A Secure and Efficient Three-Pass Authenticated Key Agreement Protocol Based on Elliptic Curves*

Meng-Hui Lim[1], Chee-Min Yeoh[1], Sanggon Lee[2], Hyotaek Lim[2], and Hoonjae Lee[2]

[1] Department of Ubiquitous IT, Graduate School of Design & IT, Dongseo University, Busan 617-716, Korea
{menghui.lim,yeohcm}@gmail.com
[2] Division of Computer and Information Engineering, Dongseo University, Busan 617-716, Korea
{nok60,htlim,hjlee}@dongseo.ac.kr

Abstract. Key agreement protocol is of fundamental importance in providing data confidentiality and integrity between two or more parties over an insecure network. In 2004, Popescu [14] proposed an authenticated key agreement protocol in which its security is claimed. However, Yoon and Yoo [19] discovered its vulnerabilities two years later and proposed an improved variant of it. In this paper, we highlight the vulnerability of this improved variant under the LaMacchia et al.'s extended Canetti-Krawczyk security model [12]. With this, we propose another enhanced version of Popescu's protocol which offers stronger security features and appears to be significantly more efficient than Yoon-Yoo's scheme. In order to justify our claims, we present a thorough heuristic security analysis on our scheme and compare the computational cost and security attributes with the surveyed schemes.

1 Introduction

Authenticated key agreement protocol is essential for two or more specific entities to communicate securely over an open network (e.g. the Internet), in which the communication can be fully controlled by adversaries. The protocol must be properly designed in such a way that at the end of the protocol execution, both communicating parties can be assured of each other's identity and agreed on an unique shared secret key (also known as *session key*) based on the information contributed equally by each party. Once the key agreement is successfully carried out, the derived session key can be used subsequently to create a confidential channel particularly for preserving data confidentiality and integrity. Therefore, an authenticated key agreement protocol should strictly prohibit other unauthenticated parties aside from the intended protocol participants from gaining access to the shared secret under any circumstances.

* This work was supported by MIC of Korea, under ITRC support program supervised by IITA (IITA-2007-C1090- 0701-0026).

A. Das et al. (Eds.): NETWORKING 2008, LNCS 4982, pp. 170–182, 2008.

Key agreement protocols can employ symmetric or/and asymmetric cryptography in its construction. Generally, symmetric protocols require a common secret key to be shared in prior between the communicating parties whereas asymmetric or public key-based protocols require the communicating entities to learn only specific authenticated public information of their peer (such as identity, public key, etc.), which is usually exchanged via certificates. This paper is concerned with two-party authenticated key agreement protocols in the asymmetric settings.

Over the years, there has been overwhelming interest in the use of public key-based cryptography, mainly due to the employment of elliptic curve operations in constructing cryptographic key agreement protocols. However, as designing a secure and efficient protocol is far from being a simple task, many protocols have been proven to be fallible. Strangio [16], in 2005, has proposed an efficient two-party elliptic curve based key agreement protocol, namely ECKE1, which is believed to be secure at the first glance since the security attributes are well discussed in his work. Unfortunately, Wang et al. [18] have revealed a weakness in Strangio's protocol recently. Specifically, they have presented a valid key compromise impersonation attack on it. Besides, Duraisamy et al. [8] have also pointed out a similar attack independently. To defeat the vulnerability of ECKE1, Wang et al. [18] and Strangio [17] have proposed their revised version of ECKE1, namely ECKE1N and ECKE1R respectively at the expense of a higher computational workload. In 2004, Popescu [14] has proposed another notable key agreement scheme, in which the author claimed such protocol to be secure and more efficient than the exisitng ones due to its low computational cost required at both communicating entities. However, in 2006, Yoon-Yoo [19] have subverted Popescu's claim by exploring a series of cryptographic attacks which encompass a key-compromise impersonation attack, a reflection attack and a replay attack. Subsequently, Yoon-Yoo have proposed another improved scheme [19] in order to defeat these vulnerabilities. To justify their claims on the security of their protocol, they have demonstrated a detailed heuristic security analysis on it.

In this paper, we point out that Yoon-Yoo's improved scheme [19] does not seem secure when both ephemeral keys of a specific completed session are exposed to a passive adversary who only eavesdropped on the session. This compromise of session-specific secret information is regarded as one of the essential attacks captured by the extended Canetti-Krawczyk security model [12]. Based on the shortcoming, we propose another improved variant of Popescu's protocol [14] and we claim that our protocol is able to achieve a stronger and higher sense of security and efficiency as compared to Yoon-Yoo's scheme. The structure of this paper is organized as follows. In Section 2, we illustrate the basic concepts needed to evaluate the security properties of a key agreement protocol and define the elliptic curve as well as several hard cryptographic problems. In section 3, we revisit Yoon-Yoo's improved scheme and illustrate its insecurity. We propose an efficient three-pass authenticated key agreement protocol and evaluate its security in sections 4 and 5 successively. In Section 6, we compare our scheme with

the other existing key agreement schemes in terms of computational efficiency and security. Lastly, we conclude this paper in Section 7.

2 Preliminaries

2.1 Security Properties of Key Agreement Protocol

There are numerous security attributes that have been defined and discussed, specifically for the analysis of the key agreement protocols. The most fundamental property of a key agreement protocol is the passive attack resilience, that is, the protocol should be able to prevent the adversary from obtaining the session key or any other useful information (such as ephemeral keys) by merely eavesdropping on the protocol. On top of that, we also assume that the adversary has full control of the communications on the open network, such that the adversary is capable of intercepting, deleting, injecting, altering and replaying messages in any online instance of the key agreement protocol. These potential attacks has in fact led to the additional security properties: known session key security, perfect forward secrecy, key compromise impersonation resilience, unknown key-share resilience and key control resilience. Please refer to Blake-Wilson and Menezes's definition of these security attributes [4,3] for an excellent overview.

2.2 Elliptic Curve Cryptography

The notion of elliptic curve cryptography was first introduced independently by Miller [13] and Koblitz [10]. Since then, numerous elliptic curve cryptosystems have been proposed and employed. The main attraction of it is that it allows much smaller parameters (e.g. key size) to be employed in order to achieve an equivalent level of security as compared to the traditional public-key cryptosystems such as RSA and DSA. Since an elliptic curve cryptosystem, that is mainly based on the intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP), takes full exponential time, its resistance against the sub-exponential attack offers potential reductions in processing power and memory size which is essential in applications on constrained devices [15].

Let $E(\mathbb{F}_q)$ be an elliptic curve of defined over a finite field $\mathbb{F}_q$ of characteristic p. The public elliptic curve domain parameters over $\mathbb{F}_q$ is defined as a 8-tuple $(q, FR, S, a, b, P, n, h)$, where q is the prime order of the field, FR (*field representation*) indicates the representation used for the elements of $\mathbb{F}_q$, S is the random seed for elliptic curve generation, the coefficients $a, b \in_R \mathbb{F}_q$ define the equation of elliptic curve E over $\mathbb{F}_q$ ($y^2 = x^3 + ax + b$ for $p = q > 3$, where $4a^3 + 27b^2 \neq 0$), the base point $P = (x_p, y_p)$ in $E(\mathbb{F}_Q)$, the prime n is the order of P ($n > 2^{160}$) and the cofactor $h = \sharp E(\mathbb{F}_Q)/n$, where $\sharp E(\mathbb{F}_Q)$ denotes the number of $\mathbb{F}_Q$-rational points on E. These parameters should be chosen appropriately to prevent the employment of any efficient algorithm from solving the Discrete Logarithm Problem (DLP) or the computational Diffie-Hellman Problem (CDHP) in the cyclic subgroup $\langle P \rangle$ [16].

Since many cryptographic primitives base their security on the underlying assumptions in which the DLP and CDHP on some cyclic groups are intractable, our proposed key agreement protocol would not be exceptional. Our protocol also rests upon a few related conjectures before its security can be claimed. Now, we define several cryptographic problems which we will assume their hardness throughout this paper.

Conjecture 1 (ECDLP). *Let $E(\mathbb{F}_q)$ and P be defined as above. The Elliptic Curve Discrete Logarithm Problem is said to be intractable if for any probablilistic polynomial time Turing Machine $\mathcal{A}$ with the knowledge of $Y = xP$, where $Y \in \langle P \rangle$, the probability of success in computing $\log_P Y = x \in_R [1, n-1]$, denoted as $\mathtt{Succ}^{\mathtt{ecdlp}}_{P,E(\mathbb{F}_q)}(\mathcal{A})$ is negligible:*

$$\mathtt{Succ}^{\mathtt{ecdlp}}_{P,E(\mathbb{F}_q)}(\mathcal{A}) = Pr\left[\begin{array}{c} x \in_R [1, n-1]; \\ Y = xP; \\ \mathcal{A}(n, P, Y) = \log_P Y \end{array}\right] \leqslant \varepsilon,$$

where the probability is taken over the coin tosses of $\mathcal{A}$ for any random choice of x.

Conjecture 2. (ECCDHP). *Let $E(\mathbb{F}_q)$ and P be defined as above. The Elliptic Curve Computational Diffie-Hellman Problem is said to be intractable if for any probablilistic polynomial time Turing Machine $\mathcal{A}$ with the knowledge of $R = rP$ and $S = sP$, where $R, S \in \langle P \rangle$, the probability of success in computing rsP, denoted as $\mathtt{Succ}^{\mathtt{eccdhp}}_{P,E(\mathbb{F}_q)}(\mathcal{A})$ is negligible:*

$$\mathtt{Succ}^{\mathtt{eccdhp}}_{P,E(\mathbb{F}_q)}(\mathcal{A}) = Pr\left[\begin{array}{c} r, s \in_R [1, n-1]; \\ R = rP; S = sP; \\ \mathcal{A}(n, P, R, S) = rsP \end{array}\right] \leqslant \varepsilon,$$

where the probability is taken over the coin tosses of $\mathcal{A}$ for any random choices of r and s.

Conjecture 3 (ECDDHP). *Let $E(\mathbb{F}_q)$ and P be defined as above. The Elliptic Curve Decisional Diffie-Hellman Problem is said to be intractable if for any probablilistic polynomial time Turing Machine $\mathcal{A}$ with the knowledge of $R = rP$ and $S = sP$, where $R, S \in \langle P \rangle$, the probability of success in distinguishing the two probability distributions (P, R, S, Q) and (P, R, S, T), denoted as $\mathtt{Succ}^{\mathtt{ecddhp}}_{P,E(\mathbb{F}_q)}(\mathcal{A})$ is negligible:*

$$\mathtt{Succ}^{\mathtt{ecddhp}}_{P,E(\mathbb{F}_q)}(\mathcal{A}) = Pr\left[\begin{array}{c} r, s, t \in_R [1, n-1]; \\ R = rP; S = sP; Q = rsP; T = tP; \\ b \in_R [0, 1]; \\ \text{if}(b = 0), U = T, \text{else } U = Q; \\ \mathcal{A}(n, P, R, S, U) = b \end{array}\right] \leqslant \frac{1}{2} + \varepsilon,$$

where the probability is taken over the coin tosses of $\mathcal{A}$ for any random choices of r, s and t.

3 Revisiting Yoon-Yoo's Key Agreement Protocol

This section reviews an improved variant of Popescu's protocol due to Yoon and Yoo [19]. Figure 1 gives a clear description of this three-pass protocol. The two communication parties are specified to be A (initiator) and B (responder), with their long term public/private key pairs (Y_A/s_A) and (Y_B/s_B) (where $Y_A = s_A P$ and $Y_B = s_B P$), and their chosen ephemeral secret key r_A and r_B respectively. Consider that the static public keys are exchanged via certificates, which are generally issued by a Certification Authority (CA) binding each entity's identity (ID_A/ID_B) to his/her long term public key (Y_A/Y_B).

Once the message exchange is completed, the session key, K_{AB} is computed as $H(ID_A, ID_B, x_{V_A}, x_{V_B}, x_{K_2}, x_{K_1})$, where $K_1 = r_A r_B s_A P$ and $K_2 = r_A r_B s_B P$ are the shared secrets. The conjectured security attributes of this protocol are known key security, perfect forward secrecy, key compromise impersonation resilience, session key security, reflection resilience and replay resilience.

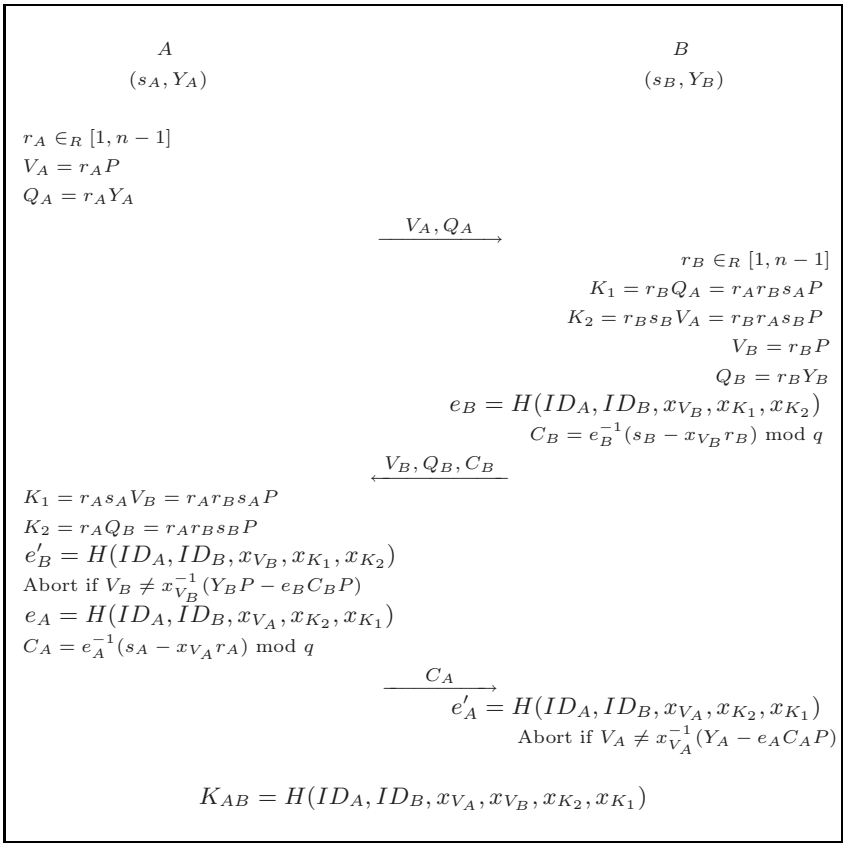

Fig. 1. Yoon-Yoo's Improved Variant of Popescu's Key Agreement Protocol

3.1 The Insecurity of Yoon-Yoo's Scheme in the Extended Canetti-Krawczyk Security Model

Recently, LaMacchia et al. [12] have extended the Canetti-Krawczyk security model [6] for authenticated key exchange to achieve a stronger sense of security by capturing all possible attacks resulting from ephemeral and long-term data compromise. Specifically, they view the message exchange and the computation of a common session key as a function of at least four pieces of secret information, which encompass their own long-term and ephemeral secret keys and the other party's static and ephemeral secret keys. Of these four pieces of information, the adversary in their model is allowed to reveal any subset of the four which does not contain both the long-term and ephemeral secrets of one of the parties that would trivially break the scheme.

For passive sessions where the adversary has only the eavesdropping ability (without active intervention in the session establishment), the adversary may reveal both ephemeral keys, both long-term secret keys, or one of each from the two different parties once the session is established. For example, as defined by Krawczyk [11], the weak Perfect Forward Secrecy (wPFS) is concerned with the security against revelation of both the long term keys after the passive session is completed. For active sessions where the adversary may forge the communication of one of the parties during protocol execution, the adversary is allowed to reveal a long-term secret key or ephemeral secret key of the other party. This is because if the adversary can also reveal the long term key of the same party, then the adversary can trivially compute the session key. As pointed out by Krawczyk [11], this is the unattainable full Perfect Forward Secrecy property for two-pass AKE protocol.

By considering only passive sessions in this part of our analysis, we now prove that Yoon-Yoo's scheme is insecure under the extended Canetti-Krawczyk security model. As claimed by Yoon and Yoo, their improved protocol indeed provides weak perfect forward secrecy as the compromise of both long term keys of A and B (s_A, s_B) would not expose any of the established session keys to the adversary. Also, it is easy to see that the security of the session key remains valid even if we consider the exposure of the static key and ephemeral key, one of each from the two different parties, (s_A, r_B) or (s_B, r_A) after the session completes. The revelations in these two cases truly prevent the adversary from reconstructing both the shared secrets K_1 and K_2 of their protocol. However, the analysis does not end here. Notice that if both the ephemeral secrets r_A and r_B are revealed to the passive adversary, she would then be able to recompute the shared secrets

$$K_1 = r_B Q_A = r_A r_B s_A P,$$

$$K_2 = r_A Q_B = r_A r_B s_B P$$

and reconstruct the session key without needing to learn any of the static private keys s_A or s_B. In fact, this leakage can possibly happen in practical scenario as the ephemeral secret information may be stored in insecure memory or the random-number generator of a party may be corrupted. As a result, the adversary is able to gain access to the ephemeral data and recompute the session key, which renders the specific session insecure.

4 A Secure and Efficient Authenticated Key Agreement Protocol

Now we propose a secure and efficient authenticated key agreement protocol. As usual, the two communicating parties are A and B with their long term public/private key pairs (Y_A/s_A) and (Y_B/s_B) respectively. In this protocol, we employ two independent one-way collision-free hash functions, namely $H_1 : \{0,1\}^* \rightarrow \mathbb{F}_q$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$, where k is the security parameter. Assume that the public keys are exchanged via certificates. Our key agreement protocol can be performed as follows:

1. Initially A selects a random $r_A \in_R [1, n-1]$ and computes

$$T_A = r_A Y_A. \tag{1}$$

   In the similar manner, B selects a random $r_B \in_R [1, n-1]$ and computes

$$T_B = r_B Y_B. \tag{2}$$

   Then, A, in the role of the initiator, initiates the key agreement process by sending T_A to B.
2. Upon receiving A's message, B computes accordingly

$$K = hr_B s_B T_A = hr_A r_B s_A s_B P, \tag{3}$$

$$e_B = H_1(ID_A, ID_B, x_{T_A}, x_{T_B}, x_K), \tag{4}$$

$$z_B = r_B + e_B s_B \bmod q, \tag{5}$$

   where x_{T_A} denotes the x-coordinate of T_A, x_{T_B} denotes the x-coordinate of T_B, x_K denotes the x-coordinate of K and h denotes the cofactor. Finally, B, in the role of the responder, sends the triple (T_B, z_B) to A.
3. On reception of B's message, A calculates

$$K = hr_A s_A T_B = hr_A r_B s_A s_B P, \tag{6}$$

$$e'_B = H_1(ID_A, ID_B, x_{T_A}, x_{T_B}, x_K), \tag{7}$$

   and verifies whether

$$T_B \stackrel{?}{=} z_B P - e'_B Y_B \tag{8}$$

   holds. If the verification fails, A terminates the execution. Otherwise, A proceeds to compute

$$e_A = H_1(ID_A, ID_B, x_{T_B}, x_{T_A}, x_K), \tag{9}$$

$$z_A = r_A + e_A s_A \bmod q \tag{10}$$

   accordingly and then sends z_A to B.

4. B computes
$$e'_A = H_1(ID_A, ID_B, x_{T_B}, x_{T_A}, x_K), \tag{11}$$
and checks whether
$$T_A \stackrel{?}{=} z_A P - e'_A Y_A \tag{12}$$
holds. If the verification fails, A terminates the execution. Otherwise, B can be assured of A's identity.
5. At last, both A and B terminate and agree on the common session key which is computed as
$$K_{AB} = H_2(ID_A, ID_B, x_{T_A}, x_{T_B}, z_A, z_B, x_K). \tag{13}$$

5 Security Analysis

In this section, we scrutinize our proposed key agreement protocol in detail so as to ensure that our protocol is able to achieve the desired security attributes of a key agreement protocol and also able to resist against the known cryptographic attacks.

Known session key security (KSK-S). As shown in our protocol description, the session key is derived from the ephemeral keys (r_A, r_B) of the specific session and the long term keys (s_A, s_B) of the protocol entities. This would result in distinct independent session key in each protocol execution. On top of that, a one-way collision-resistant cryptographic hash function is used to derive the session key. Thus, obtaining any other session keys would not benefit the adversary in mounting a successful attack against a protocol run without the information set (r_A, s_A) or (r_B, s_B) which is required in the computation of the shared secret K. Therefore, based on the conjectures that we have defined in Section 2, we claim that the knowledge of some previous session keys would not allow the adversary to gain any advantage in deriving any future and other previous session keys.

Weak Perfect Forward secrecy (wPFS). Suppose that both A and B's long term private key s_A and s_B have been exposed. However, the adversary, with the eavesdropped information (T_A, T_B, z_A, z_B) of any particular session, would not be able to recover the respective established session key since the adversary does not know the involved ephemeral private key r_A or r_B which are needed in the computation of the shared secret K. And also, the intractability of ECCDHP has significantly thwarted the adversary's attempt in computing K by using merely T_A and T_B. Hence, we claim that our enhanced protocol enjoys weak perfect forward secrecy.

Key-Compromise Impersonation Resilience (KCI-R). Suppose that A's long term private key s_A has been compromised and instead of directly impersonating A, the adversary now wishes to impersonate B in order to establish a session with A. However, the adversary is unable to compute the shared secret K with the available information (s_A, r_B, T_A) since the required information set is (r_A, s_A, T_B) or (r_B, s_B, T_A). Even if the adversary

is able to compute K by some means and subsequently e_B, she would again face difficulty in computing z_B in which it is impossible without the knowledge of s_B. Hence, the adversary is significantly prevented from launching a successful KCI attack against our protocol. Generally, the same situation will result when the long term key s_B is compromised (the adversary would impersonate A in this case and her effort will be foiled in computing K and z_A) as our key agreement protocol is symmetric. As a result, we claim that this protocol is able to withstand the KCI attack under all circumstances.

Key Replicating Resilience (KR-R). The key replicating attack was first introduced by Krawczyk [11] where the illustration of it involves oracle queries described in Bellare and Rogaway's random oracle model [1,2]. This attack, if successfully carried out by the adversary, would force the establishment of a session, $\mathcal{S}$ (other than the Test session or its matching session) to agree on a same session key as the Test session, by means of intercepting and altering the message from both communicating parties during transmission. Since the Test session and $\mathcal{S}$ are non-matching, the adversary may issue a Reveal query to the oracle associated with $\mathcal{S}$ and she can then distinguish whether the Test session key is real or random. (See [5,11] for more details on this attack.) Notice that the message integrity of T_A and T_B has been guaranteed by having each party to calculate the hash function e_A and e_B which will be bound to z_A and z_B respectively. Since the adversary has no idea in forging z_A or z_B along with T_A or T_B in such a way that the verification steps in Eqs. (8) or (12) would yield a positive result, she would not be able to force the establishment of non matching sessions to possess a common session key. As a result, if the adversary reveals A's session key, she would not be able to guess B's session key correctly with non-negligible probability and vice versa. On top of that, A and B also include the session identifiers (T_A, T_B, z_A, z_B) in the key derivation function in Eq. (13) so as to fully eliminate the possibility of this attack [7]. Therefore, we claim that our protocol is secure against the key replicating attack.

Unknown Key-Share Resilience (UKS-R). It is apparent that an adversary masquerading as another legal entity E cannot deceive A into believing that messages received from E are originated from B, even if the adversary could obtain E's certificate. Suppose that in a protocol run, the adversary intercepts the second flow of message transmission, erases B's certificate and attaches E's certificate to the message, and sends (T_B, z_B) along with E's certificate to A. After computing

$$e'_E = H_1(ID_A, ID_E, x_{T_A}, x_{T_B}, x_K) \tag{14}$$

A would eventually terminate the protocol execution as she verifies that

$$T_B \stackrel{?}{=} z_B P - e'_E Y_E \tag{15}$$

does not hold. As a result, the attack attempt fails. It is easy to see that this attack does not work on tricking B in the similar manner by attaching E's certificate to A's message. In a nutshell, we speculate that the inclusion

of A and B' identity (ID_A and ID_B) in the hash function e_A and e_B along with the employment of the sender's public key in the verification steps in Eqs. (8) and (12) would significantly prevent the adversary from launching the unknown key-share attack on our proposed protocol.

Replay Resilience (R-R). In any protocol run, an adversary may attempt to deceive a legitimate participant through retransmitting the eavesdropped information of the impersonated entity from a previous protocol execution. Although the adversary might be unable to compute the session key at the end of the protocol run, her attack is still considered successful if she manage to trick the protocol entity to complete a session with her, believing that the adversary is indeed the impersonated party. In this replay analysis, we reasonably assume that the prime order n of point P is arbitrarily large such that the probability of a protocol entity selecting the same ephemeral key (r_A , $r_B \in [1, n-1]$) in two different sessions is negligible. Consider a situation where the adversary (masquerading as A) replays A's first message ($T_{A(\text{old})}$) from a previous protocol run between A and B. After B has sent her a fresh (T'_B, z'_B) in the second message flow, the adversary would abort since she could not produce (by means of forging or replaying) z_A corresponding to T'_B. Notice that the same replay prevention works in the reverse situation where B's message is replayed. The adversary would fail eventually in generating z_B corresponding to the fresh T'_A. Hence, we claim that message replay in our protocol execution can always be detected by both A and B.

Key Control Resilience (KC-R). Obviously in our protocol, no single protocol participant could force the session key to a predetermined or predicted value since the session key K_{AB} is computed from the shared secret K which consists of both A and B's long term and ephemeral keys.

Besides all these security attributes, we also speculate that our protocol is able to overcome the security flaw of Yoon-Yoo's protocol and achieves a stronger sense of security as specified in LaMacchia's model. Note that in our protocol, the only way to compute the shared secret $K = r_A r_B s_A s_B P$ (for reconstructing the session key) is to employ both ephemeral key and secret key of the same protocol participant, along with the other participant's public information T_A or T_B. Thus, this fully prevents a passive adversary to recompute the established session key by revealing any other subset of the four secret keys (r_A, s_A, r_B, s_B).

6 Comparison of Elliptic Curve Based Key Agreement Schemes

The computational efficiency of elliptic curve based key agreement schemes mainly relies on the number of major operations (elliptic curve scalar point multiplication and point addition) that is required to be performed by each protocol entity in a protocol run. In other words, while preserving the desired security features, one may need to minimize the number of these operations in ensuring the performance of key agreement protocol.

Table 1. Comparison of Online Computational Efficiency

		Without precomputations				With precomputations			
Protocol	Pass	PSM	PA	FM	H	PSM	PA	FM	H
Popescu [14]	2	3	0	0	1	1	0	0	0
ECKE1 [16]	2	3	1	4	3	2	1	2	2
ECKE1N [18]	2	4	2	2	2	3	2	0	1
ECKE1R [17]	3	4	1	3	3	3	1	1	2
Yoon-Yoo [19]	3	6	1	5	3	4	1	2	3
Our Protocol	3	4	1	3	2	3	1	1	2

Table 1 summarizes the comparison of the online computational efficiency between the existing elliptic curve based key agreement schemes with our proposed scheme. Note that we do not consider pairing based schemes in our comparison work as the computation of a paring operation is equivalent to several times of an elliptic curve point scalar multiplication, which would turn out to be trivially inefficient. Moreover, we record the online computational operations such as point scalar multiplications (PSM), point addition/point subtraction (PA), field multiplication (FM) and hash function (H) with and without precomputations. For ease of comparison, we treat all the key derivation functions employed in the surveyed protocols as hash functions. And also, among the surveyed schemes, Popescu [14] and Yoon-Yoo [19] have ignored the cofactor in describing their protocol which is required to ensure that the point sent is a member of the prime order subgroup. To illustrate a fairer comparison, we assume that the shared secret(s) in their schemes are computed as a function of the cofactor h, in which we replace $K = r_A r_B P$ with $K = hr_A r_B P$ in Popescu's protocol and in Yoon-Yoo's protocol, we replace $K_1 = r_A r_B s_A P$ with $K_1 = hr_A r_B s_A P$, $K_2 = r_A r_B s_B P$ with $K_2 = hr_A r_B s_B P$ in this efficiency analysis. Table 2 illustrates the comparison of security properties offered by the surveyed schemes with our proposed scheme. EE basically refers to the secuity against the compromise of both parties' ephemeral secret for a passive completed session and ES refers to the similar security except that for the ephemeral and the static key, one of each from the two different parties is exposed.

From Table 1, it is apparent that the two-pass key agreement protocols require lesser computational requirements than the three-pass schemes do. Despite being more efficient, all of them except Popescu's scheme exchange unauthenticated messages, which means that private keys are not used in the message construction. This has led to session key establishment without authenticating the partner's identity in prior. And also, as time synchronization is not always feasible, two-pass protocols can hardly withstand the replay attack even though the transmitted message is signed with the long term private key (since the signed message can also be replayed), as illustrated in Table 2.

Basically, the inherent demerits of two-pass protocol can be successfully overcome by the three-pass schemes, where Yoon-Yoo's scheme [19] (except for the EE attribute), Strangio's revised scheme ECKE1R [17] and our scheme offer

Table 2. Comparison of Conjectured Security Attributes

Protocol	KSK-S	wPFS	KCI-R	KR-R	UKS-R	R-R	KC-R	EE	ES
Popescu [14]	√	√	X	√	√	X	√	X	X
ECKE1 [16]	√	√	X	√	√	X	√	√	√
ECKE1N [18]	√	√	√	√	√	X	√	√	√
ECKE1R [17]	√	√	√	√	√	√	√	√	√
Yoon-Yoo [19]	√	√	√	√	√	√	√	X	√
Our Protocol	√	√	√	√	√	√	√	√	√

more attractive security attributes than the two-pass protocols do. Comparing to the existing three-pass schemes without precomputations, our protocol appears to be the most efficient as our protocol only requires each participant to perform 4 point scalar multiplications, 1 point addition, 3 field multiplications and 2 hash functions online in order to establish a secure session key, while Yoon-Yoo's scheme requires 6 point scalar multiplications, 1 point addition, 5 field multiplications and 3 hash functions, and ECKE1R requires 4 point scalar multiplications, 1 point addition, 3 field multiplications and 3 hash functions to be computed by each party in a protocol execution. Even with precomputation where extra storage requirement is incurred, our protocol achieves the same online computational efficiency as ECKE1R as it requires only the minimum computational effort for a party to compute 3 point scalar multiplications, 1 point addition, 1 field multiplication and 2 hash functions in a protocol run.

7 Conclusion

In conclusion, we have revealed the weakness of Yoon-Yoo's protocol in providing security as the ephemeral secret of both communicating parties are disclosed. By aiming to eliminate this security flaw, we have proposed another improved variant of Popescu's authenticated key agreement protocol. Besides that, we have evaluated the heuristic security of the protocol thoroughly to ensure that the analyzed security features are all offered. Furthermore, we have also compared our proposed scheme with several existing elliptic curve public-key key agreement schemes in terms of computational efficiency and security attributes. As a result, our protocol turns out to be a more secure and efficient improved variant of Popescu's key agreement scheme as compared to Yoon-Yoo's protocol.

References

1. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 110–125. Springer, Heidelberg (1994)
2. Bellare, M., Rogaway, P.: Provably Secure Session Key Distribution: The Three Party Case. In: 27th ACM Symposium on the Theory of Computing - ACM STOC, pp. 57–66 (1995)

3. Blake-Wilson, S., Johnson, D., Menezes, A.: Key Agreement Protocols and their Security Analysis. In: Darnell, M.J. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 30–45. Springer, Heidelberg (1997)
4. Blake-Wilson, S., Menezes, A.: Authenticated Diffie-Hellman key Agreement Protocols. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 339–361. Springer, Heidelberg (1999)
5. Boyd, C., Choo, K.-K.R.: Security of Two-Party Identity-Based Key Agreement. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 229–243. Springer, Heidelberg (2005)
6. Canetti, R., Krawczyk, H.: Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer, Heidelberg (2001)
7. Choo, K.-K.R., Boyd, C., Hitchcock, Y.: On Session Key Construction in Provably-Secure Key Establishment Protocols. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 116–131. Springer, Heidelberg (2005)
8. Duraisamy, R., Salcic, Z., Strangio, M.A., Morales-Sandoval, M.: Supporting Symmetric 128-bit AES in Networked Embedded Systems: An Elliptic Curve Key Establishment Protocol-on-Chip. EURASIP Journal of Embedded Systems 2007(9) (2007)
9. Diffie, W., Hellman, M.: New Directions in Cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
10. Koblitz, N.: Elliptic Curve Cryptosystems. Mathematics of Computation 48, 203–209 (1987)
11. Krawczyk, H.: HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 546–566. Springer, Heidelberg (2005)
12. LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger Security of Authenticated key Exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 1–16. Springer, Heidelberg (2007)
13. Miller, V.S.: Use of Elliptic Curves in Cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)
14. Popescu, C.: A Secure Authenticated Key Agreement Protocol. In: Proceedings of the 12th IEEE Mediterranean (MELECON 2004), vol. 2, pp. 783–786 (2004)
15. Raju, G.V.S., Akbani, R.: Elliptic Curve Cryptosystem and its Applications. In: Proceedings of the 2003 IEEE International Conference on Systems. Man and Cybernetics (IEEE-SMC), vol. 2, pp. 1540–1543 (2003)
16. Strangio, M.A.: Efficient Diffie-Hellmann Two-Party Key Agreement Protocols based on Elliptic Curves. In: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 324–331 (2005)
17. Strangio, M.A.: Revisiting an Efficient Elliptic Curve Key Agreement Protocol. Cryptology ePrint Archive: Report 081 (2007)
18. Wang, S., Cao, Z., Strangio, M.A., Wang, L.: Cryptanalysis of an Effcient Diffe-Hellman Key Agreement Protocol based on Elliptic Curves. Cryptology ePrint Archive: Report 026 (2006)
19. Yoon, E.-J., Yoo, K.-Y.: An Improved Popescu's Authenticated Key Agreement Protocol. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 276–283. Springer, Heidelberg (2006)