ACM CCS 1998[1], pp. 17-26, ACM Press, 1998.

15. G. Ateniese, M. Steiner, and G. Tsudik “*New Multi-party Authenticated Services and Key Agreement Protocols”*. In Journal of Selected Areas in Communications, 18(4), pp. 1-13, IEEE, 2000.

16. M. Steiner, G. Tsudik and M.Waidner ” *Cliques : A New Approach to GroupKey Agreement*” In IEEE Conference on Distributed Computing Systems, May 1998, pp. 380.

17. E. Bresson, O. Chevassut, and D. Pointcheval ” *Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions*”. In proceedings of Eurocrypt 2002, LNCS 2332, pp. 321-336, Springer-Verlag, 2002.

18. E. Bresson, O. Chevassut, and D. Pointcheval. “*Provably Authenticated Group Diffie-Hellman ”Key Exchange - The Dynamic Case*”. In proceedings of Asiacrypt 2001, LNCS 2248, pp. 290-309, Springer-Verlag, 2001.

19. E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater ”*Provably Authenticated Group Diffie-Hellman Key Exchange*”. In proceedings of ACM CCS 2001, pp. 255-264, ACM Press, 2001.

20. Matthew N. Anyanwu, Lih-Yuan Deng and Dipankar Dasgupta "*Design of Cryptographically Strong Generator By Transforming Linearly Generated Sequences"* . In International Journal of Computer Science and Security, pp 186-200, Vol-3, issue-3

21. O. Pereira and J.J. Quisquater"*A Security Analysis of the Cliques Protocol Suite*". In Computer Security Foundations Workshop (CSFW 2001), pp. 73-81, IEEE Computer Society Press, 2001.

22. M. Steiner, G. Tsudik and M.Waidner "Cliques : "*A New Approach to Group Key Agreement*". In IEEE Conference on Distributed Computing Systems, May 1998, pp. 380.