Two-party Authenticated Multiple-key Agreement Based on Elliptic Curve Discrete Logarithm Problem[⋆]

Li-Chin Huang[1] and Min-Shiang Hwang[2]
Department of Computer Science and Engineering[1],
National Chung Hsing University
phd9406@cs.nchu.edu.tw
Department of Computer Science and Information Engineering[2],
Asia University
mshwang@asia.edu.tw
Corresponding author: Prof. Min-Shiang Hwang

Abstract

In this paper, we propose a protocol to generate n^2 keys in one session under the assumption of the intractability of the elliptic curve discrete logarithm problem and MQV protocol. Our protocol has the advantage of requiring less computing time compared with other protocols. Therefore, it is easy to apply in resource-constrained key agreement such as wireless sensor networks, mobile Ad-hoc networks, and cell phones which are severely constrained processor, battery, and memory.

Key words: *Cryptography, Elliptic curve discrete logarithm, key agreement, key authentication, resource constrained, wireless networks*

1 Introduction

Cryptography algorithms are classified two categories: private-key (sysmmetric) and public-key (asymmetric). Over Internet, the security protocols (e.g. SSL, IPSec) utilize a public-key cryptosystem to exchange private keys and then adopt faster private-key algorithms to ensure confidentiality of streaming data. In private-key algorithms, communicating parties share a common private key to transform the original message into a ciphered message. The ciphered message is decrypted by the same private key at another side. Public-key systems need not exchange keys because keys are public to all. Nevertheless, the ciphertext can still be decrypted utilize the receiver's private key. Currently, the security applied public-key cryptosystems is based on the intractability of certain mathematical problems such as presuming difficulty of factoring large integers or depending upon a certain hard problem in discrete logarithms. Rivest, Shamir, and Adleman devised a trapdoor one-way function utilizing elementary number theory. In other words, the security of RSA cryptography is based on multiplication of two large prime numbers p and q to get a composite number N = qp depending on the presumed difficulty of factoring large integers. Based on the presumed difficulty of inverting the function $x \mapsto g^x$ in a finite field, ElGamal had proposed an alternative to RSA encryption. For public-key cryptography, the ElGamal encryption system is an asymmetric key encryption algorithm based on the Diffie-Hellman key exchange [10] published by Taher Elgamal in 1984. ElGamal encryption [11] is adopted by the GNU Privacy Guard and other cryptosystems.

In 1984, Hendrik Lenstra developed a new method for factoring large integers using elliptic curves in cryptography. The elliptic curve discrete logarithm problem is a more difficult problem than integer factorization substantially. In elliptic curve cryptography (ECC), it is supposed that finding the

[⋆]Partial results of this paper have been presented in ACN 2012.

discrete logarithm of a random elliptic curve element with respect to a publicly known base point is feasible. Although RSA is well established, the ECC is still more commercial importance and has attracted attention because of a smaller key size, reducing storage, low on CPU consumption, and transmission requirements. On transmission requirements, an elliptic curve group may provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key such as a 256 bit ECC public key should provide comparable security to a 3072 bits RSA public key. Due to the favorable characteristics, ECC has been incorporated into two important public-key cryptography standards, FIPS 186-2 [30] and IEEE-P1363 [16]. These standards specify how to utilize elliptic curves over prime fields GF(p) and binary fields $GF(2^m)$. Furthermore, they recommended curves have well-studied properties to resistant known attacks. In Table 1 [28], the comparison of the energy consumed by the three federal information processing standard (FIPS)-approved asymmetric algorithms for generating and verifying signatures in security protocols: RSA, DSA, and ECDSA. From Table 1 [33], we can observe the key size growth for various private and public-key cryptosystems with correspondingly larger key. The energy values are reported for three main steps to associated with digital signature algorithms: key generation,signature creation (Sign), and signature verification (Verify). Suppose a priori generation of the parameters utilized in the key generation process for resource-constrained devices. Obviously, 163-bit ECDSA is energy-efficient compared to 1024-bit DSA. Furthermore, 163-bit ECDSA and 1024-bit RSA digital signature algorithms have complementary energy costs. In the sign operation, the private key is applied which posses the same size as the modulus. And the much smaller public key is adopted for verify operation. In Table 2, the comparison of the standard algorithms adopted for key exchange, Diffie-Hellman (DH) and elliptic curve Diffie-Hellman (ECDH). Apparently, a 163-bit ECDH will consume much lesser energy than a 1024-bit DH key exchange. When symmetric key sizes increase, the required key sizes of RSA and Diffie-Hellman will increase at a more rapidly rate than the required key sizes for elliptic curve cryptosystems. An elliptic curve system provides more security per bit increase in key size than either RSA or Diffie-Hellman public key systems [10] shown in Table 3. In computation, Elliptic curve cryptosystems are more efficient than RSA and Diffie-Hellman. The ratio of computation between DH and EC for every key size shown in Table 4. In key size and cost, ECC is apparently superior to RSA demonstrated on Table 5. For getting a similar security level, ECC requires much shorter key lengths. For instance, 160-bit ECC-key reaches the same level of security as 1024-bit RSA key. The shorter key size of ECC will suitable for computing in the future because more powerful computers will need more longer key length to obtain higher security level.

In 1976, Diffie and Hellman [10] proposed the first practical solution to key agreement problem for two parties to establish a session key, which can be applied to provide security or data integrity for later communications between the two parties. Unfortunately, the original Diffie-Hellman protocol suffers from the "main-in-the-middle" attack because of lack of authentication between two communication parties. Over the years, many solutions [5, 6, 8, 14, 15, 21, 24, 31, 32] have been developed to solve this problem. In 1995, Menezes, Qu and Vanstone [26] developed a well-known key agreement protocol applying a digital signature to sign the Diffie-Hellman public keys without using any hash function. MQV (Menezes-Qu-Vanstone) had been chosen by the NSA as the key exchange mechanism underlying "the next generation cryptography to protect US government information". The protocol was also adopted as IEEE P1363-2000 standard [1].

Key management is one of the most challenging security issues in wireless sensor networks, mobile Ad-hoc networks, smart cards, personal digital assistants (PDAs), and cell phones, which are severely constrained in the resources such as processor, battery, and memory. Therefore, the energy-efficient security will be a critical issue. In order to establish multiple common secret keys between two parties efficiently, in 1998 Harn and Lin [12] developed an authenticated key agreement protocol based on the MQV protocol without using any one-way hash function. In Harn-Lin protocol, two parties can authenticate each other and establish n^2 common session keys. The Harn-Lin protocol sets up the limit that only $(n^2 - 1)$ common session keys can be used to avoid the known key attack [29]. Later, Yen and Joye [41] pointed out a security problem and a solution in the Harn-Lin protocol. This security problem let an attacker impersonates one party to generate common session secret keys with another party by forging a signature message modified by the previous one, called a forgery

Table 1. Energy Cost Digital Signature Algorithms

Algorithm	Key size (bits)	Key Generation (mJ)	Sign (mJ)	Verify (mJ)
RSA	1024	270.13	546.50	15.97
DSA	1024	293.20	313.60	338.02
ECDSA	163	226.65	134.20	196.23
ECDSA	193	281.65	166.75	243.84
ECDSA	233	323.30	191.37	279.82
ECDSA	283	504.96	298.86	437.00
ECDSA	409	1034.92	611.40	895.98

Table 2. Energy Cost of Key Exchange Algorithms

Algorithm	Key size (bits)	Key Generation (mJ)	Key Exchange (mJ)
DH	1024	875.96	1046.5
ECDH	163	276.70	163.5
DH	512	202.56	159.6

attack. However, according to Wu et al. [38], the Yen-Joye protocol cannot withstand the same attack from which Harn-Lin protocol suffers. Afterward, we proposed an improved protocol [14] to improve the Yen-Joye protocol. In 2001, Harn and Lin [13] then modified the signature in which the equation is signed [12] to prevent the forgery attack. But still only (n^2-1) common session keys are allowed to be used in their protocol. In 2002, Tseng [36] proposed a robust protocol that even two parties use all the n^2 common session keys, the known-key attack can be avoided. Until now, there are many schemes had been proposed [4, 7, 23, 25, 40].

In this paper, we propose a protocol based on the elliptic curve discrete logarithm problem to generate n^2 common session-keys in one session, and all the keys can be used to withstand the known-key attack, replay attack, forgery attack, key-compromise impersonation, and key control.

2 Elliptic Curve Cryptography Based on Group Theory

The elliptic curve crypto system (ECCS) is a crypto-algorithm method base on a discrete logarithm problem (DLP) [18] over the points on an elliptic curve. Recently, ECC [20, 33, 34, 39, 42] has become important cryptography applying to many resource constrained environments in the Internet such as smart cards, mobile ad hoc networks and communication devices because of smaller key sizes, less memory, fewer bandwidth, low communication cost, and faster implementation. Based on ECCS, abelian groups are extensively utilized in cryptography and also have commutative or symmetric property. Because of the smaller key size, the abelian group of point of an elliptic curve is much smaller in size at the same time maintains the same level of security. Closure is a fundamental property of groups. Thus, the modulo (n) operation allow the domain to have finite number of members. This property will ensure that the problem is solvable for the valid receiver, as well as for the problem to be hard.

Table 4. Computation Costs

Security Level (bits)	Ratio of DH:EC
112	6:1
128	10:1
192	32:1
256	64:1

Table 5. Key Sizes in bits

ECC Key Size	RSA Key Size	Key-Size Ratio
163	1024	1:6
256	3072	1:12
384	7680	1:20
512	15360	1:30

Table 3. Comparison of Key Sizes in Three Difference Methods

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic curve Key Size (bits)
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

A set of ECC [3, 37] mathematical operations is defined over the elliptic curve E:

$$y^2 = x^3 + ax + b$$

where $4a^3 + 27b^2 \neq 0$. Various values of a and b will provide different elliptic curves. Then the addition law on E describes as follows:

1. $Identity: \ P + \mathcal{O} = \mathcal{O} + P = P \ for \ all \ P \in E.$
2. $Inverse: \ P + (-P) = \mathcal{O} \ for \ all \ P \in E.$
3. $Associative: (P + Q) + R = P + (Q + R) \ for \ all \ P, Q, R \in E.$
4. $Commutative: P + Q = Q + P \ for \ all \ P, Q \in E.$

2.1 Elliptic Curve Discrete Logarithm Problem

The security of ECC relies on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is infeasible to find k computationally if k is enough large. The scaler k is the discrete logarithm of Q to base P. Therefore, point multiplication is a principle operation in ECC. In other words, multiplication of a scalar k with any point P on the curve will find another point Q on the curve.

2.2 Elliptic Diffie-Helman Key Exchange

ECDH [2, 9] is a key agreement protocol to establish a share secret key between two parties. Alice and Bob agree to adopt a particular elliptic curve $E(\mathbb{F}_p)$. and a particular point $P \in E(\mathbb{F}_p)$. Alice selects a secret integer n_A and compute the associated multiples $Q_A = n_A P$. Also, Bob selects a secret integer n_B and compute the associated multiples $Q_B = n_B P$. Then, they exchange the values of Q_A and Q_B. Alice utilizes her secret multiplier to compute $n_A Q_B$. And Bob similarly computes $n_B Q_A$. Now, they have the shared secret value

$$n_A Q_B = (n_A n_B)P = n_B Q_A.$$

3 Review Tseng's Protocol

In this section, let us briefly review Tseng's protocol that can establish n^2 common session keys between two parties. The protocol is divided into two phases: the initiation phase and the multiple-key agreement phase. To give a simple example, let's assume that Bob and Alice want to establish four common session keys by using 2 short-term secret keys. They are required to go through the following processes:

The initiation phase: In the Diffie-Hellman scheme, the system publishes a large prime number p. Bob and Alice select their random numbers x_A and x_B and compute the corresponding long-term public keys $y_A = g^{x_A} \bmod p$ and $y_B = g^{x_B} \bmod p$, respectively.

The multiple-key agreement phase:

1. Alice selects two random short-term secret keys k_{A_1} and k_{A_2}, where $k_A = k_{A_1} + k_{A_2} \bmod q$, and then calculates the corresponding short-term public keys $r_A = g^{k_A} \bmod p$, $r_{A_1} = (y_B)^{k_{A_1}} \bmod p$ and $r_{A_2} = (y_B)^{k_{A_2}} \bmod p$. After obtaining the signature s_A based on the equation $s_A r_A = x_A - r_{A_1} k_A \bmod q$. Alice sends $\{r_{A_1},\, r_{A_2},\, s_A,\, Cert(y_A)\}$ to Bob, where $Cert(y_A)$ is a certificate for the public key signed by a trustworthy party.
2. In the same way as Alice does, Bob also generates $k_{B_1}, k_{B_2}, r_{B_1}, r_{B_2}$ and s_B and sends $\{r_{B_1},\, r_{B_2},\, s_B,\, Cert(y_B)\}$ to Alice.
3. Alice verifies the messages $\{r_{B_1},\, r_{B_2},\, s_B,\, Cert(y_B)\}$ from Bob, and furthermore checks the following equation:

$$y_B = (r_B)^{r_{B1}} g^{s_B r_B} \bmod p, \tag{1}$$

where $r_B = r_{b1} r_{b2} \bmod p$, $r_{b_1} = (r_{B_1})^{x_A^{-1}} \bmod p$ and $r_{b_2} = (r_{B_2})^{x_A^{-1}} \bmod p$. If the above equation is correct, Alice will compute four common session keys as follows:

$$K_1 = r_{b1}^{k_{A1}} \bmod p,$$
$$K_2 = r_{b1}^{k_{A2}} \bmod p,$$
$$K_3 = r_{b2}^{k_{A1}} \bmod p,$$

and

$$K_4 = r_{b2}^{k_{A2}} \bmod p.$$

Just like Alice, Bob also verifies the authenticated messages and generates four common secret keys: $K_1 = r_{a1}^{k_{B1}} \bmod p$, $K_2 = r_{a2}^{k_{B1}} \bmod p$, $K_3 = r_{a1}^{k_{B1}} \bmod p$ and $K_4 = r_{a2}^{k_{B2}} \bmod p$.

4 The Proposed Protocol

In this section, we shall propose a more efficient protocol to establish n^2 common session keys between two parties based on the elliptic curve discrete logarithm problem. The protocol is composed of two phases: the initiation phase and the multiple-key agreement phase. Let's assume that Bob and Alice want to establish four common session keys by using 2 short-term keys. The following two phases enable Alice and Bob to authenticate each other and to generate multiple common secret keys.

Generation of pairing parameters and key initiation: The system publicly chooses an elliptic curve E over a finite field $GF(q)$ and a base point G with order p [27]. Bob and Alice choose their secret key $k_A \in [1, p-1]$ and $k_B \in [1, p-1]$, and compute the corresponding long-term public keys $Q_A = k_A G$ and $Q_B = k_B G$, respectively.

The multiple-key agreement phase:

1. Alice chooses two short-term secret keys k_{a_1} and k_{a_2}, and then computes the corresponding short-term public keys $R_{A_1} = Q_B k_{a1} = k_{A_1} G$ and $R_{A_2} = Q_B k_{a2} = k_{A_2} G$. After getting the signature s_A based on the equation $s_A = (R_{A2})_x k_{A1}^{-1} [(Q_A)_x - k_A (R_{A1})_x] \bmod q$, Alice sends $\{R_{A_1},\, R_{A_2},\, s_A$, and $Cert(Q_A)\}$ to Bob, where $Cert(Q_A)$ is a certificate for the public key signed by a trustworthy party.

2. Just as Alice does, Bob also generates $k_{B_1}, k_{B_2}, R_{B_1}, R_{B_2}$ and s_B and sends $\{R_{B_1}$, R_{B_2}, s_B, and $Cert(Q_B)\}$ to Alice.
3. Alice verifies the authenticated messages $\{R_{B_1}$, R_{B_2}, s_B, and $Cert(Q_B)\}$ from Bob, and then checks the following equation:

$$(R_{B2})_x(R_{B1})_x Q_B + S_B R_{B1} = (R_{B2})_x(Q_B)_x G. \tag{2}$$

4. If the equation is correct, Alice computes the following equations:

$$\begin{aligned} R_{b1} &= R_{B1}k_A^{-1} = k_{b1}G \bmod p, \\ R_{b2} &= R_{B2}k_A^{-1} = k_{b2}G \bmod p. \end{aligned}$$

5. Alice generates four common session keys as follows:

$$\begin{aligned} K_1 &= R_{b1}k_{a1} \bmod p, \\ K_2 &= R_{b1}k_{a2} \bmod p, \\ K_3 &= R_{b2}k_{a1} \bmod p, \end{aligned}$$

and

$$K_4 = R_{b2}k_{a2} \bmod p.$$

Like Alice, Bob also verifies the authenticated messages and generates four common secret keys: $K_1 = R_{a1}k_{b1} \bmod p$, $K_2 = R_{a2}k_{b1} \bmod p$, $K_3 = R_{a1}k_{b2} \bmod p$, and $K_4 = R_{a2}k_{b2} \bmod p$.

5 Security Analysis

The security level of our protocol is based on the intractability of the elliptic curve discrete logarithm problem (ECDLP). Any adversary that intends to reveal a secret key k_i from its corresponding public key Q_i has to face the ECDLP. In the remainder of this section, several attacks will be raised and fought against to demonstrate the security of our protocol.

– **Known-key Attack:** We show that our protocol could resist the known-key attack even though two parties use the four common secret keys. In our protocol, the signatures for the two parties are shown as follows:

$$\begin{aligned} k_A &= (R_{A1})_x^{-1}(Q_A)_x - s_A(R_{A1})_x^{-1}(R_{A2})_x^{-1}k_{A1} \bmod q \\ k_B &= (R_{B1})_x^{-1}(Q_B)_x - s_B(R_{B1})_x^{-1}(R_{B2})_x^{-1}k_{B1} \bmod q. \end{aligned}$$

Thus, we have

$$\begin{aligned} k_A k_B G = {} & [(R_{A1})_x^{-1}(Q_A)_x - s_A(R_{A1})_x^{-1}(R_{A2})_x^{-1}k_{A1}] \\ & \times [(R_{B1})_x^{-1}(Q_B)_x - s_B(R_{B1})_x^{-1}(R_{B2})_x^{-1}k_{B1}]G \bmod p. \end{aligned}$$

From the above equation, if an intruder gets all common secret keys (K_1, K_2, K_3, and K_4), it is still very difficult for him to calculate $k_A k_B G$ by intercepting the transmitted message between the two parties, where the transmitted message involves $(R_{A1}, R_{A2}, R_{B1}, R_{B2}, s_A, s_B)$. The intruder cannot derive $(R_{a1}, R_{a2}, R_{b1}, R_{b2})$ from any transmitted message. The security is based on the intractability of ECDLP. Therefore, our protocol can withstand known-key attack [29] and allow two parties to establish n^2 common secret keys if they use n Diffie-Hellman's public-keys.

- **Replay Attack:** To resist the replay attack, our protocol uses short-term keys. The lifetime of the short-term keys (k_{a_i} and k_{b_i}, $i \in 1, 2, \ldots$) is only one session long, with a view of establishing n^2 keys. The two parties have to randomly choose new short-term keys again for the next session. When the intruder replays the previously intercepted message to Bob for masquerading as Alice, the request will be rejected because Bob will find that the message has been used previously.

- **Forgery Attack:** Assume that an intruder wants to impersonate Bob to establish the common session keys with Alice. The intruder forges the previously intercepted message (R_{B_1}, R_{B_2}, s_B, $Cert(Q_B)$) to (R'_{B_1}, R'_{B_2}, s'_B, $Cert(Q_B)$) and sends it to Alice, where (R'_{B_1}, R'_{B_2}, s'_B).

$$\begin{aligned} R'_{B_1} &= k'_{B_1} G \bmod p, \\ R'_{B_2} &= k'_{B_2} G \bmod p, \\ s'_B &= (R_{B2})x' - r'_{B_1} k'_{B_1} \bmod q. \end{aligned}$$

  Bob will reject the transmitted message from the intruder because the message cannot pass verification in the Equation (2).

- **Key-compromise Impersonation:** If the participating entity Alice's long-term private key had been stolen by Eve, Eve could impersonate Alice to cheat Bob. Without knowing the private key of Bob, Eve couldn't impersonate Bob to trick Alice.

- **Key Control:** Neither participating entities can control the session key alone. In this protocol, Alice and Bob selected random number k_{a1}, k_{a2}, k_{b1}, and k_{b2} as short-term secret key to form common session keys further. If the adversary Eve altered the exchanged message, Alice and Bob can hardly compute the same session keys.

6 Conclusions

Recently, security has become a critical research issue in wireless sensor networks, mobile Ad-hoc networks, smart cards, personal digital assistants, and cell phones. Owing to they are severely constrained in the resources e.g. processor, battery, and memory, key management will be the most challenging security issues to meet energy-efficient security. Key agreement protocol is an essential tool for communications security. In other words, each communication entity computes the common secret key utilize the information contributed by all the entities involved. ECC has attracted attention because of a smaller key size, reducing storage, low on CPU consumption, and transmission requirements. Especially, ECC is a popular method to apply key generation for resource-constrained devices. In this paper, we adopt the elliptic curve discrete logarithm problem (ECDLP) to establish n^2 common session keys between two communication parties in one session. This protocol is very efficient to produce session keys and suitable for resource-constrained key agreement. The proposed protocol is also secure to against the known-key attack, replay attack, forgery attack, key-compromise impersonation, and key control.

References

1. IEEE 2000, "IEEE Standard 1363-2000: Standard specifications for public key cryptography," *IEEE*, 2000.

2. I. F. Blake, G. Seroussi, and N. -P. Smart, "Advances Elliptic Curves in Cryptography," *Cambridge University Press*, 2005.

3. M. Brown, D. Hankerson, J. Lopez, and A. Menezes, "Software Implementation of the NIST Elliptic Curves Over Prime Fields," *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, pp. 250–265, 2001.

4. Ting-Yi Chang, Min-Shiang Hwang, and Wei-Pang Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, pp. 217–226, 2011.

5. Ting Yi Chang and Min-Shiang Hwang, "User-anonymous and short-term Conference Key Distribution System via link-layer routing in mobile communications," *International Journal of Mobile Communications*, vol. 9, no. 2, pp. 144–158, 2011.

6. Ting Yi Chang, Min-Shiang Hwang, and Wei-Pang Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, no. 1, pp. 217–226, 2011.

7. Ting-Yi Chang, Wei-Pang Yang, and Min-Shiang Hwang, "Simple authenticated key agreement and protected password change protocol," *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

8. Kou-Min Cheng, Ting-Yi Chang, and Jung-Wen Lo, "Cryptanalysis of security enhancement for a modified authenticated key agreement protocol," *International Journal of Network Security*, vol. 11, no. 1, pp. 55–57, 2010.

9. Certicom, "Standards for Efficient Cryptography," *SEC 2: Recommended Elliptic Curve Domain Parameters,*, Version 1.0, 2000.

10. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

11. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, pp. 469–472, Nov. 1985.

12. Lein Harn and Hung-Yu Lin, "An authenticated key agreement protocol without using one-way functions," in *Proceedings of the 8th National Conference on Information Security*, pp. 155–160, Kaohsiung, Taiwan, May 1998.

13. Lein Harn and Hung-Yu Lin, "Authenticated key agreement without using one-way hash functions," *Electronics Letters*, vol. 37, no. 10, pp. 629–630, 2001.

14. Min-Shiang Hwang, Chih-Wei Lin, and Cheng-Chi Lee, "Improved yen-joye's authenticated multiple-key agreement protocol," *Electronics Letters*, vol. 38, no. 23, pp. 1429–1431, 2002.

15. Min-Shiang Hwang, Song-Kong Chong, and Hsia-Hung Ou, "On the security of an enhanced UMTS authentication and key agreement protocol," *European Transactions on Telecommunications*, vol. 22, no. 3, pp. 99–112, 2011.

16. IEEE Standard Specifications for Public-Key Cryptography, "1363-2000," *IEEE Computer Society*, Jan. 2000, http://grouper.ieee.org/groups/1363.

17. Wen-Shenq Juang and Jing-Lin Wu, "Efficient user authentication and key agreement with user privacy protection," *International Journal of Network Security*, vol. 7, no. 1, pp. 120–129, 2008.

18. PAN Jin, LIU Xiaoqiong, XIE Minghui, and LIU Qiong, "Certificateless-based two-party authenticated key agreement protocols in a multiple PKG environment," *International Conference on Computer Science and Network Technology*, vol. 4, pp. 2364–2367, 2011.

19. M. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

20. K. Kumar, J. Nafeesa Begum, Dr. V. Sumathy, "A novel approach towards cost effective region-based group key agreement protocol for secure group communication," *International Journal of Computer Science and Information Security*, vol. 8, no. 2, pp. 65–74, 2010.

21. Cheng-Chi Lee, Min-Shiang Hwang, and Li-Hua Li, "A new key authentication scheme based on discrete logarithms," *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343–349, 2003.

22. Jie Liu and Jianhua Li, "A better improvement on the integrated diffie-hellman-dsa key agreement protocol," *International Journal of Network Security*, vol. 11, no. 2, pp. 114–117, 2010.

23. Jung-Wen Lo, Ji-Zhe Lee, Min-Shiang Hwang, and Yen-Ping Chu, "An advanced password authenticated key exchange protocol for imbalanced wireless networks," *Journal of Internet Technology*, vol. 11, no. 7, pp. 997–1004, 2010.

24. Jung-Wen Lo, Min-Shiang Hwang, and Chia-Hsin Liu, " An efficient key assignment scheme for access control in a large leaf class hierarchy," *Information Sciences*, vol. 181, no. 4, pp. 917–925, 2011.

25. Jung-Wen Lo, Shu-Chen Lin, and Min-Shiang Hwang, "A parallel password-authenticated key exchange protocol for wireless environments," *Information Technology and Control*, vol. 39, no. 2, pp. 146–151, 2010.

26. A. J. Menezes, M. Qu, and S. A. Vanstone, "Some key agreement protocols providing implicit authentication," in *Proceedings of 2nd Workshop Selected Areas in Cryptography*, pp. 22–32, May 1995.

27. Alfred J. Menezes, *Elliptic Curve Public Key Cryptosystem.* Kluwer Academic Publishing, 1993.

28. Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan, and Niraj K. Jha, "Characteristics of Cryptographic Algorithms and Security Protocols," *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 128–143, 2006.

29. K. Nyberg and R. A. Rueppel, "Weakness in some recent key agreement protocol," *IEE Electronics Letters*, vol. 30, no. 1, pp. 26–27, 1994.

30. Digital Signature Standard, "National Institute of Standards and Technology," *FIPS*, pub186-2, Jan. 2000.

31. Hsia-Hung Ou, Min-Shiang Hwang, and Jinn-ke Jan, "A cocktail protocol with the Authentication and Key Agreement on the UMTS," *Journal of Systems and Software*, vol. 83, no. 2, pp. 316–325, 2010.

32. Hsia-Hung Ou, Iuon-Chang Lin, Min-Shiang Hwang, and Jinn-ke Jan, "TK-AKA: using temporary key on Authentication and Key Agreement protocol on UMTS," *International Journal of Network Management*, vol. 19, no. 4, pp. 291–303, 2009.

33. Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan and Niraj K. Jha, "A study of cryptographic algorithms and security protocols," *IEEE Transaxtions on Mobile Computing*, vol. 5, pp. 128–148, 2006.

34. Chinniah Porkodi, Ramalingam Arumuganathan, and Krishnasamy Vidya, "Multi-authority Electronic Voting Scheme Based on Elliptic Curves," *International Jouranl of Network Security*, vol. 12, no. 2, pp. 84–91, 2011.

35. Marimuthu Rajaram and Thilagavathy Dorairaj Suresh, "An interval-based contributory key agreement," *International Journal of Network Security*, vol. 13, no. 2, pp. 92–97, 2011.

36. Y. M. Tseng, "Robust generalized MQV key agreement protocol without using one-way hash function," *Computer Standards & Interfaces*, vol. 24, no. 3, pp. 241–246, 2002.

37. Youliang Tian, Changgen Peng, and Jianfeng Ma, "Publicly Verifiable secret Secret Sharing Schemes Using Bilinear Piarings," *International Jouranl of Network Security*, vol. 14, no. 3, pp. 142–148, 2012.

38. Tzong-Sun Wu, Wei-Hua He, and Chien-Lung Hsu, "Security of authenticated multiple-key," *Electronics Letters*, vol. 35, no. 5, pp. 391–392, 1999.

39. Y. Wang, B. Ramamurthy and X. K. Zou, “The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks,” *IEEE International Conference on Communication*, vol. 5, pp. 2243–2248, 2006.

40. Chou-Chen Yang, Ting-Yi Chang, and Min-Shiang Hwang, “Cryptanalysis of simple authenticated key agreement protocols,” *IEICE Transactions on Foundations*, vol. E87-A, no. 8, pp. 2174–2176, 2004.

41. Sung-Ming Yen and M. Joye, “Improved authenticated multiple-key agreement protocol,” *Electronics Letters*, vol. 34, no. 18, pp. 1738–1739, 1998.

42. Mingwu Zhang, Bo Yang, Yusheng Zhong, Pengcheng Li, and Tsuyoshii Takagi, “Cryptanalysis and Fixed of Short Signature Scheme Without Random Oracle from Bilinear Parings,” *International Jouranl of Network Security*, vol. 12, no. 3, pp. 130–136, 2011.