Establishment of ECC-based Initial Secrecy Usable for IKE Implementation

Sangram Ray and G. P. Biswas, *Member, IAENG*

***Abstract*—Internet Key Exchange (IKE) protocol is the most common usable mechanism to exchange keying materials and negotiate security associations between two distant entities. Similar to the several enhancements of the IKE protocols, the present paper proposes a new flexible approach for complexity reduction and security improvement of the IKE implementation. In this paper, an initial secret key negotiation based on Elliptic Curve Cryptography (ECC) for phase 1 of IKE has been proposed, which instead of RSA, uses ECC-based public key certificate for authentication of the entities. The proposed scheme thus requires comparatively less processing time and provides equivalent secrecy with less key size. An in-depth security analysis of the proposed method against several attacks is given that shows the protection of all attacks.**

***Index Terms*— Certificate Authority (CA), Elliptic Curve Cryptography (ECC), Internet Key Exchange (IKE) protocol, Internet Security Association and Key Management Protocol (ISAKMP), Security Association (SA)**

I. Introduction

THE secure communication over the Internet has become increasingly important and since it has no self security, the IPSec protocol has been developed for the protection of the same. The IPSec comprises two protocols called AH (Authentication header) and ESP (Encapsulated Security Payload) that provide security to the IP data packet with data integrity, confidentiality, authentication, protection against replay attack etc. The IKE protocol [1]-[19] is usually used in conjunction with IPSec as a key management mechanism or PKI (Public Key Infrastructure) that establishes security association (SA) to be used by the entities for secure transmission of IP packet over the Internet [6], [9]. That is, the IKE protocol provides SA to the IPSec, which in turn provides security to the IP datagrams. The IKE is mainly based on the ISAKMP (Internet security association and key management protocol) designed by NSA (National security agency). In essence, the SA comprises two communication entities/ security gateways for their mutual authentication, the generation of shared secret session keys, the negotiation and exchange confidential parameters between them. The parameters exchanged are SA lifetime, sequence number counter, security parameter index (SPI), IPSec mode, different cryptographic techniques, key related materials etc. [6], [7].

The IKE protocol works in two phases: Phase I initiates the beginning of IKE to provide the SA for the Phase II, which finally provides the SA to IPSec. In phase I, after SA-offered and negotiation, a common secret key is established between two entities. On the other hand, in Phase II, the SA for IPSec is established and the final keying materials to be used in IPSec are generated. The phase I, however, is again implemented in two modes namely Main Mode and Aggressive Mode. The main mode refers to the identity-protection exchange with six messages while the aggressive mode refers to the aggressive exchange with three messages. The four different authentication methods for implementation of Phase-I are defined in [5] and they are namely pre-shared key, public key signature, public key encryption and revised public key encryption techniques. The phase-II of IKE is normally implemented using a single mode called "quick mode", which uses three messages for establishing the SA of the IPSec protocol.

As stated, although the Phase-I of IKE can be implemented in four different ways, each of them uses a common scheme called Diffie-Hellman (DH) protocol to establish an initial common secret between the two parties [20], [21]. Since DH, due to incapability of authenticating the participants, is vulnerable to the man-in-the-middle attack, each method is supported by different mechanism like preshared secret, public key certificate, digital signature, cookies etc. for authentication purpose [2], [18], [19]. In fact, each of the above methods directly or indirectly requires a public key certificate issued by a CA (certificate authority). Since it is mainly a RSA based public key certificate, the overhead for maintaining and using it is sufficiently high, and this overhead appears to be much more than when an ECC based public key certificate of comparable security is used. The computation cost in ECC is much less as it involves mainly the point multiplication and the ECC based cryptosystem [22]-[26] with 160-bit key size provides equivalent security of the RSA cryptosystem with 1024-bit key size [22]-[26].

The Public-Key Infrastructure (PKI-X.509) working group of Internet Engineering Task Force (IETF) similar to the X.509 RSA based public key certificate [27]-[31], provides ECC based public key certificate standardized as the PKI-X.509, which specified as PKIX. Subsequently, the Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic Curve Diffie-Hellman (ECDH) public keys and the generation of ECC based certificate with ECDSA signature of PKIX are proposed in [25], [30], [31]. It may be noted that the PKIX is easily interoperable with PKI (X.509) and the Certificate Authority (CA) issues and certifies both the

Manuscript received January 05, 2012; revised March 20, 2012.

Sangram Ray (Research Scholar) is with the Department of Computer Sc. & Engg, Indian School of Mines, Dhanbad, DHANBAD-826004, INDIA (phone: 879-736-9171; e-mail: sangram.ism@ gmail.com).

G. P. Biswas (Professor) is with the Department of Computer Sc. & Engg, Indian School of Mines, Dhanbad, DHANBAD-826004, INDIA (e-mail: gpbiswas@gmail.com).

certificates. Thus, the efficient ECC based certificate scheme PKIX can be implemented on the existing PKI infrastructure, and the present paper without additional overheads uses it for the implementation of the phase I of IKE protocol.

In brief, this paper addresses the development of an initial secret key negotiation protocol usable for IKE implementation based on PKIX in elliptic curve cryptosystem. We assume that the existing tree-type hierarchical model [27]-[29] for CAs with PKI is capable for creating, storing, issuing and revoking any number of PKIX certificates. This model verifies the ECC based public keys of any entity in a chaining fashion from leave nodes towards the root of the tree, where the root CA, which has self-signed, self-issued certificate, completes the verification processes. All intermediate CAs issue certificates to the entities to relief the burden of the root CA and participate in the chaining verification process as mentioned above.

The remaining parts of the paper are organized as follow. Section 2 briefly introduces the basics of the Elliptic Curve Cryptography (ECC), computational problems in ECC and ECC-based public key certificate generation based on PKIX-X.509. In section 3, the ECC-based initial secret key negotiation protocol usable for IKE implementation is proposed. The security analysis of our proposed protocol against different related attacks is given in section 4. Finally, section 5 concludes the paper.

II. Preliminaries

To facilitate understanding of our proposed scheme, the following articles on ECC cryptosystem are briefly introduced.

A. Elliptic Curve Cryptography (ECC)

The elliptic curve cryptosystem [22], [26] was initially proposed by Koblitz [23] and then Miller [24] in 1985 to design public key cryptosystem and presently, it becomes an integral part of the modern cryptography.

Let *E/Fp* denotes an elliptic curve *E* over a prime finite field *Fp*, can be defined by the following equation

$$y^2 = x^3 + ax + b, \qquad - \ - \ - (1)$$

where, $\mathrm{a, b} \in \mathrm{F}_p$ and the discriminate $D = 4a^3 + 27b^2 \neq 0$

The points on *E/Fp* together with an extra point *O* called the point at infinity used for additive identity form an additive group *A* as

$$A = \{(x, y) : x, y \in F_p,\ E(x, y) = 0\} \cup \{0\} \ - \ - - (2)$$

Let the order of *A* be *n*, which is very large and it can be defined as *n* × *G mod q* = *O*, where *G* is the generator of *A*. The *A* be a cyclic additive group under the point addition ''+'' defined as follows: *P* + *O* = *P*, where $P \in A$.

The scalar point multiplication over *A* can be defined as

$$tP = P + P + \cdots + P \text{ (t times)} - \ - \ - (3)$$

If $P, Q \in A$, the addition *P* + *Q* be a point -*R* (whose inverse is *R* with only changing the sign of *y* coordinate value and lies on the curve) on the E/F$_P$ such that all the points *P, Q* and –*R* lie on the straight line, i.e., the straight line cuts the curve at *P, Q* and –*R* points. Note that if *P* = *Q*, it becomes a tangent at *P* or *Q* that intersects the curve at the point –*R*.

The security strength of the ECC lies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) [22]-[24] and it provides same level of security of RSA with less bit-size key. An overview of ECC is given below:

B. Computational Problems

Similar to the DH problem (known as discrete logarithm problem), some computational problems on ECC are defined below, where it is assumed that they have not any polynomial time algorithm.

- *Elliptic Curve Discrete Logarithm Problem (ECDLP)*

Given $Q, R \in A$, find an integer $k \in F_p^*$ such that *R*=*k.Q*.

- *Computational Diffie-Hellman Assumption (CDHA)*

Given $P, xP, yP \in A$, it is hard to compute $xyP \in A$.

- *Decisional Diffie–Hellman Problem (DDHP)*

Given $P, aP, bP, cP \in G$ for any $a, b, c \in F_p^*$, decide whether or not *cP* = *abP*.

C. ECC Based Certificate

As stated earlier, the ECC based certificate has been standardized by IETE as PKIX-X.509, which is almost similar to another standard X.509 with a main difference of using ECC based public key signed by the ECDSA. A simple ECC-based X.509 certificate format [25] to combine user's identity and the ECC-based public key proposed by ITU (International Telecommunication Union) is described in "Fig. 1".

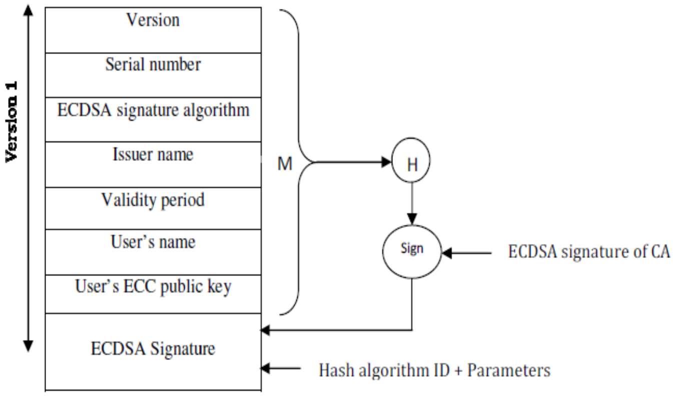


Fig.1. ECC-based X.509 Certificate Format

III. Proposed ECC based Phase I of IKE protocol

In this section, the proposed initial secrecy establishment scheme required for IKE protocol is discussed. The notations used are given below.

A. *Notations Used*

• *p, n*	Two large prime numbers;
• *Fp*	A finite field;
• *E*	An elliptic curve defined on *Fp* with prime order *n*;
• *G*	The group of elliptic curve points on *E*;
• *P*	A point on elliptic curve *E* with order *n*;
• *H (_)*	One-way hash function (e.g. SHA1, MD5);
• *I*	Initiator;
• *R*	Responder;
• *HDR*	ISAKMP-Header;
• SA_{PROP}	Security association proposal of *I*;
• SA_{SELEC}	Security association selected by *R*;
• ID_I	Identity of initiator *I*;
• ID_R	Identity of responder *R*;
• CA_I	Public key certificate of initiator *I*;
• CA_R	Public key certificate of responder *R*;
• (s_I, V_I)	*I*'s private/public key pair, where $V_I = s_I P$;
• (s_R, V_R)	*R*'s private/public key pair, where $V_R = s_R P$;

B. *The Proposed Protocol*

The initiator *I* and the responder *R* initially collect ECC based public key certificate from CA and start to exchange five messages for negotiation of initial secrecy as shown in "Fig. 2". Note that each certificate contains a user's public key and the signature of the CA over the hash value of the public key, user-ID, issue-date, issuer-name etc using ECDSA algorithm.

The messages *1* to *3* as shown are the parameters for negotiation as well as the mutual authentication of *I* and *R* through ECC-based public key certificate, where the messages are preceded by *HDR*, the standard ISAKMP message format that contains the information required by the protocol to maintain state, process payloads, and possibly to prevent *replay attack* and *denial-of-service attack*. The ISAKMP-Header format contains the following fields:

- Initiator's Cookie (8 bytes)
- Responder's Cookie (8 bytes)
- Next Payload (1 byte)
- Major Version (4 bits)
- Minor Version (4 bits)
- Exchange Type (1 bytes)
- Flags (1 byte)
- Message ID (4 bytes)
- Length (4 bytes)

The cookie of either initiator (C_I) or responder (C_R) may be formed using the following information as subfields:

- Hash value of the IP address, port number, and protocols
- A secret random number known to the initiator (or responder), and finally
- A timestamp

The initiator and responder in messages *1* and *2* only send their respective cookies, however, in subsequent message headers include both the cookies < C_I, C_R>.

The SA_{PROP}, a list of cryptographic proposals, is sent by the initiator to the responder for negotiation and SA_{SELEC}, the cryptographic protocols, is selected by the responder from the list sent by the initiator. If necessary, the responder can reject the entire list sent by *I* and sends back an error message in reply. The proposed protocol comprises five steps, the details of which are given below along with a flow diagram as shown in "Fig. 2".

Step 1: *Initiator → Responder: HDR,* SA_{PROP}, CA_I

The initiator generates cookie C_I and send his SA proposal SA_{PROP} and ECC-based public key certificate CA_I.

Step2: *Responder → Initiator: HDR,* SA_{SELEC}, CA_R, $E_{K_X}(ID_R, U_R)$

The responder selects the cryptographic proposal SA_{SELEC} from SA_{PROP}, generates his cookie C_R and calculates the following parameters:

(i) A random number k_R from [1, n-1],
(ii) A secret key K_x for symmetric encryption using $K = s_R \cdot V_I = s_R \cdot s_I \cdot P = (K_X, K_Y)$,
(iii) A private value $U_R = k_R \cdot V_R = (k_R \cdot s_R) \cdot P$, which is then encrypted with his identity using the symmetric key K_X as obtained in (ii).

The responder now sends SA_{SELEC}, ECC certificate and the encrypted (ID_R, U_R) to the initiator.

Step 3: *Initiator → Responder: HDR,* $E_{K_X}(ID_I, U_I)$

The initiator similarly computes the following:

(i) The decryption key K_X by using $K = s_I \cdot V_R = s_I \cdot s_R \cdot P = (K_X, K_Y)$,
(ii) Obtain U_R by decrypting $E_{K_X}(ID_R, U_R)$,
(iii) Selects a random number k_I from [1, n-1],
(iv) Obtain a private value $U_I = (k_I \cdot s_I) \cdot P$, which is then encrypted using K_X as obtained in (i).

The above values are then sent to the responder as shown in "Fig. 2".

Step 4: *Responder → Initiator: HDR,* $E_{SK_X}(HASH\text{-}R)$

After receiving the message, the responder decrypts it using K_X, compares the received ID_I with ID_I stored in initiator's certificate and the cookies as well. If everything is alright, he then calculates the initial secret key to be required in phase II of the IKE as

$$SK = (k_R \cdot s_R) \cdot U_I = k_R \cdot s_R \cdot k_I \cdot s_I \cdot P = (SK_X, SK_Y)$$

The *HASH-R* as given below is encrypted with the key SK_x, the x-coordinate of the secret *SK* and sends to the initiator.

$$HASH - R = H(U_I \parallel U_R \parallel C_I \parallel C_R \parallel SA_I \parallel ID_R)$$

Step 5: *Initiator → Responder: HDR,* E_{SK_X} *(HASH-I)*

The Initiator after some comparisons as mentioned in *step 4,* computes the same secret key as

$$SK = (k_I \cdot s_I) \cdot U_R = k_I \cdot s_I \cdot k_R \cdot s_R \cdot P = (SK_X, SK_Y)$$

The *HASH-R* value send by responder is obtained by decrypting E_{SK_X} (*HASH-R*) using the x-coordinate of *SK* and compared with its own calculated HASH-R. If match is found, the responder becomes authenticated to the initiator; otherwise the protocol execution is terminated.

Finally, the initiator generates the message *HASH-I* as given below, and sends it to the responder after encrypting with the SK_x, the x-coordinate of the point *SK*.

$$HASH - I = H(U_I \parallel U_R \parallel C_I \parallel C_R \parallel SA_I \parallel ID_I)$$

On receiving, the responder decrypts it and compares with his own computed *HASH-I.* If it passes, the responder is assured that the initiator is authenticated and the valid common secret key is negotiated. If not, the responder terminates the execution and closes SA session.

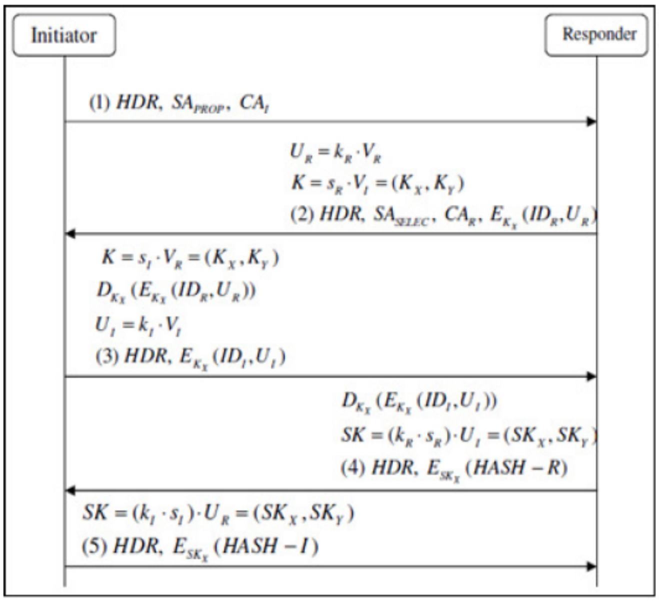

Initiator
Responder
(1) HDR, SA_{PROP}, CA_I
$U_R = k_R \cdot V_R$
$K = s_R \cdot V_I = (K_X, K_Y)$
(2) HDR, SA_{SELEC}, CA_R, $E_{K_X}(ID_R, U_R)$
$K = s_I \cdot V_R = (K_X, K_Y)$
$D_{K_X}(E_{K_X}(ID_R, U_R))$
$U_I = k_I \cdot V_I$
(3) HDR, $E_{K_X}(ID_I, U_I)$
$D_{K_X}(E_{K_X}(ID_I, U_I))$
$SK = (k_R \cdot s_R) \cdot U_I = (SK_X, SK_Y)$
(4) HDR, $E_{SK_X}(HASH - R)$
$SK = (k_I \cdot s_I) \cdot U_R = (SK_X, SK_Y)$
(5) HDR, $E_{SK_X}(HASH - I)$

Fig. 2. Proposed Initial Secret Key Negotiation Protocol

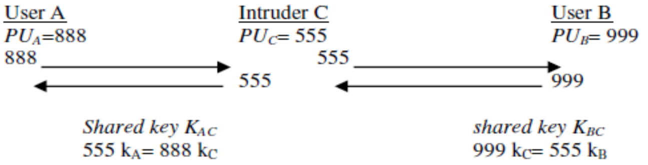
It can be seen that the proposed scheme uses four random numbers in establishing the common secret *SK* with users' authentication and thus, according to [32], it can be concluded that the proposed method is well secured as a shared secret with only one random number is assumed to be compromised. The details of the security analysis of the present work against different attacks are discussed in the next section.

IV. Security Analysis of the Proposed Protocol

The in-depth security analysis of the proposed scheme is given in this section. For this, a number of attacks are considered and the protection against each of them is given below.

A. *Man-in-the-middle Attack*

Suppose two users *A* and *B* are negotiating a secret key over an open channel in a session and an intruder *C* comes in between and reacts in the way as given below.



As shown, two shared keys are generated and they are— (1) Between *A* and *C* $K_{AC} = 555.\ k_A = 888.\ k_C$, where k_A and k_C are the private keys of the *A* and *C* respectively, and (2) Between *B* and *C* $K_{BC} = 999.k_C = 555.k_B$, where k_B is the private key of *B*. As a result, any encrypted message from *A* to *B* is easily interpreted by *C*, which defined as *man-in-the-middle* attack.

The proposed protocol is free from man-in-middle attack since we have used entity's public key certificate signed by CA. Each certificate binds entity's identity, its public key and other relevant information. Before using an entity's public key, the certificate must be validated by the user. The messages *2* to *3* as shown in "Fig. 2" contain the identity of responder and initiator which are encrypted using ECDH symmetric key generated by either initiator's private key and responder's public key, or responder's private key and initiator's public key. The decrypted identity is compared with the identity stored in the certificate. If match is not found, initiator/responder terminates the process. Thus, our proposed protocol is free from the man-in-the-middle attack.

B. *Replay Attack*

Replay attack is an illegal action by which an attacker may take off the legal client by reusing the information obtained from a previous communication between legal entities. In our proposed protocol, an ISAKMP-Header is used in every message exchanged, which contains the initiator/responder's cookie (C_I/C_R) with other relevant information. The C_I/C_R is the result of hashing a unique identifier of the peer (such as IP address, port number, and protocol), a secret random number known to the party and *a timestamp*. When initiator/responder sends a message to responder/initiator, it includes its best estimate of the time in cookie. The responder/initiator only accepts messages for which the *timestamp* is within a reasonable tolerance. Now suppose an attacker wants to impersonate the initiator/responder by replaying the previous message, the initiator/responder can detect it by comparing with the previous cookie which includes the *timestamp* and terminates the execution as the timestamps are mismatched. Thus our proposed protocol prevents the replay attack.

C. Denial-of-service (DoS) Attack

In our proposed protocol, an ISAKMP-Header *HDR* precedes every message and contains cookies of initiator (C_I) and responder (C_R) that can prevent the *denial-of-service attack*. As the same cookies are accompanied with every massage in our proposed protocol, the DoS is never possible. Because a repeated and unchanged in every message. If an attacker acts as an initiator using a bogus IP address, he does not receive the reply message and thus, he is not capable to return the same cookie to the sender. Thus the denial-of-service attack is not possible.

D. Impersonation Attack

The proposed protocol is free from impersonation attack. If an attacker makes an effort to impersonate the initiator/responder to exchange a session key, then it is impossible for the attacker to figure out U_R, U_I from messages *2* and *3* since these are encrypted by a symmetric secret key *Kx*, known to the initiator and the responder. Then the attacker replies with the wrong message in step *4* and *5* which direct the termination of the process. Thus our proposed protocol prevents the impersonation attack.

E. Perfect Forward/Backward Secrecy

The Perfect forward/backward secrecy is the property that the disclosure of the initiator/responder's private key (or any session key) does not compromise the secret key negotiated from earlier/latter runs. In our proposed protocol, the initiator/responder's private key is used for authentication purpose whereas the secret key negotiation is done by the initiator/responder's secret random number (k_I/k_R). Now if the initiator/responder's private key is known to an attacker, and he computes $K = s_I \cdot V_R = s_I \cdot s_R \cdot P = s_R \cdot V_I = (K_X, K_Y)$, U_R and U_I from the messages *2* and *3*, even then he cannot derive the session key $SK = k_I \cdot s_I \cdot k_R \cdot s_R \cdot P = (SK_X, SK_Y)$, because the attacker tries to compute the session key *SK* from the pair $(U_I, U_R) = (k_I \cdot s_I \cdot P, k_R \cdot s_R \cdot P)$ directly, which is impossible due to difficulties of Computational Diffie–Hellman Assumption (CDHA). Also if any session key is leaked, the attacker cannot derive any other session keys or the current one. Hence, the proposed protocol holds these properties.

F. Known-Key Security

The proposed protocol results in a unique shared session key after completion of each negotiation. The compromise of one shared session key (k_X) in one negotiation is never compromised with the shared session key (SK_X) agreed on any other negotiation session.

G. Identification Privacy

The proposed protocol does not disclose the initiator/responder's identification; since it is encrypted using ECDH shared session key and CA signed public key certificate is used. In message *2* and *3* of our proposed protocol ("Fig. 2"), the responder/ initiator's identification is only verified by initiator/responder respectively.

H. Key Control

In our proposed protocol, no pre-shared secret is used to calculate new shared session key, thus the key control of the initiator/responder is supported.

I. Explicit Key Confirmation

The explicit key confirmation means that before using the key to encrypt confidential data, one communication party has to confirm that the other party has actually computed the correct shared session key. In message 4 and 5 of "Fig. 2", the responder/initiator makes a message digest and sends it to each other for verification. This supports the explicit key confirmation in our proposed protocol.

J. Efficiency

The proposed ECC based protocol is more efficient than the existing protocols used for initial security association of IKE protocol. It consists of five message exchanges, which is one less than the main mode of phase I that uses six messages. The first three messages of our proposed scheme perform the mutual authentication of the initiator/responder and the shared secret session key is established in last two messages. Further, it requires much lesser key length, computation- and communication-cost than any RSA-based and/or other schemes for providing same level of security.

The proposed ECC based secret key negotiation scheme for providing security association to phase II of IKE not only supports less computation- and communication-cost, but also protects all relevant attacks, and in this regard, a theorem is given below.

Theorem-1: Proposed ECC based initial secrecy negotiation protocol for IKE is efficient and secured.

Proof: In order to proof the theorem, the following points regarding the processing costs and the security aspects of the proposed scheme may be followed.

- *For public challenges*, the existing RSA based IKE protocols use Diffie–Hellman key exchange protocol in which the required public challenges generated by using expensive modular exponential operation are $V_I = g^{s_I} \bmod n$ and $V_R = g^{s_R} \bmod n$, where the size of the modulus *n* should be at least 1024 bits length for its security. On the other hand, the public challenges in ECC $V_I = s_I \cdot P$ and $V_R = s_R \cdot P$ require 160 bits only for comparable security. Thus, the computation cost is also reduced in the ECC based technique.

- *For encryption/decryption,* the existing protocols apply RSA-based public key encryption/decryption technique which is due to modular exponentiation operation, much slower than the scalar point multiplication used in ECC. This is because the modular exponentiation is used over 1024-bit discrete logarithm problem (DLP) in RSA, whereas the ECC requires point multiplication using 160-bit ECDLP. The processing speed in ECC is further enhanced by incorporating symmetric encryption rather than the RSA-based public key encryption as used in existing techniques.

Therefore, the proposed protocol reduces the computation cost over the RSA-based techniques.

- *In terms of operation,* the existing RSA based IKE protocols require six messages exchange, implicit key confirmation, long 1024-bit key size, longer negotiation time, whereas in ECC, it has five messages exchange, explicit key confirmation, short 160-bit key size, shorter negotiation time. Therefore, the proposed protocol have low communication cost, faster processing speed and low network traffic.

- *In terms of security,* some relevant cryptographic attacks of IKE like man-in-the-middle attack, replay attack, denial-of-service attack, impersonation attack etc are discussed. It has been shown that the proposed ECC based scheme prevents all these attacks.

Hence, the theorem is verified. □

V. CONCLUSION

A new initial secret key negotiation protocol for implementation of phase I of the IKE is proposed in this paper, where the ECC-based public key certificate for the users' authentication is used. The scheme follows the main mode approach and instead of six, it completes the negotiation through the exchanges of five messages. The main advantages of the proposed scheme over the RSA-based certificate are the requirement of less computation cost, high processing speed, low network traffic and comparable security even using small secret key-size and thus suitable for efficient implementation. The security analysis of the proposed method against a number of attacks is given and it is found that all the attacks are well protected.

REFERENCES

[1] J. Zhou, "Further analysis of the Internet key exchange protocol", Computer Communications, 23, 2000, pp. 1606-1612.

[2] B. A. Forouzan, "Cryptography and Network Security", Special Indian Edition 2007, TMH, pp. 563-588.

[3] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[4] R. Canetti, H. Krawczyk. "Analysis of key exchange protocols and their use for building secure channels", In Proc. Of Advances in Cryptology –Eurocrypt'01, Springer-Verlag, Berlin/Heidelberg, 2001, 453-474, LNCS.

[5] ZHU Jian- ming, MA Jian-feng. An Internet Key Exchange Protocol Based on Public Key Infrastructure, Journal of Shanghai University (English Edition), 2004.[1] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[6] Radia Perlman and Charlie Kaufman, "Key Exchange in IPSec: Analysis of IKE," *IEEE Internet Computing*, November-December, 2000, pp. 50-56.

[7] Michael S. Borella, "Methods and Protocols for Secure Key Negotiation Using IKE," *IEEE Networks*, July-August 2000, pp. 18-29.

[8] Niklas Hallqvist and Angelos D. Keromytis, "Implementing Internet Key Exchange (IKE)," *USENIX Annual Conference*, June 2000.

[9] Charlie Kaufman, "The Internet Key Exchange (IKEv2) Protocol," IETF draft-ietf-ipsec-ikev2-17, September 2004.

[10] Ajmal S. Mian and Ashraf Masood, "Arcanum: A Secure and Efficient Key Exchange Protocol for the Internet," *IEEE Proc. of the Intl. Conf. on Information Technology: Coding and Computing*, Vol. 1, 2004, pp.17-21.

[11] Haddad H., Berenjkoub M. and Gazor S., "A Proposed Protocol for Internet Key Exchange (IKE)," *Electrical and Computer Engineering*, Canadian Conf., May 2004.

[12] Haddad H. and Mirmohamadi H., "Comparative evaluation of successor protocols to Internet key exchange IKE)," *Proceedings of the IEEE Intl. Conf. on Industrial Informatics*, August 2005, pp.692-696.

[13] H. Orman, "The OAKLEY Key Determination Protocol," RFC 2412, 1998.

[14] D. Maughan et al., "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, 1998.

[15] S. Hirose and K. Matsuura, "Key Agreement Protocols Resistant to a Denial-to-Service," *IEICE Trans. On Information and Systems*, April 2001, pp. 477-484.

[16] Ming-Yang Su and Jia-Feng Chang, "An efficient and secured internet key exchange protocol design", proceedings of the fifth annual conference on Communication Networks and Services Research (CNSR'07), 2007, pp. 184-192.

[17] H. Fereidooni, H. Taheri and M. Mahramian, "A new authentication and key exchange protocol for insecure networks", proceedings of the fifth international conference on Wireless Communication, Networking and Mobile Computing (WiCom'09), 2009, pp. 1-4.

[18] V. Nagalakshmi and I. Rameshbabu, "A protocol for internet key exchange (IKE) using public encryption key and public signature key", International Journal of Computer Science and Network Security, vol. 7, No. 7, July 2007, pp. 342-346.

[19] V. Nagalakshmi, I. Rameshbabu and P.S. Avadhani, "Modified protocols for internet key exchange (IKE) using public encryption key and signature keys", proceedings of the eighth international conference on Information Technology: New Generations, 2011, pp. 376-381.

[20] W Diffie, ME Hellman. "New directions in cryptography". IEEE Transactions on Information Theory, 1976, 22(6):644–654.

[21] Blake-Wilson S and Menezes A, "Authenticated Diffie-Hellman Key Agreement Protocols," *Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography* (SAC'98); *Lecture Notes in Computer Science* 1556, 1999, pp. 339-361.

[22] D Hankerson , A Menezes, S Vanstone. "Guide to elliptic curve cryptography". 2004, Springer-Verlag, New York, USA.

[23] N Koblitz. "Elliptic Curve Cryptosystem". Journal of mathematics computation, Janaury 1987, 48(177):203- 2009.

[24] V Miller. "Use of elliptic curves in cryptography". Advances in Cryptology-CRYPTO, 85 (LNCS 218), 1985:417–426.

[25] Q. Dang, S. Santesson, K. Moriarty, D. Brown and T. Polk, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", RFC 5758, January 2010.

[26] SK Hafizul Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography", Mathematical and Computer Modeling (2011), doi:10.1016/j.mcm.2011.07.001, in press.

[27] J. Weise, 'Public Key Infrastructure Overview', Sun PSSM Global Security Practice, Sun Blue Prints™ Online - August 2001.

[28] 'Introduction to Public Key Technology and the Federal PKI Infrastructure', National Institute of Standards and Technology, 26th Feb, 2001.

[29] W. Stallings, 'Cryptography and Network Security: Principles and Practices', 4th Edition, Prentice Hall, 2009 International Edition, pp. 420-430.

[30] W. Polk, R. Housley and L. Bassham, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[31] J. Schaad, B. Kaliski and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, June 2005.

[32] G. P. Biswas, "Establishment of Authenticated Secret Session Keys Using Digital Signature Standard", Information Security Journal: A Global Prospective, 2011, 20(1): 09-16.