A Secure Two-Party Identity-Based Key Exchange Protocol Based on Elliptic Curve Discrete Logarithm Problem

Jayaprakash Kar [1] and Banshidhar Majhi [2]

[1]Department of Information Technology
Al Musanna College of Technology, Sultanate of Oman
jayaprakashkar@yahoo.com

[2]Department of Computer Science and Engineering
National Institute of Technology, Rourkela, India
bmajhi@nitrkl.ac.in

Abstract

This paper presents a secure identity-based key exchange protocol whose security is based on Elliptic Curve Discrete Logarithm Problem. The attractiveness of Elliptic Curve Discrete Logarithm Problem is that the best algorithm known for solving the underlying mathematical problem takes fully exponential time. In contrast, sub-exponential time algorithms are known for underlying other mathematical problems namely the integer factorization (IFP) and the discrete logarithm (DLP). For this reason, ECC offers security equivalent to RSA and DSA while using far smaller key sizes. The 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. So as compared with the previously proposed protocols, it has better performance. Key exchange protocols allow two parties communicating over a public network to establish a common secret key called session key to encrypt the communication data. 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.The proposed key exchange protocol provides implicit key authentication as well as the desired security attributes of an authenticated key exchange protocol.

Keywords : authentication, identity-based, key exchange, ECDLP, security.

1 Introduction

Key-exchange protocols are among the most basic and widely used cryptographic protocols. Such protocols are used to derive a common session key between two (or more) parties; this session key may then be used to communicate securely over an insecure public network. Thus, secure key-exchange protocols serve as basic building blocks for constructing secure, complex, higher-level protocols. For this reason, the computational/communication efficiency and round complexity of key-exchange protocols are very important and have received much attention, both in the two-party and multi-party (i.e., group) [40] [29] [39] [33] [28] [26] [34] settings.

A key establishment protocol allows principals to establish a common key for encrypting their communications over an insecure network. A two-party key exchange (or agreement)protocol is used to establish a common session key for two specified entities, in which both two entities contribute some information to derive the shared session key. If three or more participants want to communicate securely over an insecure network,they may employ a conference-key establishment protocol to compute a conference key [17], Ingemaresson et al., 1982; [22] [23]. [31] first proposed a secure key exchange protocol. However, it does not allow two entities to authenticate each other, so their protocol requires an authentication channel to exchange the public keys. According to technical categories of authentication approach, key exchange protocols may be classified into a number of categories: public-key-based key exchange protocols. A public-key based key exchange protocol adopts public-key cryptographic techniques to achieve the purposes of user authentication and key exchange. On the way of key management, although the public-key-based key exchange protocol is better than password-based key exchange protocol. However, on-line access to get and verify public keys from a public key system in a network system is time-consuming. Moreover, it needs to require extra efforts to maintain public-keys in a public key system . On the other hand,an identity-based key exchange protocol can be regard as a variation of the public-key based key exchange protocol. An identity-based key exchange protocol is a protocol that uses userŠs identity or some other information combined with his identity as oneŠs public key to achieve user authentication and key exchange. Thus, a verifier does not verify the certificates of the public keys. Mean while, no on-line system authority is required.

One common assumption is that each communicating party has an associated public private-key pair, with the public key

Received January 29, 2010

known to all other parties in the network (of course, this includes the adversary). We assume this model here.Most protocols for two-party key exchange have been designed and analyzed assuming that parties alternate sending messages (equivalently, that the parties communicate over a bidirectional half-duplex channel). However, in many common scenarios parties can actually transmit messages simultaneously (i.e., they have access to a bidirectional duplex channel). Of course, any key-exchange protocol designed and proven secure in the former model will also be secure in the latter model; however, it may be possible to design protocols with improved round complexity by fully exploiting the communication characteristics of the underlying network, and in particular the possibility of simultaneous message transmission. As a simple example, consider the traditional Diffie-Hellman key-exchange protocol [31]which does not provide any authentication. However, the situation is more complex when authentication is required. For instance, authenticated Diffie-Hellman key exchange typically involves one party signing messages sent by the other party; this may be viewed as a type of challenge-response mechanism. (For example, the work of Bellare, et al. [24]suggests implementing authenticated channels in exactly this way.) When this is done, it is no longer possible to collapse the protocol to a single round. Motivated by the above discussion, we explore the possibility of designing protocols for authenticated key exchange which can be implemented in only a single round (assuming simultaneous message transmission). Of course, we will also ensure that our protocols are efficient with respect to other measures, including communication complexity and computational efficiency.

Over the past years, many two-party authenticated key exchange protocols have been proposed. However, to our best knowledge, not all of them can meet the requirements of security and efficiency simultaneously.

The proposed key exchange protocol provides implicit key authentication as well as the desired security properties of an authenticated key exchange protocol.The remainder of this article is organized as follows.

The organization of the paper is in Section 3 we review briefly about key exchange protocols and section 4 about two-party key exchange protocol.Section 5 describes security goals and attributes, section 6 our new propose identity-based key exchange protocol. The security analysis of the new protocol is presented in Section 7. In Section 8, the performance analysis. Section 8 gives our conclusions and finally we have described about further research work.

2 Background

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

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 $0,1,2,\ldots p-1$ with the following arithmetic operations [1] [2] [3]

- 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.
- 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.
- 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 a abelian group with the following addition rules [4]:

1. Identity : $P + \mathscr{O} = \mathscr{O} + \mathscr{P} = \mathscr{P}$, for all $P \in E(F_p)$
2. Negative : if $P(x,y) \in E(F_p)$ then $(x,y) + (x,-y) = \mathscr{O}$, 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 $P(x_1,y_1) \in E(K)$ where $P \ne -P$ 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 $Q = lP$. The integer l is called discrete logarithm of Q to base P,denoted $l = log_p Q$ [4].

2.4 The Diffe-Hellman problems

Let $\mathscr{GG}$ be an algorithm which on input 1^k outputs a (description of a) group G of prime order q (with $|q| = \ k$) along with a generator $g \in G$. The computational Diffie-Hellman (CDH) problem is the following: given g^{u_1}, g^{u_2} for random $u_1, u_2 \in Z_q^\star$, compute $g^{u_1 u_2}$. We say that $\mathscr{GG}$ satisfies the CDH assumption if this problem is infeasible for all PPT algorithms. More formally, for any PPT algorithm $\mathscr{A}$

consider the following experiment:

$\mathbf{Exp}^{cdh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)}$

1. $(G,q,g) \leftarrow \mathscr{G}\mathscr{G}(1^k)$
2. $u_1,u_2 \leftarrow Z_q$
3. $U_1 = g^{u_1}; U_2 = g^{u_2}$
4. if $W = g^{u_1u_2}$ return 1 else return 0

The advantage of an adversary $\mathscr{A}$ is defined as follows:

$$\mathbf{Adv}^{cdh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)} = Pr[\mathbf{Exp}^{cdh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)} = 1]$$

We say that $\mathscr{G}\mathscr{G}$ satisfies the CDH assumption if $\mathbf{Adv}^{cdh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)}$ is is negligible for all ppt algorithms $\mathscr{A}$. When we are interested in a concrete security analysis, we drop the dependence on k and say that $\mathscr{G}\mathscr{G}$ is (t,ε) -secure with respect to the CDH problem if $\mathbf{Adv}^{cdh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)} \leq \varepsilon$ for all $\mathscr{A}$ running in time at most t. (We will sometimes be informal and say that a group G output by $\mathscr{G}\mathscr{G}$ satisfies the CDH assumption).

Letting $\mathscr{G}\mathscr{G}$ be as above, we may define a DDH tuple to be a tuple of the form $(g,g^{u_1},g^{u_2},g^{u_1u_2})$ and a random tuple to be a tuple of the form $(g,g^{u_1},g^{u_2},g^{u_3})$. The decisional Diffie-Hellman assumption is to distinguish a random DDH tuple from a random tuple. We say that $\mathscr{G}\mathscr{G}$ satisfies the DDH assumption if this problem is infeasible for all PPT algorithms. More formally, for any PPT algorithm $\mathscr{A}$ consider the following experiment:

$\mathbf{Exp}^{ddh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)}$

1. $(G,q,g) \leftarrow \mathscr{G}\mathscr{G}(1^k)$
2. $u_1,u_2 \leftarrow Z_q$
3. $U_1 = g^{u_1}; U_2 = g^{u_2}$
4. $V_0 = g^{u_1u_2}; V_1 \leftarrow G$
5. $b \leftarrow \{0,1\}$
6. $b' \leftarrow \mathscr{A}(G,q,g,U_1,V_b)$
7. if $b' = b$ return 1 else return 0

The advantage of an adversary $\mathscr{A}$ is defined as follows:

We say that $\mathscr{G}\mathscr{G}$ satisfies the DDH assumption if $\mathbf{Exp}^{ddh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)}$ is negligible for all PPT algorithms $\mathscr{A}$.When we are interested in a concrete security analysis, we drop the dependence on k and say that $\mathscr{G}\mathscr{G}$ is (t,ε)-secure with respect to the DDH problem if $\mathbf{Exp}^{ddh}_{\mathscr{A},\mathscr{G}\mathscr{G}(k)} \leq \varepsilon$ for all $\mathscr{A}$ running in time at most t. (We will sometimes be informal and say that a group G output by $\mathscr{G}\mathscr{G}$ satisfies the DDH assumption.)

2.5 Elliptic Curve Diffie-Hellman Problem

Elliptic curve Diffie-Helman Problem 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 [46]. At the end of the protocol,the communicating parties end up with the same value K which is a point on the curve.

3 Key Exchange protocol

The key agreement problem is stated as follows: two entities wish to agree on keying information in secret over a distributed network. Since the seminal paper of Diffe and Hellman in 1976 [31], solutions to the key agreement problem whose security is based on the Diffe-Hellman problem in finite groups have been used extensively. Suppose now that entity i wishes to agree on secret keying information with entity j. Each party desires an assurance that no party other than i and j can possibly compute the keying information agreed. This is the authenticated key agreement (AK) problem. Clearly this problem is harder than the key agreement problem in which i does not care who (or what) he is agreeing on a key with, for in this problem i stipulates that the key be shared with j and no-one else. Several techniques related to the Diffe-Hellman problem have been proposed to solve the AK problem . However, no practical solutions have been provably demonstrated to achieve this goal, and this deficiency has lead in many cases to the use of awed protocols . The flaws have, on occasion, taken years to discover; at best, such protocols must be employed with the fear that a flaw will later be uncovered. Since in the AK problem, i merely desires that only j can possibly compute the key, and not that j has actually computed the key, solutions are often said to provide implicit (key) authentication. If i wants to make sure in addition that j really has computed the agreed key, then key confirmation is incorporated into the key agreement protocol, leading to so-called explicit authentication. The resulting goal is called authenticated key agreement with key confirmation (AKC). The incorporation of entity authentication into the AK protocol provides i the additional assurance that j can compute the key, rather than the (slightly) stronger assurance that j has actually computed the key. Practical solutions that employ asymmetric techniques to solve the AK and AKC problems are clearly of fundamental importance to the success of secure distributed computing. The motivation for this paper stems in part from the recent successes of the ‘random oracle model’ in providing practical, provably good asymmetric schemes, and in part from the desire of various standards’ bodies (in particular IEEE P1363 to lift asymmetric techniques in widespread use above the unsuccessful ‘attack-response’ design methodology. The goal of this paper is to make strides towards the provision of practical solutions for the AK and AKC problems which are provably good. Firstly by providing clear, formal definitions of the goals of AK and AKC protocols, and secondly by furnishing practical, provably secure solutions in the random oracle model. The model of distributed computing adopted appears particularly powerful, and the definitions of security chosen particularly strong. The approach we take closely follows the approach of [41],where provable security is provided for entity authentication and authenticated key transport using symmetric techniques. Also relevant is the adaptation of techniques from to the asymmetric setting found in . Roughly speaking, the process of proving security comes in five stages:

- specification of model;
- definition of goals within this model;
- statement of assumptions;
- description of protocol;
- proof that the protocol meets its goals within the model.

We believe that the goals of AK and AKC currently lack formal definition. It is one of our central objectives to provide such definitions. We particularly wish to stress the important roles that appropriate assumptions, an appropriate model, and an appropriate definition of protocol security play in results of provable security|all protocols are provably secure in some model, under some definitions, or under some assumptions. Thus we believe that the emphasis in such work should be how appropriate the assumptions, definitions, and model which admit provable security are, rather than the mere statement that such-and-such a protocol attains provable security. It is a central thesis of this work, therefore, that the model of distributed computing we describe models the environment in which solutions to the AK and AKC problems are required, and that the definitions given for the AK and AKC problems are the 'right' ones.

3.1 Properties of Key Exchange Protocol

There is a vast literature on key agreement protocols [2]. Unlike other primitives, such as encryption or digital signatures, it is not clear what constitutes an attack on a key agreement protocol. A number of distinct types of attacks have been proposed against previous schemes, as well as a number of less serious weaknesses. Therefore, before we can begin to analyze any protocol, it is necessary to identify what attacks a protocol should withstand, and what attributes are desirable for a protocol to have. First we identify two types of attack:

1. **Passive attacks**: Here an adversary attempts to prevent a protocol from achieving its goal by merely observing honest entities carrying out the protocol;

2. **Active attacks**: Here an adversary additionally subverts the communications them- selves in any way possible: by injecting messages, intercepting messages, replaying messages, altering messages, and the like.

Clearly it is essential for any secure protocol to withstand both passive and active attacks, since an adversary can reasonably be assumed to have these capabilities in a distributed network.

4 Two-Party Key Exchange Protocol

Numerous Diffie-Hellman based authenticated key agreement protocol and authenticated key agreement with key confirmation protocols have been designed to add authentication (and key confirmation) to the Diffie-Hellman protocol; however,many have subsequently been found to have flaws. One of the well-known authenticated key agreement (AK) protocol in the Diffie-Hellman family is MTI protocol by Matsumoto, Takashima and Imai [36]. They designed three infinite families of key agreement protocols to provide implicit key authentication in the classical Diffie-Hellman key agreement protocol. However, the security analysis against active adversary is only heuristic. Law et al pointed out flaws in the protocols and presented an efficient authenticated key agreement protocol, often called MQV protocol. The security analysis of MQV protocol against active adversary is also heuristic. Both MTI and MQV family of protocols are certificate-based. There are many ID-based key agreement protocols based on pairing. Scott [7] proposed an ID based key agreement protocol where each user selects his own personal identity number (PIN) and a trusted PKG issues each user an individual secret associated with the identity of corresponding user. A value is calculated from both the individual secret and PIN number and placed inside a hardware token. The individual secret can be reconstructed from their memorized PIN number,identity and token.Another ID-based authenticated key agreement was proposed by Smart [20] that combines the idea of Boneh and Franklin [9] with the tripartite Diffie-Hellman protocol of Joux [11]. The scheme uses weil pairing and requires all users involved in the key agreement to be clients of the same PKG. The protocol allows efficient ID-based escrow facility for sessions that enables low enforcement agencies to decrypt messages encrypted with the session keys, after having obtained the necessary warrants. Chen and Kudla [43] developed an ID-based authenticated key agreement protocol more efficient than Smart's protocol [20]. They have suggested a mechanism to turn escrow off which can also be applied to Smart's protocol [20] (the escrow-free environment may be desirable for personal communications the users wish to keep confidential even from the PKG). They also provided a modification that allows key agreement between users under different PKGs.None of the two party key agreement protocols by Scott [7], Smart [20] and Chen and Kudla [43] were broken, although heuristic arguments are adopted to prove their security against active adversary. Shim [19] presented an ID-based key agreement protocol. However, Sun and Heish [8] showed that Shim's key agreement protocol is insecure against the man-in-the-middle attack. Another efficient ID-based authenticated key agreement protocol was proposed by McCullagh and Barreto [15] that can be used in either escrow or escrow-free mode. They also developed a scheme for key agreement between clients of different PKGs. The scheme is twice as efficient as the scheme in [43] without pre computation. Later, Xie [10] pointed out a flaw in it and removed this flaw by suggesting modifications for the protocol. Recently, Choo [12] showed that both the scheme and its modified variant are not secure if the adversary is allowed to reveal non-partner players who had accepted the same session key. Jeong et al. [13]proposed three simple single-round two-party key agreement protocols with detail security analysis in the security model of [14]. A practical two party-key exchange protocol comply with the following requirements.

1. The session key should be agreed by the communication parties instead of being assigned by the server directly.
2. Except the password, no extra secret information should be needed - the public key for example.
3. Computation and round efficiencies should be provided at the same time.

4.1 Security Model for Authenticated Key Exchange

We use the standard notion of security for key-exchange protocols as defined in , taking into account forward secrecy following [42].We assume that there are N parties, and each party's identity is denoted as P_i. Each party P_i holds a pair of private and public keys, where the public key is assumed to be known to all other parties in the network (and the adversary, too). We consider key-exchange protocols in which two parties want to exchange a session key using their public keys to provide authentication. $\prod_i^k$, represents the k-th instance of player P_i, and we assume a given instance is used only once. If a key-exchange protocol terminates, then $\prod_i^k$ generates a session key sk_i^k . A session identifier of an instance, denoted sid_i^k , is a string different from those of all other sessions in the system (with high probability).

Consider instance $\prod_i^k$ of player P_i. The *partner* of this instance is the player with whom Pi believes it is interacting. We say that two instances $\prod_i^k$ and $\prod_i^{k'}$ are *partnered* if $\prod_i^k = \prod_i^{k'}$, P_j is the partner of $\prod_i^k$, and P_i is the partner of $\prod_i^{k'}$. Any protocol should satisfy the following correctness condition: two partnered instances (of uncorrupted parties) compute the same session key. To define security, we define the capabilities of an adversary. We allow the adversary to potentially control all communication in the network via access to a set of oracles as defined below. We consider an experiment in which the adversary asks queries to oracles, and the oracles answer back to the adversary. Oracle queries model attacks which an adversary may use in the real system. We consider the following types of queries in this paper, specialized for the case of 2-round protocols.

- The query **Initiate**(P_i, k, P_j) is used to "prompt" the unused instance $\prod_i^k$ of party P_i to initiate execution of the protocol with partner $P_j \neq P_i$. This query will result in P_i sending a message, which is given to the adversary.
- A query **Send**(P_i, k, M) is used to send a message M to instance $\prod_i^k$; this models active attacks on the part of the adversary. We assume without loss of generality that an adversary always queries **Initiate**$(P_i, k, \star)$ before querying Send (P_i, k, M); this corresponds to assuming that the adversary always "rushes" the messages of honest parties, which only gives the adversary more power.
- A query **Execute**(P_i, P_j) represents passive eavesdropping of the adversary on an execution of the protocol by parties P_i and P_j (with $P_i \neq P_j$). In response to this query, parties P_i and P_j execute the protocol without any interference from the adversary, and the adversary is given the resulting transcript of the execution.(Although the actions of the Execute query can be simulated via repeated Initiate and Send oracle queries, this particular query is used to distinguish between passive and active attacks.)
- A query **Reveal**(P_i, k) models known key attacks (or Denning-Sacco attacks) in the real system. In response to this query, the adversary is given the session key sk_i^k for the specified instance.
- A query **Corrupt**(P_i) models exposure of the long-term key held by player P_i. The adversary is assumed to be able to obtain long-term keys of players, but cannot control the behavior of these players directly (of course, once the adversary has asked a query **Corrupt**(P_i), the adversary may impersonate P_i in subsequent **Send**(queries)
- A query **Test**(Pi, k) is used to define the advantage of an adversary. In response to this query, a coin b is flipped. If b is 1, then the session key sk_i^k is returned. Otherwise, a random session key (i.e., one chosen uniformly from the space of session keys) is returned. The adversary may make a single test query to a fresh instance at any time during the experiment.

5 Security Goals and Attributes

In the past, some desired security goals and attributes have been identified for an authenticated key exchange protocol [16] . In general, the importance of providing these security goals and attributes is dependent on the applications. In the following, we first describe two kinds of fundamental security goals. An authenticated key exchange protocol should provide one of two kinds of security goals.

- Implicit key authentication. It means that each principal only shows the other principal,who can compute the session key.
- Explicit key authentication. It means that a principal is assured that another principal have actually computed the session key.

Although it is important to provide formal security proof on any cryptographic protocols,key exchange protocols remain one of the most challenging research issues. Until now, a provably secure two-pass authenticated key exchange protocol is still an important subject of research [18]. The notion of provable security makes several concrete security attributes to be presented as desirable.Several desirable security attributes have been presented in the past literatures. We summary these attributes as follows [21] a detail discussions):

1. Known-key security: In each run of a key exchange protocol, two specified entities should produce a unique session key. When an adversary has learned some other session key produced by previous runs, the adversary is unable to learn some other session key between the two entities.

2. Full forward secrecy: It means that if oneŠs long-term private key is disclosed to some adversaries, they can not learn the previous session key. So this security goal makes the secrecy of previous session key not affected, even if the long-term private key loss. A further distinction is that a single entityŠs private key is compromised or the private keys of both participating entity are compromised. The former is called half forward secrecy, and the latter is called full forward secrecy.

3. Key-compromise impersonation. Assume that entities A and B are two principals. Suppose AŠs secret key is disclosed. Obviously, an adversary who knows this secret key can impersonate A to other entities. However, it is desired in some situation that this disclosure does not allow the adversary to impersonate other entities to A.

4. Unknown key-share: When entities B believes the key is shared with some entity $C \neq A$, and A believes the key is shared with B. The above scenario can not be permitted. This scenario was first described in (Diffie et al., 1992).

6 Proposed Identity-Based Key Exchange Protocol

Let A and B be two legal clients in the system who wish to establish a session key, and S be a trusted authentication server which chooses the system parameters and generating key pair for each user. In the setup phase, the authority chooses 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 $4a^3 + 27b^2 \neq 0$. Then, the authority possess a one-way hash function $\mathscr{H}$. Let d is the number to be randomly choose from the interval $[1, n-1]$, computes the point $Q = d \cdot P$, where P and Q are group element in $E(F_p)$. The key pair (d, Q), in which the private key d and Q is a public key, and publishes P, Q and $\mathscr{H}$. For each user, the authority computes $I = \mathscr{H}(ID)$, where ID is the identity string that may include the name, e-mail address, birthday or physical description corresponding to the user's identity. Then, the authority chooses a random number k from the interval $[1, n-1]$ and computes $R = k \cdot P$ as user's Public key and $s = k + d \cdot \mathscr{H}(ID)$ as the user's Private key. That is, each legal user i with the identity information ID_i has a key pair (R_i, s_i). Assumed that the users A and B are two legal users in the system. Thus, A and B have the key pairs $R_A = k_A \cdot P, s_A = k_A + d \cdot \mathscr{H}(ID_A)$ and $R_B = k_B \cdot P, s_B = k_B + d \cdot \mathscr{H}(ID_B)$ respectively. Thus, A and B carry out the following steps to generate the session key shared between them.

1. Step-I **(round 1)**: A generates a random integer $t_A \in Z_q^\star$ and computes $U_A = t_A \cdot P$. Then, A uses her private key s_A to compute $v_A = t_A + s_A \cdot U_{A_x}$, where U_{A_x} is x-coordinate of point U_A and sends U_A, R_A and ID_A to B.

2. Step-II **(round 2)**:B also generates a random integer $t_B \in Z_q^\star$ and computes $U_B = t_B \cdot P$ and then B use his private key s_B and to compute $v_B = t_B + s_B \cdot U_{B_x}$, and sends U_B, R_B and ID_B to A.

The detailed of the two rounds have been illustrated in the following table

Client A	Client B
Select random number $t_A \in Z_q^\star$ Compute $U_A = t_A \cdot P$ Compute $v_A = t_A + s_A \cdot U_{A_x}$ Where U_{A_x} is the x-coordinate of the point U_A	Select random number $t_B \in Z_q^\star$ Compute $U_B = t_B \cdot P$ Compute $v_B = t_B + s_B \cdot U_{B_x}$ Where U_{B_x} is the x-coordinate of the point U_B
$\xrightarrow{(U_A, R_A, ID_A)}$	
	Computation of Session key $Z_B = R_A + \mathscr{H}(ID_A) \cdot Q$ $= k_A \cdot P + \mathscr{H}(ID_A) d \cdot P$ $= s_A \cdot P$ $K_B = v_B \cdot (U_A + U_{A_x} \cdot Z_B)$ $= v_B \cdot (U_A + U_{A_x} \cdot s_A \cdot P)$ $= v_B \cdot (t_A \cdot P + U_{A_x} \cdot s_A \cdot P)$ $= v_B \cdot (t_A + U_{A_x} \cdot s_A) \cdot P)$ $= (v_B \cdot v_A) \cdot P$
$\xleftarrow{(U_B, R_B, ID_B)}$	
Computation of Session key $Z_A = R_B + \mathscr{H}(ID_B) \cdot Q$ $= k_B \cdot P + \mathscr{H}(ID_B) \cdot d \cdot P$ $= (k_B + d \cdot \mathscr{H}(ID_B)) \cdot P$ $= s_B \cdot P$ $K_A = v_A \cdot (U_B + U_{B_x} \cdot Z_A)$ $= v_A \cdot (U_B + U_{B_x} \cdot s_B \cdot P)$ $= v_A \cdot (t_B \cdot P + U_{B_x} \cdot s_B \cdot P)$ $= v_A \cdot (t_B + U_{B_x} \cdot s_B) \cdot P)$ $= (v_A \cdot v_B) \cdot P$	

6.1 Key Computation

To compute the the session key K_A, A will follows the following steps.

1. $Z_A = R_B + \mathscr{H}(ID_B) \cdot Q = k_B \cdot P + \mathscr{H}(ID_B) \cdot d \cdot P$
   $= (k_B + d \cdot \mathscr{H}(ID_B)) \cdot P = s_B \cdot P$

2. $K_A = v_A \cdot (U_B + U_{B_x} \cdot Z_A)$
   $= v_A \cdot (U_B + U_{B_x} \cdot s_B \cdot P)$
   $= v_A \cdot (t_B \cdot P + U_{B_x} \cdot s_B \cdot P)$
   $= v_A \cdot (t_B + U_{B_x} \cdot s_B) \cdot P)$
   $= (v_A \cdot v_B) \cdot P$

B also computes the session key K_B as follows

1. $Z_B = R_A + \mathscr{H}(ID_A) \cdot Q = k_A \cdot P + \mathscr{H}(ID_A) \cdot d \cdot P$
   $= (k_A + d \cdot \mathscr{H}(ID_A)) \cdot P = s_A \cdot P$

2. $K_B = v_B \cdot (U_A + U_{A_x} \cdot Z_B)$
   $= v_B \cdot (U_A + U_{A_x} \cdot s_A \cdot P)$
   $= v_B \cdot (t_A \cdot P + U_{A_x} \cdot s_A \cdot P)$
   $= v_B \cdot (t_A + U_{A_x} \cdot s_A) \cdot P)$
   $= (v_B \cdot v_A) \cdot P$

It is clear that A and B have the common session key $K = K_A = K_B = (v_A \cdot v_B) \cdot P$

7 Security Analysis

Here, let us discuss the security of the proposed protocol. The security of the proposed protocol is based on the difficulty of computing the elliptic curve discrete logarithm problem [2] and the DiffieŰHellman scheme [31].

- Firstly, we show that if an adversary eavesdrops the transmitted messages U_A, R_A, ID_A, U_B, R_B and ID_B between two entities, he is unable to obtain the secret key s_A of the user A from R_A and ID_A, or the secret key s_B of the user B from R_B and ID_B. Since $s_A = k_A + d \cdot \mathscr{H}(ID_A)$ has two unknown variable variables k_A and d selected by the system authority, and the adversary wants to obtain two unknown variables from the transmitted messages, he must compute k_A and d from $R_A = k_A \cdot P$ and $Q = d \cdot P$. Thus, it is equivalent to solving the elliptic curve discrete logarithm problem. In the proposed protocol, the adversary may find $Z_A = R_B + \mathscr{H}(ID_B) \cdot Q = s_B \cdot P$. If the adversary tries to find s_B from $R_B + \mathscr{H}(ID_B = s_B \cdot P$, he still faces the difficulty of elliptic curve solving the discrete logarithm problem.
- Considering another situation, if an adversary eavesdrops the transmitted messages U_A, R_A, ID_A, U_B, R_B and ID_B between two entities, he is still unable to obtain the established common session key. For computing the established common session key $K_A = v_A \cdot (U_B + U_{B_x} \cdot Z_A)$ or $K_B = v_B \cdot (U_A + U_{A_x} \cdot Z_B)$, the adversary must know v_A or v_B. However, both v_A and v_B are not transmitted in the proposed protocol. Thus, the adversary is also unable to compute v_A or v_B because $v_A = t_A + s_A \cdot U_{A_x}$ and $v_B = t_B + s_B \cdot U_{B_x}$ contain the usersŠ secret keys s_A and s_B, respectively.
- In the following, let us consider that any legal user i with a key pair (R_i, s_i) is unable to compute the secret key d of the system authority. In fact, the key pair $(R_i = k_i \cdot P, s_i = k_i + d \cdot \mathscr{H}(ID_i))$ may be viewed as a SchnorrŠs signature (Schnorr, 1990) generated by the system authority for the identity information ID_i. Pointcheval and Stern (1996) have shown that to compute the secret key d from (R_i, s_i) is equal to the difficulty of solving the DiffieŰHellman problem.

In fact, a provably secure two-pass authenticated key exchange protocol is still an important subject of research (Kaliski, 2001). Fortunately, the notion of provable security 132 Y.-M. Tseng makes several concrete security attributes to be identified as desirable. In the following, let us discuss that the new proposed protocol satisfies the desirable security attributes described in Section (Security Goal and Attribute).

1. Known-key security. If the session key K is disclosed, the protocol may withstand known-key attack. Suppose that the adversary has known a pre-session key K_1 established between A and B. Since $K_1 = v_{A_1} \cdot v_{B_1} \cdot P$ we have $K_1 = (t_{A_1} + s_A \cdot U_{A_{1x}}) \cdot (t_{B_1} + s_B \cdot U_{B_{1x}}) \cdot P = (t_{A_1} \cdot t_{B_1}) \cdot P + (s_A \cdot U_{A_{1x}} \cdot t_{B_1}) \cdot P + (t_{A_1} \cdot s_B \cdot U_{B_{1x}}) \cdot P + (s_A \cdot U_{A_{1x}} \cdot s_B \cdot U_{B_{1x}}) \cdot P$

   Suppose that there is another value K_2 established between A and B now. As the same reason, we have $K_2 = (t_{A_2} + s_A \cdot U_{A_{2x}}) \cdot (t_{B_2} + s_B \cdot U_{B_{2x}}) \cdot P$. First, because K_1 is the multiplicative addition of four items $(t_{A_1} \cdot t_{B_1}) \cdot P, (s_A \cdot U_{A_{1x}} \cdot t_{B_1}) \cdot P, (t_{A_1} \cdot s_B \cdot U_{B_{1x}}) \cdot P$ and $(s_A \cdot U_{A_{1x}} \cdot s_B \cdot U_{B_{1x}}) \cdot P$ and each itemŠs multiplication consists of two unknown values, thus the adversary is unable to obtain the valid information such as, (s_A, s_B) from K_1. Certainly, he/she does not find another session key K_2 from K_1. Therefore, the proposed protocol can withstand known-key attack.

2. Full forward secrecy. If both secret keys of A and B are disclosed, the adversary tries to compute v_A or v_B, and then to compute $K = (v_A \cdot v_B \cdot) \cdot P$. However, to find v_A or v_B must require to know t_A or t_B from U_A or U_B, respectively. Thus, this will be equivalent to solving the elliptic curve discrete logarithm problem. Moreover, because of the session key K includes the value of $(t_A \cdot t_B) \cdot P$, which is still unknown to the adversary. Therefore, the proposed protocol can provide full forward secrecy.

3. Key-compromise impersonation. Suppose that the secret key of B is disclosed. An adversary who knows this secret key tries to impersonate some entity A to B. Because of it is necessary to compute v_A for impersonating A, and it must be computed using the secret key s_A of A. In such case, impersonating A to B is impossible. Therefore, the proposed protocol can withstand key-compromise impersonation attack.

4. Unknown key-share. The kind of attack has a precondition, which is that the public key of the adversary must determine by oneself. Obviously, since the userŠs public key is determined by the authority, it can withstand unknown key-share attack (Kaliski, 2001).

Finally, let us consider the security goal about key authentication. Suppose that there are two honest entities A and B, who want to execute the proposed key exchange protocol to establish a common session key. Since $K = (v_A \cdot v_B) \cdot P$, other entities must know either s_A or s_B to compute v_A or v_B for computing the session key . That is, no other entities can learn the session key. Thus, the new key exchange protocol provides implicit key authentication between A and B.

8 Performance Analysis

For convenience, the following notations are used to analyze the computational cost. T_{mul} is the time for sclar multiplication; T_{add} is the time for addition; T_H is the time of executing the one way hash function $\mathscr{H}()$; As for the computational cost in our proposed protocol, any user i of two entities must compute U_i, v_i, Z_i, and K. It requires $5T_{mul} + T_{add} + T_H$ for each entity.

9 Conclusion

An identity-based key exchange protocol has an advantage, that to avoid the on-line access of obtaining the public keys in a network environment, because of the verification of the public key in an identity-based system is embedded in the key establishing process between two entities. An efficient identity-based key exchange protocol based on the difficulty of computing the elliptic discrete logarithm problem has been proposed. The proposed key exchange protocol provides implicit key authentication, and it provides the desired security attributes of an authenticated key exchange protocol. As compared with the previously proposed protocols, it reduces the computational cost. In this research a new protocol for exchanging key between two 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 to verify 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.The attractiveness of ECC will increase relative to other public-key cryptosystem 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.

10 Further research

The proposed Protocol can also be designed using Pairing technique. Pairing has recently had a number of positive applications in cryptography, for instances, identity-based encryption [44], identity-based signatures [45], key agreement and short signatures .

Let $\mathbf{G}_1$ denotes a cyclic additive group generated by an element P, whose order is a prime q, and $\mathbf{G}_2$ denotes a cyclic multiplicative group of the same prime order q.

A pairing is a computable bilinear map between these two groups. Two pairings have been studied for cryptographic use. They are Weil pairing and a modified version ref to [44], and Tate pairing. The protocol in pairing setting based on Bilinear Diffe-Hallman Problem (BDH) and Computational Diffe-Hellman (CDH) Problem. The security depands upon the insolvability of these problems. The following section describes about the two problems.

10.1 Bilinear and Computational Diffe-Hellman Problem

we let $\hat{e}$ denote a general bilinear map, i.e. $\hat{e} : G_1 X G_1 \longrightarrow G_2$, which can be either a modified Weil pairing or a Tate pairing. A Diffie-Hellman (DH) tuple in G_1 is $(P, xP, yP, zP) \in G_1$ for some $x, y, z \in Z_q$ satisfying $z = xy \bmod q$.

- Computational Diffie-Hellman (CDH) problem :- Given any three elements from the four elements in a DH tuple compute the remaining element. CDH assumption: There exists no algorithm running in expected polynomial time, which can solve the CDH problem with non-negligible probability.

- Decision Diffie-Hellman (DDH) problem:- Given $P, xP, yP, zP \in G_1$, decide if it is a valid DH tuple. This can be solved in polynomial time by verifying $\hat{e}(xP, yP) = \hat{e}(P, zP)$.

- Bilinear Diffie-Hellman (BDH) problem: Let P be a generator of G_1. The BDH problem in . $< G_1, G_2, \hat{e} >$. is that given (P, xP, yP, zP) for some $x, y, z \in Z_q$, compute $W = \hat{e}(P,P)^{xyz} \in G_2$.
  BDH assumption: There exists no algorithm running in expected polynomial time, which can solve the BDH problem in $< G_1, G_2, \hat{e} >$ with non-negligible probability.

References

[1] K. H Rosen *"Elementary Number Theory in Science and Communication", 2nd ed., Springer-Verlag, Berlin, 1986.*

[2] A. Menezes, P. C Van Oorschot and S. A Vanstone *Handbook of applied cryptography. CRC Press, 1997.*

[3] D. Hankerson, A .Menezes and S.Vanstone. *Guide to Elliptic Curve Cryptography, Springer Verlag, 2004.*

[4] *"Certicom ECC Challenge and The Elliptic Curve Cryptosystem" available :http://www.certicom.com/index.php.*

[5] T. Matsumoto, Y. Takashima and H. Imai *" On Seeking Smart Public-key Distribution Systems". In Transactions of the IECE of Japan, E69, pp. 99-106, 1986.*

[6] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. *"An Efficient Protocol for Authenticated Key Agreement". Technical Report CORR 98-05,Department of C & O, University.of Waterloo, 1998. Also available at http://citeseer.nj.nec.com/law 98efficient.*

[7] M. Scott. *"Authenticated ID-based Key Exchange and Remote Log-in with Insecure Token and PIN Number". Available at http://eprint.iacr.org/2002/164.*

[8] H. M. Sun and B. T. Hsieh. *Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings. Available at http://eprint.iacr.org/2003/113.*

[9] D. Boneh and M. Franklin. *Identity-Based Encryption from Weil Pairing. In proceedings of Crypto 2001, LNCS 2139, pages 213-229, Springer-Verlag, 2001.*

[10] G. Xie *Cryptanalysis of Noel McCullagh and Paulo S. L. M. Barreto's Two-party Identity- Based Key Agreement. Available at http://eprint.iacr.org/2004/308.*

[11] A. Joux. *"A One Round Protocol for Tripartite Diffie-Hellman." In proceedings of ANTS 4, LNCS 1838, pp. 385-394, Springer-Verlag, 2000.*

[12] K. R. Choo. *"Revisit of McCullagh-Barreto Two-Party ID-Based Authenticated Key Agreement Protocols." Available at http://eprint.iacr.org/2004/343.*

[13] I. R. Jeong, J. Katz and D. H. Lee. *"One-Round Protocols for Two-Party Authenticated Key Exchange". In proceedings of ACNS 2004, LNCS 3089,pp. 220-232, Springer-Verlag, 2004*

[14] 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.*

[15] N. McCullagh and P. S. L. M. Barreto *" A New Two-Party Identity-Based Authenticated Key Agreement". In proceedings of CT-RSA 2005, LNCS 3376, pp. 262-274, Springer-Verlag, 2005.Also available at http://eprint.iacr.org/2004/122.*

[16] BlakeŰWilson, S., and A. Menezes (1999) *"Authenticated DiffieŰHellman key agreement protocols". In Proc. of the 5th Annual Workshop on Selected Areas in Cryptography (SAC Š98). Lecture Notes in Computer Science,1556. pp. 339Ű361.*

[17] Hwang, M.S, and W.P. Yang *"Conference key distribution schemes for secure digital mobile communications".IEEE J. Sel. Areas Comm., 13, 416Ű420.*

[18] Kaliski, B. (2001). *"An unknown key-share attack on theMQV key agreement protocol". ACMTrans. Information and System Security, 4(3), 275Ű288.*

[19] Shim, K. (2003). *"Efficient ID-based authenticated key agreement protocol based on Weil pairing". Electronics Letters, 39(8), 653Ű654.*

[20] Smart, N.P. (2002). *"An identity based authenticated key agreement protocol based on the Weil pairing". Electronics Letters, 38, 630Ű632.*

[21] BlakeŰWilson, S., and A. Menezes (1999). *"Authenticated DiffieŰHellman key agreement protocols". In Proc. of the 5th Annual Workshop on Selected Areas in Cryptography (SAC Š98). Lecture Notes in Computer Science, 1556. pp. 339Ű361.*

[22] Tseng, Y.M. (2005a). *"An improved conference-key agreement protocol with forward secrecy". Informatica,16(2), 275Ű284.*

[23] Tseng, Y.M. (2005b). *"A robust multi-party key agreement protocol resistant to malicious participants". The Computer Journal, 48(4), 480Ű487.*

[24] M.Bellare, R. Canetti, and H. Krawczyk. *"A Modular Approach to the Design and Analysis of Authentication and Key-Exchange Protocols. STOC Š98.*

[25] S. Blake-Wilson and A. Menezes *" Authenticated Diffie-Hellman Key Agreement Protocols. Selected Areas in Cryptography, 1998".*

[26] C. Boyd and J.M.G. Nieto. *"Round-OptimalContributory Conference Key Agreement. Public Key Cryptography, 2003".*

[27] E. Bresson, O. Chevassut,and D. Pointcheval *" Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. Adv.in Cryptology Ů Eurocrypt 2002 .*

[28] E.Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater. *"Provably Authenticated Group Diffie-Hellman Key Exchange.ACM Conf. on Computer and Communications Security, 2001.*

[29] M.Burmester and Y. Desmedt *"A Secure and Efficient Conference Key Distribution System.Advances in Cryptology Ů Eurocrypt Š94*

[30] D. Denning and G. M. Sacco. *"Timestamps in Key Distribution Protocols. Comm. ACM 24(8): 533Ű536 (1981).*

[31] W. Diffie and M. Hellman. *"New Directions in Cryptography. IEEE Trans. Information Theory 22(6): 644Ű654 (1976).*

[32] W. Diffie, P. van Oorschot, and M. Wiener *"Authentication and Authenticated Key Exchanges. Designs, Codes, and Cryptography 2(2): 107Ű125 (1992).*

[33] M. Just and S. Vaudenay *"Authenticated Multi-Party Key Agreement Adv. in Cryptology Ů Asiacrypt Š96*

[34] J. Katz and M. Yung *"Scalable Protocols for Authenticated Group Key Exchange". Adv. in Cryptology Ů Crypto 2003*

[35] L. Law, A. Menezes, M. Qu, J.Solinas, and S. Vanstone. *"An Efficient Protocol for Authenti- cated Key Agreement. Technical report CORR 98-05, University of Waterloo,1988*

[36] T. Matsumoto, Y. Takashima, and H. Imai *"On Seeking Smart Public-Key Distribution Sys- tems. Trans. of the IECE of Japan, E69,pp. 99Ű106, 1986".*

[37] *"National Security Agency. SKIPJACK and KEA Algorithm Specification. Version 2.0, May 29, 1998".*

[38] V. Shoup *"On Formal Models for Secure Key Exchange. Available at http://eprint.iacr.org".*

[39] M. Steiner, G. Tsudik, and M. Waidner. *"Diffie-Hellman Key Distribution Extended to Group Communication. ACM Conf. on Computer and Communications Security, 1996".*

[40] W.-G.Tzeng *"A Practical and Secure-Fault-Tolerant Conference-Key Agreement Protocol. Public Key Cryptography, 2000.19"*

[41] M.Bellare and P.Rogaway *Optimal assymetric encryption. In advanced in Cryptology"Eurocrypt 94, pages 92-111, 1995*

[42] W. Diffie, P. van Oorschot, and M. Wiener. *Authentication and Authenticated Key Exchanges. Designs, Codes, and Cryptography 2(2): 107Ű125 (1992).*

[43] L. Chen, C. Kudla *Identity based key agreement protocols from pairings, in: Proc. the 16th IEEE Computer Security Foundations Workshop, 2002, pp. 219-213.*

[44] Boneh, D. and M. Franklin *Identity-based encryption from the Weil pairing. In Advances in Cryptology Ű CRYPTO Š01, LNCS 2139, pages 213Ű229, Springer-Verlag, 2001*

[45] Hess F *Efficient identity based signature schemes based on pairings. Proceedings of the Ninth Annual Workshop on Selected Areas in Cryptography.*

[46] N. Koblitz *A course in Number Theory and Cryptography ,2nd edition Springer-Verlag-1994.*

Author Biographies

Jayaprakash Kar has completed his M.Sc and M. Phil. in Mathematics from Sambalpur University, M.Tech in Computer Science from Utkal University, and pursuing Ph.D at Utkal University, Bhubaneswar, India. He has worked on key agreement,password-based protocols, word-based public key cryptography and RFID authentication Protocol. His current research interests is on Elliptic Curve Cryptosystem & Generation of Random numbers and Cryptographic Protocol include key management problem of broadcast encryption. Mr. Kar is presently working as a Lecturer at Department of Information Technology, Al Musanna College of Technology, Ministry of manpower, Sultanate of Oman.

Banshidhar Majhi is currently working as a professor and head of the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, India. He has completed his M.Tech and Ph.D. from Sambalpur University. He has 15 Journal papers and 60 Conference articles to his credit. His research interests include image processing, cryptography and network security, soft computing.