Elliptic Curve based Authenticated Session Key Establishment Protocol for High Security Applications in Constrained Network Environment

K R Chandrasekhara Pillai[1] and M P Sebastian[2]

[1]Dept.of Computer Science and Engineering, N S S College of Engineering
Palakkad- 678008, Kerala - 678 008, India.
krcp@rediffmail.com
[2]Indian Institute of Management Kozhikode,
Calicut - 673570, Kerala, India
sebasmp@iimk.ac.in

ABSTRACT

The existing authenticated session key establishment protocols are either vulnerable to dictionary attack on identity privacy of a client or the methods adopted to resist this attack are found to be computationally inefficient. This paper proposes a new authenticated key establishment protocol which uses elliptic curve based DDH problem. The protocol provides identity privacy of the client in addition to the other security properties needed for a session key establishment protocol. In comparison with the existing protocols, the proposed protocol offers equivalent security with less parameters resulting in lower computational load, communication bandwidth cost, power consumption and memory requirement.

KEYWORDS

Elliptic Curve Cryptography, Authentication, Session Key Establishment, Network Security & Identity Privacy

1. INTRODUCTION

In the recent years, a variety of authenticated session key exchange protocols have been proposed for high security applications like banking, mobile telephony, and public wireless LANs (PWLANs). In such applications generally two different factors are used to authenticate and thus provide higher level of authentication assurance than one-factor authentication. An authentication factor can be defined as any information and process, which can be used to authenticate the identity of some entity. Park and Park [1] proposed a two factor authenticated key exchange (PP-TAKE) protocol with two factors including a password and a token (e.g., a smart card with a stored secret key) suitable for low-power PDAs in PWLANs. This scheme was supposed to provide mutual authentication and key exchange with identity privacy, half-forward secrecy, and low computation and communication cost.

Following the PP-TAKE protocol, a variety of authenticated session key exchange protocols have been proposed as improvement on it. Juang and Wu [2] pointed out that the PP-TAKE protocol is vulnerable to the dictionary attack upon identity privacy as the entropy of all possible clients' identifications is not very high. They proposed two new schemes for mutual

DOI : 10.5121/ijnsa.2010.2310

authentication and key exchange with less message exchanges than PP-TAKE protocol. They also claimed that both the schemes provide forward secrecy at client and one of the schemes has the ability to ensure identity privacy. However, we observe that the implementation of identity privacy in that Juang et al.'s scheme is not clear.

Yoon and Yoo [3] proposed another session key exchange protocol based on the PP-TAKE protocol with lower computation cost and less number of message exchanges, claiming the other desirable properties remained intact. However, we observe that their scheme does not provide identity privacy and is vulnerable to the dictionary attack.

Lee, Kim, and Won [4] suggested two session key exchange protocols and one of them provides identity privacy. However, we observe that the communication cost in these protocols is higher than that of the other related protocols having similar features. Further, these protocols cannot provide explicit key confirmation and provide only half forward secrecy.

In this paper, we propose a new elliptic curve based authenticated session key establishment protocol with the ability to ensure strong identity privacy. The proposed protocol uses elliptic curve based Decision Diffie-Hellman (DDH) problem. As we use elliptic curve cryptographic system with higher strength per key bit, the proposed protocol has the benefits of lower computational load, communication bandwidth cost, power consumption and memory requirement.

The rest of the paper is organized as follows. Section 2 reviews the related work and Section 3 presents the proposed protocol. Section 4 and 5 analyze the security and efficiency of proposed protocol, respectively. Section6 concludes the paper.

2. RELATED WORK

Since Lamport [5] proposed a password authentication scheme for remote user authentication with insecure communication, several password authentication schemes [6-11] and password authenticated key exchange schemes [12-15] have been proposed. However, these schemes are not designed for high security wireless environment, as the wireless devices are low powered and require low communication and computation cost. Park & Park [1] proposed a two factor authenticated key exchange (PP-TAKE) protocol for mutual authentication and session key exchange suitable for high security wireless environment. Following the PP-TAKE protocol, a variety of two factor authenticated key exchange protocols have been proposed as improvement on it. In this section, we briefly review the features and weaknesses of the existing TAKE protocols. The following notations are used throughout this article.

A: the client A

B: the server B

π: the password of A

t: the shared master key between A and B

$E_f( )$: symmetric encryption function using the symmetric key f

$D_f( )$: symmetric decryption function using the symmetric key f

ID_A: client A's identification

$h( )$: secure one-way hash function

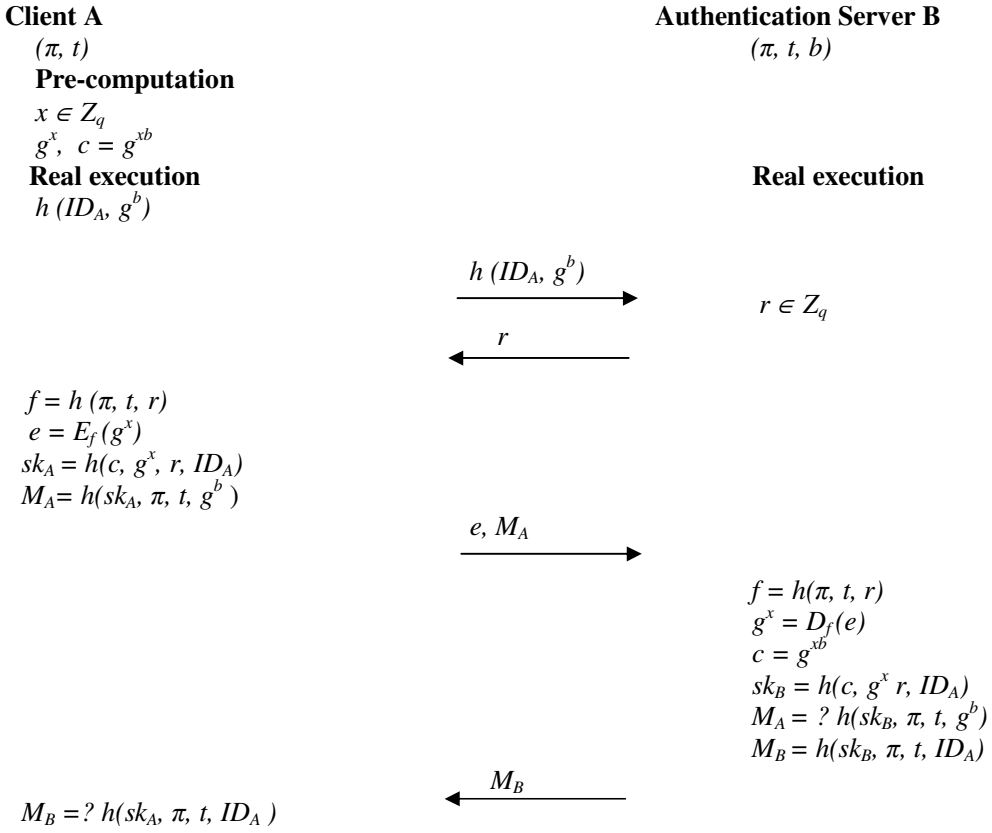
sk_A: session key generated by A

sk_B: session key generated by B, where $sk_A = sk_B$

2.1 The PP-TAKE protocol

The PP-TAKE protocol is based on the discrete logarithm based DDH problem [16-18] and has three phases: the enrollment phase, the pre-computation phase and the real execution phase. The summary of the protocol is shown in Figure 1. It assumes that *A* and *B* share the domain parameters *(p, q, g)*, where *p* is a large prime number, *q* is a prime divisor of *(p-1)* and *g* is an element of order *q* in Z_p^*. For simplicity, *(mod p)* operations are not explicitly indicated in this article.

In the enrollment phase, A and B share a password π and a shared master key *t*, where π is stored in both *A*'s and *B*'s storage and *t* is stored in a secure token (such as a smart card) at *A* and also in *B*'s storage along with ID_A. Then, *B* chooses a random number $b \in Z_q$ and computes g^b, where *b* denote the server's static private key and g^b denote the server's public key. *A* is informed of the domain parameters and g^b. The pre-computation phase is executed off-line prior to the real execution phase. In this phase, *A* chooses a random number $x \in Z_q$ and computes g^x and $c = g^{xb}$ in advance so that the computation cost in real execution phase is reduced.



Session key, $sk_A = sk_B = h(c, g^x, r, ID_A)$

Figure 1. The PP-TAKE protocol

The real execution phase performs the execution of the protocol for the mutual entity authentication and session key establishment and it is composed of the following steps:

1. A computes the hash value $h(ID_A, g^b)$ and sends it to B for requesting the authentication service.
2. Upon receiving $h(ID_A, g^b)$, B searches its database entries for a match with the content of hash value field. If a matching entry is obtained it extracts (ID_A, π, t, b) from the corresponding entry and thus obtains the real identity ID_A of A. If identity ID_A is obtained, B selects a random number $r \in Z_q$ and sends it back to A.
3. Upon receiving r from B, A computes the symmetric key $f = h(\pi, t, r)$ and $e = E_f(g^x)$. A then computes the session key $sk_A = h(c, g^x, r, ID_A)$ and the authenticator $M_A = h(sk_A, \pi, t, g^b)$ and sends (e, M_A) to B.
4. Upon the receipt of (e, M_A), B generates $f = h(\pi, t, r)$ and $g^x = D_f(e)$. B then computes $c = g^{xb}$ and the session key $sk_B = h(c, g^x, r, ID_A)$ and checks whether $M_A = h(sk_A, \pi, t, g^b)$. If yes, B can ensure A's identity and A's authentication is completed successfully. B computes the authenticator $M_B = h(sk_B, \pi, t, ID_A)$, and sends M_B to A.
5. Upon receiving M_B, A checks if $M_B = h(sk_A, \pi, t, ID_A)$. If yes, A believes that B is the valid server and B's authentication is successful. Thus the mutual authentication is successfully achieved.

The main weakness of the PP-TAKE protocol is that, it does not provide adequate identity privacy using $h(ID_A, g^b)$ because with the server's public key g^b, the adversaries can also compute $h(ID_A, g^b)$ easily using the dictionary attack [19, 20] for all possible identifications. The user identity cannot be protected with this protocol, since the entropy of all possible clients' identifications is not very high. Moreover, wireless devices require low power and low communication and computation cost for user authentication. Four messages are exchanged between the server and the client in this protocol, whereas, in most of the other TAKE protocols, only two/three messages are exchanged.

2.2. The Juang et al.'s protocols

Juang et al.'s protocols are modifications of the PP-TAKE protocol [2]. These protocols are also based on the discrete logarithm based DDH problem with three phases, but they have fewer message exchanges than the PP-TAKE protocol. The first protocol is simpler but it does not provide identity privacy. The second protocol provides identity privacy. In this protocol, during the enrollment phase (in additional to the task in PP-TAKE scheme), A has to store an index value i, whose initial value is equal to zero, indicating that A and B are in the i^{th} connection. The pre-computation phase task is same as in the PP-TAKE scheme. In the real execution phase, for achieving identity privacy, instead of using the real identification ID_A of the client, a pseudo identification $SID_{A,i} = h(\pi, t, i)$ is used [2]. Here, three messages are exchanged between the client and the server.

In the second Juang et al.'s protocol, the procedure for the protection of identity privacy at B is not described. It is not clear whether B stores $SID_{A,i}$ in its database or not. In the first step of the real execution phase of the protocol, A sends $(e, SID_{A,i}, i)$ to B, requesting the service. If B stores $SID_{A,i}$ in its database, after B receives $(e, SID_{A,i}, i)$, it can use the parameter i to identify the database table, if separate tables are created and updated dynamically for every next possible index values for reducing the search time for $SID_{A,i}$. If B does not store $SID_{A,i}$ in its database, it has to perform an exhaustive search to find a $SID'_{A,i} = h(\pi', t', i)$ from its database that is identical to the received value $SID_{A,i} = h(\pi, t, i)$. For each entry in the database, B has to compute $SID'_{A,i} = h(\pi', t', i)$ and compare it with $SID_{A,i}$ until both values are identical. In this case, the search and hash operations at the server during the login phase of a client are time

consuming and require high computation cost when a reasonably large number of clients are enrolled with *B*. Further, it may not be appropriate to compute the hash values of master secrets *(π, t)* with index values (ascending natural numbers) and make them public for the sake of achieving identity privacy as it may open up new opportunities for the adversaries to make more serious attacks.

2.3. The Yoon et al.'s protocol

Yoon et al.'s protocol is another TAKE protocol which attempts to optimize the PP-TAKE protocol by reducing the communication and computation loads [3]. The enrollment phase and the pre-computation phase of this protocol are similar to that of the PP-TAKE protocol. In the real execution phase of Yoon et al.'s scheme, three messages are exchanged between the client and the server. For the calculation of e at A and g^x at B, simple $\oplus$ operation is used instead of the symmetric encryption/decryption of PP-TAKE protocol. However, $\oplus$ operation of two parameters of different size (e.g., f =160 bits and g^x = 1024 bits) may reduce the security offered by the protocol. Yoon et al.'s protocol also cannot ensure identity privacy similar to that of the PP-TAKE protocol.

2.4. The Lee et al.'s protocols

Lee et al.'s [4] proposed two TAKE protocols requiring only two message exchanges and one of them provides identity privacy. Even though these protocols require less message exchanges, the total number of parameters exchanged and the corresponding communication load are more than that of the other related protocols having similar security features. Moreover, these protocols cannot satisfy explicit key confirmation since the server cannot be assured that the client actually possesses the session key.

3. THE PROPOSED PROTOCOL

We propose a new authenticated session key establishment protocol, which is based on the elliptic curve DDH problem. The primary advantage of elliptic curve DDH problem over discrete logarithm based DDH problem is that the current best algorithms known for solving the elliptic curve DDH problem to break the security takes fully exponential time where as the discrete logarithm DDH problem takes sub exponential time [21-23]. Consequently, smaller parameters can be used in elliptic curve based system than in the discrete logarithm based system, while maintaining the same level of security. It is seen that 1024 bits discrete logarithm based DDH is approximately equivalent to 139 bits elliptic curve logarithm based DDH [21]. In the proposed protocol an elliptic curve E defined over $GF(p)$ with a large group G of points on the curve of order q and a base point (generator) g of large order n (the order of a point g on an elliptic curve is the smallest positive integer n such that $ng = O$, where O is the point at infinity) is assumed. Let the group G has a large embedding degree k (a group is said to have an embedding degree k if the group order q divides p^k-1, but does not divide p^i-1 for all $0 <i <k$. It assumes that A and B share the parameters of the elliptic curve E and group G and the generator g. The proposed protocol has three phases: the enrolment phase, the pre-computation phase and the real execution phase. The summary of the protocol is as shown in Figure 2.

3.1. The enrolment phase

In the enrolment phase, A and B share a password π and a shared master key t and store them in the secure storage/smart card. B also chooses a random number $b \in Z_n^*$ and then computes bg, where b denotes B's static private key and bg denotes B's public key. The initial hash value $(HID_A = h(ID_A, bg, t))$ is computed and stored in the field *FHID* of A's entry in B's secure database. The field *FHID'* is initialized to zero and the parameters (ID_A, π, t, b) are also stored

in the appropriate fields of the database. *A* is informed of the domain parameters *(p, q, g)* and *B*'s public key *bg*.

3.2. The pre-computation phase

The pre-computation phase is executed off-line prior to the real execution phase. In this phase, *A* chooses a random number $x \in Z_n^*$ and computes *xg*. The integer variable *n* takes the value *t* if the authentication service request is the first one after the enrolment phase and otherwise it takes the first 128 bits of the current value of *c*. Then the value *x(bg)* is computed and it is assigned to the variable *c*. These computations are performed in advance so that the computation cost in the real execution phase is lower.

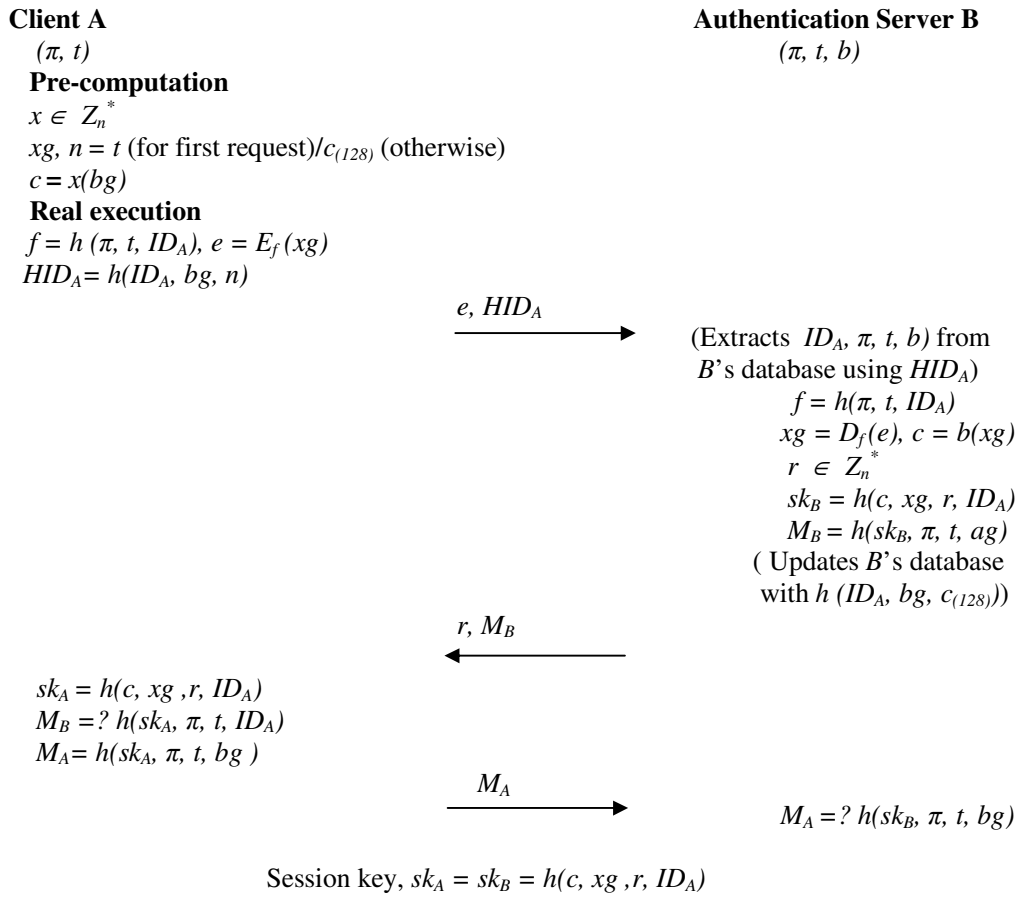


Figure 2. The proposed protocol

3.3. The real execution phase

The real execution phase performs mutual entity authentication and session key establishment, which consists of the following steps:

1. *A* computes the hash value $HID_A = h(ID_A, bg, n)$, and $f = h(\pi, t, ID_A)$ and then the value $e = E_f(xg)$. *A* then sends HID_A and e to *B* for requesting the authentication service.
2. Upon receiving HID_A, *B* first searches its database entries for a match with the contents of the field *FHID* and HID_A, if no match is obtained, it tries for a match with the contents of the field *FHID'* and HID_A. If a matching entry is obtained, it extracts (ID_A, π, t, b) from the corresponding entry of its database and thus obtains the real identity of *A*. After that, *B* computes $f = h(\pi, t, ID_A)$, and $xg = D_f(e)$ and, then computes $c = b(xg)$. Next, *B* selects a random number r and generates the session key $sk_B = h(c, xg, r, ID_A)$, and the authenticator $M_B = h(sk_B, \pi, t, ID_A)$. *B* then updates the corresponding database entry's *FHID'* field with the contents of *FHID*, and the *FHID* field with the value $h(ID_A, bg, c_{(128)})$ and sends (r, M_B) to *A* (To minimize the computational cost in the real execution phase, this updating of database can be deferred to off-line by properly maintaining a flag).
3. Upon the receiving of (r, M_B), *A* generates the session key $sk_A = h(c, xg, r, ID_A)$ and checks whether $M_B = h(sk_A, \pi, t, ID_A)$. If yes, *A* believes that *B* is authenticated and uses this session key sk_A to communicate with *B*. When *A* verifies successfully the validity of *B*'s identity, *A* computes the authenticator $M_A = h(sk_A, \pi, t, bg)$, and sends M_A to *B*.
4. Upon receiving M_A, *B* checks whether $M_A = h(sk_B, \pi, t, bg)$. If yes, *B* believes that *A* is authenticated and uses the session key sk_B to communicate with *A* securely later.

As in any communication protocol, after sending message 1, if the reply message 2 is not reaching *A* within a timeout period, then *A* will resend the message 1 to *B* until the reply message 2 reaches or the number of attempts exceeds the maximum permissible number. Similarly, after sending message 2 if the reply message 3 is not reaching *B* within a timeout period, then *B* will resend the message 2 to *A* until the reply message 3 reaches or the number of attempts exceeds the maximum permissible number.

4. SECURITY ANALYSIS

In this section, we analyze the security of the proposed protocol. The security parameters considered include identity privacy, explicit mutual authentication, session key establishment, forward secrecy, resistance to off-line dictionary attack, key confirmation, and non-repudiation [1-4]. The security analysis is summarized in Table 1.

Table 1. Security and functionality comparison of the proposed protocol and related protocols

Security and functionality	**P1**	**P2**	**P3**	**P4**
Withstanding the dictionary attack upon a client's identity	No	Yes	Yes	Yes
Providing the forward secrecy at the client	Yes	Yes	Yes	Yes
Explicit key confirmation	Yes	Yes	No	Yes
Computation cost at the client for the login phase	Low	Low	Low	Low
Computation cost at the server for the login phase of a client	Low	High	Low	Low
Hash value of master secrets in public domain	No	yes	No	No

P1: PP-TAKE; P2: Juang et al.'s; P3: Lee et al.'s; P4:Proposed protocol

4.1. Identity privacy

To ensure the privacy in personal communication, it is necessary to protect the client's identity from passive attacks such as eavesdropping. In Step1 of the proposed protocol, *A* sends $HID_A = h(ID_A, bg , n)$, so that even if the adversaries try to find the identity of the client using the dictionary attack for all possible identifications, they cannot succeed as there is no possibility of knowing the random value *n*. *B*, during the enrollment phase of *A*, initialized the *FHID* field of *A*'s entry in the server database with the hash value $h(ID_A, bg, t)$ and it will be the value of HID_A generated during the first authentication service request in the real execution phase. Thereafter, upon receiving the HID_A, *B* searches the database for an entry with the content of *FHID* field matches with the current value of HID_A and if no such entry is obtained, it tries for a match with the contents of the *FHID'* field and HID_A. If a matching entry is obtained, *B* extracts the corresponding parameters and identifies *A*. After computing fresh values for *c*, sk_B and M_B, *B* updates the database before sending the message 2 to *A*.

For the generation of HID_A at *A*'s side, in addition to ID_A and *bg* the previous value of *c* (i.e.,n = $c_{(128)}$) is also used. *B* has already updated client *A*'s entry in its database with this hash value during the previous service request, so that when the current service request reaches the server, *A* can be identified when an identical hash value HID_A is obtained from *B*'s database. As the hash value HID_A is used to identify *A*, the generation of the hash value at *A*'s side and the updating of the database at *B*'s side are to be synchronized. When *A* sends a service request to *B* and if the reply message is not received at *A*, then it may be due to the loss of message from *A* to *B*/*B* to *A* or due to *B*'s failure. When there is a loss of message from *A* to *B* or *B*'s failure, *A* will subsequently send the service request with the same hash value and the hash value is to be available in the *FHID* field of *A*'s entry in *B*'s database if the service request reaches *B*, and thus client *A* is identified. If *B* to *A* message is lost and *A* again sends a service request and as it reaches *B*, *B*'s database has already been updated and hence the corresponding hash value HID_A is available in the *FHID'* field of an entry of the database so as to identify *A*. Accordingly, if a matching value of HID_A is obtained from the *FHID* field, then the *B* updates the contents of the corresponding entry's *FHID'* field (with the contents of the *FHID* field) and then the *FHID* field (with the current value of HID_A) so that the next service request from *A* will be identified using the updated hash values. If a matching value of HID_A is obtained from the *FHID'* field of *B*'s database (which means that the currently received message 1 from *A* is a repeat message due to the non-receipt of the message 2 sent by *B* to *A*), the database has already been updated and no further updating is needed. Hence, for obtaining the parameters and the identity of a client, *B* has to first search the *FHID* field and then *FHID'* field of the database for a matching hash value (e.g., HID_A). In *B*'s database entries, by maintaining a fixed field with initial hash values (e.g., $HID_A = h(ID_A, bg, t)$) of the clients, the protocol can be re-initialized in the event of loss of values of *c* and/or *n*.

4.2. Explicit mutual authentication

Explicit mutual authentication between the client and the authentication server is necessary to prevent Man-in-the-Middle (MitM) attack. The goal of mutual authentication is to establish an agreed session key $sk_A = sk_B$ between *A* and *B* [5, 6, 25, 26]]. The protocol establishes an internal secret authentication key (*f*) to protect the parameters exchanged between *A* and *B*. The mutual authentication between *A* and *B* is completed, if there is a $sk_A = sk_B$, such that *A* believes that *A* and *B* share a common session key sk_A and *B* believes that *A* and *B* share a common session key sk_B for the transaction. In step 2 of the execution phase of our protocol, after *B* receives the message (e, HID_A) from *A*, it computes the symmetric key $f = h(\pi, t, ID_A)$ and the random challenge $xg = D_f(e)$ and then computes $c = b(xg)$. After that, *B* chooses the random number *r* and computes the session key $sk_B = h(c, xg, r, ID_A)$ and the authenticator $M_B = h(sk_B, \pi, t, ID_A)$, and believes that *A* and *B* share a common session key sk_B. In step 3, after *A* receives the message (r, M_B) from *B*, *A* first computes the session key $sk_A = h(c, xg, r, ID_A)$ and then checks

whether the authenticator M_B is equal to $h(sk_A, \pi, t, ID_A)$. If yes, A believes that sk_A and the random number r are authenticated by B. Thus, A believes that A and B share a common session key $sk_A = sk_B$. Since the random number x is chosen by A, A knows that x is fresh and sk_A generated using that, is the current session key. On verifying the authenticator $M_B = h(h(bxg, xg, r, ID_A), \pi, t, ID_A)$, A can make sure that xg is embedded in bxg by B and then A is sure that B believes A and B share a common session key $sk_B = sk_A$ Since the random number r is chosen by B, B knows that the random number r is fresh and the sk_B generated using that is the current session key. In step 4, after B receives the message M_A from A, B checks if $M_A = h(sk_B, \pi, t, bg)$. If yes, B is sure that A believes A and B share a common session key $sk_A = sk_B$, since the random number r is embedded in $sk_A = h(c, xg, r, ID_A)$, and which is embedded in $M_A = h(sk_A, \pi, t, bg)$ [2, 5]. Hence, the proposed protocol satisfies the explicit mutual authentication property.

4.3. Session key establishment

The communicating parties establish a secret session key for protecting data to be exchanged during the current session. The random values x and r, are separately generated by different entities for the current session. Therefore, the established session keys sk_A and sk_B are fresh for the current transaction. Hence, the proposed protocol is free from replay attack and modification attack.

4.4. Forward secrecy

Ensuring forward secrecy is a must so that the adversaries are prevented from computing the session keys even when the long-term secret parameters of an entity participating in the key exchange protocol have been revealed. If an adversary knows (π, t) on the client's side, he/she may compute $xg = D_f(e)$, but cannot compute $c = b(xg)$ and hence the session key because of the elliptic curve DDH problem [21-23]. Therefore, the forward secrecy of the client is ensured. However, if the adversary knows (b, π, t) on the server's side, he/she may compute the session key. The reason is that the adversary can compute $xg = D_f(e)$ followed by $c = b(xg)$ and the session key. Here, it is assumed that the server is secure and its long term secret parameters can no be compromised and forward secrecy at the server is not necessary.

4.5. Resistance to off-line dictionary attack

Two types of dictionary attack are considered here, one is the attack on the client's identity privacy and the other is the on the session key [2]. As stated above, the proposed protocol is capable of ensuring protection from the dictionary attack on the client's identity privacy. Also, without knowing the shared master key t, the shared password π and the random challenge xg simultaneously, an adversary cannot perform the dictionary attack to obtain the session key.

4.6. Key confirmation

In the proposed protocol, both client and server compute the same session key, $sk_A = sk_B = h(c, xg, r, ID_A)$ and the authenticators M_A and M_B respectively and exchange them for key confirmation. From Table 1, it can be observed that all the protocols except Lee et al.'s protocol provide explicit key confirmation by both server and client. In Lee et al.'s protocol, the server computes the authenticator and sent it to the client so that the client can be assured that the server possesses the session key, where as the client does not send any authenticator to the server and hence the server can not be assured that the client possesses the session key.

4.7. Non-repudiation

Even without using a digital signature, the proposed protocol can ensure non-repudiation by means of the strong two-factor authentication. It provides non-repudiation of origin of data by the user and the server for the data sent from the user to the server and vice versa.

The interoperability feature, that is allowing the negotiation of symmetric key algorithm between the communicating parties, can be added to the protocol by changing the exchanged message format and implementing several well known encryption algorithms at both server and client terminals.

5. EFFICIENCY ANALYSIS

In this section, we analyze the efficiency of the proposed protocol and compare it with that of the other related TAKE protocols. The properties such as computational load, number of message exchanges, communication cost and memory requirement are the parameters used for the efficiency analysis [2-4]. Let p be of 1024 bits and q be of 160 bits in order to make the discrete logarithm problem practically difficult in the related protocol considered in this article. Even though 1024 bits discrete logarithm based DDH is equivalent to 139 bits elliptic curve based DDH, for achieving higher security for now and immediate future, for the proposed protocol we consider an elliptic curve over Z_p with the parameters p and n (key size) as 160 bits and 161 bits respectively [21-23]. Let the output size of the secure one-way hash functions be 160 bits. Let the key size of symmetric cryptosystems be 128 bits [27, 28]. Let the bit lengths of the identification of clients (e. g., ID_A) and the parameter i in Juang et al.'s scheme be 32 bits [2]. Let the bit length of current time representation t_s in Lee et al.'s protocol be 32 bits [4]. The efficiency analysis is summarized in Table 2.

5.1. Computational load

It is desirable for the session key establishment protocol to have low computational load as the client devices are usually low power and constrained devices such as PDAs. In order to minimize the computational load in the real execution phase, maximum possible operations are included in the pre-computation phase itself. All session key exchange protocols considered for comparison include two exponential operations of client in the pre-computation phase. The proposed protocol includes two elliptic curve point multiplication operations of client in the pre-computation phase. In the real execution phase of the client side, PP-TAKE, Juang et al.'s and Lee et al.'s protocols require five hash operations and one encryption operation, whereas the proposed protocol requires five hash operations, one encryption operation. At the server side, PP-TAKE and Lee et al.'s protocols require four hash operations, one decryption operation and one exponential operation, whereas the proposed protocol requires four hash operations, one decryption operation and one elliptic curve point multiplication operation. In Juang et al.'s protocol, the total number of hash operations depends on the database search and the identification of pseudo identification of clients (e.g., ID_A). In an efficient and optimized elliptic curve cryptography implementation having minimum computation time and code size requirements, 160 bits elliptic curve point multiplication is less complex than 1024 bits exponential operation [21-23]. Hence, the computational load of the proposed protocol is lower than the computational load of the existing relevant protocols.

5.2. Number of message exchanges

For achieving network resource efficiency and minimum latency and set up time, the number of message exchanges between the client and the server should be kept as minimum as possible. The PP-TAKE and Lee et al.'s Protocols require four message exchanges and two message exchanges respectively, whereas the remaining two protocols (including the proposed protocol) require three message exchanges.

Table 2. Efficiency comparison of the proposed protocol and the related protocols

Protocol	E1	E2	E3	E4	E5
PP-TAKE	2 Exp	5 Hash+1Sym	4 Hash+1Sym+1Exp	4	1664
Juang et al.'s	2 Exp	5 Hash+1Sym	Not specified	3	1696
Lee et al.'s	2 Exp	5 Hash+1Sym	4 Hash+1Sym+1Exp	2	1856
Proposed	2 Emul	5 Hash+1Sym	4 Hash+1Sym+1Emul	3	960

E1: computation cost of the pre-computation phase; E2: computation cost of the execution phase for a client; E3: computation cost of the execution phase for a server; E4: numbers of message exchanges; E5: communication cost during execution phase (bits); Exp: exponential operation; Emul: elliptic curve multiplication operation; Hash: hashing operation; Sym: symmetric encryption or decryption.

5.3. Communication bandwidth

The communication bandwidth of a protocol depends upon the size of the exchanged messages and the number of messages. In the real execution phase, the communication cost in bits, in accordance with the assumptions made, is 1664 bits for PP-TAKE protocol, 1696 bits for Juang et al.'s protocol, 1856 bits for Lee et al.'s protocol and 960 bits for the proposed protocol. Hence, the proposed protocol is more efficient than the existing relevant protocols in terms of communication bandwidth cost.

5.4. Memory requirement

The memory needed in the smart card of a client to store the shared master keys is 256 bits in Lee et al.'s protocol, and in all other protocols it is 128 bits. It is assumed that the password is not stored in the secure smart card. The other types of memories needed to store the data is less in the proposed protocol as the size of the parameters p is of 160 bits, n is of 161 bits and e is of 320 bits, where as in other protocols both p and e are of 1024 bits.

From the efficiency analysis, it is clear that the proposed protocol is superior to the other relevant session key exchange protocols.

6. CONCLUSIONS

In this paper, we have proposed a secure and authenticated session key establishment protocol for high security applications in constrained network environment. It uses cryptographic mechanisms of hash operation, symmetric encryption/decryption and elliptic curve based DDH problem. The proposed protocol resists dictionary attack on identity privacy and provides forward secrecy at the client, in addition to the other security properties needed for a session key establishment protocol. The protocol thereby protects the clients' identities and ensures the privacy in personal communication. As it also provides the necessary forward secrecy at the client's side, the adversaries are prevented from computing the session key even when the long term secret parameters of clients have been revealed. The security and efficiency analyses show that the proposed protocol performs better than the existing popular protocols.

REFERENCES

[1] Y. Park & S. Park, (2004) "Two factor authenticated key exchange (TAKE) protocol in public wireless LANs", IEICE *Trans Commun.*, Vol. E87-B(5), pp1382–1385.

[2] W. Juang & J. Wu, (2009) "Two factor authenticated key exchange protocol in public wireless LANs", *Computers and Electrical Engineering*, Vol. 35, pp 33-40.

[3] E. Yoon & K. Yoo, (2006) "An optimized two factor authenticated key exchange protocol in PWLANs", ICCS, Vol. II-LNCS 3992, pp1000-1007.

[4] Y.Lee, S. Kim & D. Won, (2009) "Enhancement of two factor authenticated key exchange protocols in public wireless LANs", *Comput. Electr. Eng, 2009*, doi: 10. 1016/j.compeleceng.2009.08.007.

[5] L. Lamport, (1981) "Password authentication with insecure communication", *Commun.* ACM, Vol. 24(11), pp770–772.

[6] H. Sun, (2000) "An efficient use authentication scheme using smart cards", IEEE *Trans. Consum. Electron.,* Vol. 46(4), pp958–961.

[7] A. Awathi & S. Lal, (2003) "A remote user authentication scheme using smart cards with forward secrecy", IEEE *Trans. Consum. Electron.,* Vol. 49(4), pp1246-1248.

[8] A. Awathi & S. Lal, (2004) " An enhanced remote user authentication scheme using smart cards", IEEE *Trans. Consum. Electron.,* Vol. 50(2), pp 583-586.

[9] W. Ku, & S. Chen, (2004) "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards", IEEE *Trans. Consum. Electron.,* Vol. 50(1), pp204-207.

[10] W. Juang, (2004) "Efficient password authenticated key agreement using smart card", *Computer Security,* Vol. 23, pp167–173.

[11] X. Wang, W. Zhang, J. Zhang, & M. Khan, (2007) "Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards", *Comput. Stand. Interf.* , Vol. 29(5), pp507–512.

[12] J. Katz, R. Ostrovsky & M. Yung, (2004) "Efficient password authenticated key exchange using human-memorable password", *In Eurocrypt 2001*, pp475-494.

[13] F. Zhu, D. S. Wong, A. H. Chan & R. Ye, (2002) "Password authenticated key exchange based on RSA for imbalanced wireless networks", *In ISC 2002*, pp150-161.

[14] D. S. Wong, A. H. Chan & F. Zhu, (2003) "More efficient password authenticated hey exchange based on RSA", *In Indocrypt 2003*, pp375-387.

[15] M. Zhang, (2004) "New approaches to password authenticated key exchange based on RSA", *In Asiacrypt 2004*, pp230-244.

[16] W. Diffie & M. Hellman, (1976) "New directions in cryptography", IEEE *Trans. Inform. Theo*., Vol. 22, pp 644–654.

[17] C. Yang & R. Wang, (2004) "Cryptanalysis of a user friendly remote authentication scheme with smart cards", *Computer Security*, Vol. 23*,* pp425–427.

[18] R. Farashahi, B. Schoenmakers & A. Sidorenko, (2007) "Efficient pseudorandom generators based on DDH assumption, *In Public key cryptography-PKC '07*, 4440, Lecture notes in computer science, Spinger-verlag, pp426-441.

[19] S. M. Bellovin & M. Merritt, (1992) "Encrypted key exchange: password-based protocols secure against dictionary attacks", *In Proceedings of the 1992* IEEE CS *conference on research in security and privacy*, pp72-84.

[20] M. Bellare, D. Pointcheval & P. Rogaway, (2000) "Authenticated key exchange secure against dictionary attacks", *In Eurocrypt 2000*, pp139-55.

[21] A. K. Lenstra & E. R. Verheul, (1999) "Selecting cryptographic key sizes", *In Proceedings of 3rd workshop on elliptic curve cryptography (ECC 99)*, Waterloo, Canada, pp1-3.

[22] M. Aydos, T.Yanik & C. K. Koc, (2001) "High-speed implementation of an ECC-based wireless authentication protocol on an ARM microprocessor", IEEE *Proc. -Commun*., Vol. 148(5), pp273-279.

[23] K. Lauter, (2004) "The advantages of elliptic curve cryptography for wireless security", IEEE *Wireless Commun*., Vol. 11(1), pp 62-67.

[24] C. Yang & M. Hwang, (2004) "Cryptanalysis of simple authenticated key agreement protocols", IEICE *Trans. Commun.*, Vol. E87-A(8), pp2174-2176.

[25] V. Kolesnikov & C. Racko, (2008) "Password mistyping in two-factor-authenticated key exchange", ICALP, II- LNCS (5126), pp702–714.

[26] Z. Chai, Z. Cao & R. Lu, (2006) "Threshold pass authentication against guessing attacks in ad hoc networks", *Ad hoc Networks*, Vol. 5(7), pp1046-54.

[27] NIST FIPS PUB 180, (1998) Secure Hash Standard, National Institute of Standards and Technology, US Department of Commerce, DRAFT.

[28] NIST FIPS PUB 197, (2001) Announcing the ADVANCED ENCRYPTION STANDARD (AES), National Institute of Standards and Technology, US Department of Commerce.

K R Chandrasekhara Pillai received the B. Sc(Engg) degree in Electrical Engineering from University of Calicut, Kerala, India in 1980 and M E degree in Computer Science and Engineering from Anna University, Chennai, India in 1989. He is a faculty member of N S S College of Engineering, Palakkad, India and a research scholar of National Institute of Technology Calicut, India. He is a member of IEEE, fellow of IEI and IETE, and senior member of CSI. His research interests include cryptography, networking and network security.

M P Sebastian received the B.Sc(Engg) degree in Electronics and Communication Engineering from University of Kerala, Trivandrum, India. He received the M E and Ph. D degrees in Computer Science & Engineering from Indian Institute of Science, Bangalore, India. He is now Professor of Information Technology and Systems at IIM Kozhikode, India. Prior to joining IIM Kozhikode, he was serving as Professor & Head of Computer Science & Engineering at National Institute of Technology Calicut, India. He is a reviewer and member of editorial boards of many International Journals. His research interests include information security management, cloud computing, ERP, cryptography and mobile networks.