Avoiding Key Off-Set attack in a Pairing-Free Certificateless Key Agreement Protocol based on ECC

Hassan M. Elkamchouchi
Elec. Eng. Dept, Fac. of Eng,
Alexandria Univ.
Alexandria
Egypt

Eman F. Abou Elkheir
Elec. Eng. Dept, Fac. of Eng,
Kafr Elsheikh Univ.
Kafr Elsheikh
Egypt

Yasmine Abouelseoud
Eng. Math. Dept, Fac. of Eng,
Alexandria Univ.
Alexandria
Egypt

ABSTRACT
A key establishment protocol allows entities to establish a common secret key to ensure secure communications over an insecure public network. This paper proposes two new two-party key agreement protocols. Both protocols do not involve bilinear pairings. The first protocol is a certificate-based key agreement protocol that is more efficient than [1] due to its dependence on the elliptic curve discrete logarithm problem and the second is an extension to a certificateless key agreement protocol. Both protocols depend on the use of an authentication message to check that the shared session key is equal for both entities before using it. This authentication message prevents the key off-set attack that is valid for the Haiyan-Sun protocol [2]. The security analysis of the second protocol is discussed. The proposed certificateless key agreement protocol is compared with other protocols in literature [2,3,4] and it requires minimal computational cost. Moreover, this protocol is implemented using the Mathematica (7) program.

General Terms
Security, Cryptography

Keywords
Key Agreement (KA) protocol, Elliptic Curve Discrete Logarithm Problem, ID-Based cryptography, Key Off-Set Attack, Certificateless cryptosystems.

1. INTRODUCTION
Public key cryptography has become the traditional way to realize network and information security. The problem of certificate management in a traditional public key infrastructure arises from needing a trusted certification authority to issue a certificate binding the identity and the public key of an entity. Shamir proposed a new public key paradigm called identity-based public key cryptography [5] to overcome this problem. However, identity-based public key cryptography requires a trusted key generation center (KGC) to generate a private key for each entity. So, we are confronted with the key escrow problem. Fortunately, the two problems in traditional public key infrastructure and identity-based public key cryptography can be prohibited by introducing certificateless public key cryptography [6], which can be conceived as an intermediate structure between traditional public key infrastructure and identity-based cryptography. The first certificateless two-party authenticated key agreement protocol appears in the seminal paper by Al-Riyami and Paterson (2003) [6].

Some early certificateless key exchange protocols (2005 till 2010) [7-13] have been proposed with heuristic security analysis. These protocols involve bilinear pairings and the pairing is regarded as a computationally expensive operation. The relative computation cost of a pairing is approximately twenty times higher than that of the scalar multiplication over elliptic curve group [14]. Therefore, the certificateless key agreement protocols without bilinear pairings would be more appealing in terms of efficiency. Recently, several certificateless key exchange protocols without pairing have been proposed in (2009 till 2011) [15-18]. However, Yang et al. [17] (2011) pointed out that both of Geng et al.'s protocol [15] (2009) and Hou et al.'s protocol [16] (2009) are not secure. They proposed an improved certificateless key agreement protocol. He et al. [18] (2011) also proposed a certificateless key agreement protocol without pairing. In (2011 and 2012), Debiao [4,3] proposed two certificateless key agreement protocols without pairings. In 2013, Haiyan-Sun [2] proposed another two-party key agreement protocol without pairings.

In this paper, the weakness of Haiyan-Sun protocol, which is its susceptibility to the key off-set attack is demonstrate. Also, two new key agreement protocols are proposed; the first is a two-party certificate-based key agreement protocol and the second is a certificateless two-party key agreement protocol. Both protocols are based on elliptic curve cryptography.

The rest of this paper is organized as follows. Section two presents the preliminaries. In section three, the security properties of a two-party key agreement protocol are provided. Section four introduces an overview of Haiyan-Sun protocol. Section five presents the first of the proposed protocols followed by the proposed certificateless key agreement protocol in section six. Section seven covers the security analysis and a comparative study of the proposed certificateless protocol followed by the implementation in section eight. Finally, section nine concludes the paper.

2. PRELIMINARIES

2.1 Notations
In this subsection, the notations used in this paper are introduced.

p,n	:two large prime numbers
F_p	:a finite field
E/F_p	: an elliptic curve defined on F_p
G	: the cyclic additive group composed of the points on E/F_p
P	: a generator of G
$H_1(.)$	: a secure one-way hash function, where $H_1:\{0,1\}^* \times G \rightarrow Z_n^*$
$H_2(.)$	: a secure one-way hash function, where $H_2:\{0,1\}^* \times G \rightarrow Z_n^*$
ID_i	: the identity of user i

(s, P_{pub}) : the KGC's private/public key pair, where $P_{pub} = s.P$

(x_i, P_i) : the user i 's secret value/public key pair, where $P_i = x_i.P$

(r_i, R_i) : a random point generated by KGC, where $R_i = r_i.P$

(s_i, R_i) : the user i 's partial private key, where $s_i = r_i + h_i.s \bmod n$, $h_i = H_1(ID_i, P_i, R_i)$

(R_i, P_i) : the user i 's public key pair

(a, T_A) : the user A's ephemeral private/public key pair, where $T_A = a.P$

(b, T_B) : the user B's ephemeral private/public key pair, where $T_B = b.P$

2.2 Background of Elliptic Curve Groups

Let the symbol E/F_p denote an elliptic curve E over a prime finite field F_p, defined by an equation (2,3)

$y^2 = x^3 + ax + b$, $a, b \in F_p$ (1)

and with the discriminate

$\Delta = 4a^3 + 27b^2 \neq 0$ (2)

The points on E/F_p together with an extra point O called the point at infinity form a group

$G = \{(x, y) : x, y \in F_p, E(x, y) = \{G\} \cup \{O\}$ (3)

G is a cyclic additive group in the point addition $"+"$ defined as follows: Let $P, Q \in G$, l be the line containing P and Q (tangent line to E/F_p if $P = Q$), then R is the third point of intersection of l with E/F_p. Let l' be the line connecting R and O. Then $P"+"Q$ is the point such that l' intersects E/F_p at R and O. Scalar multiplication over E/F_p can be computed as follows:

$tP = P + P + ... + P(t\ times)$ (4).

Let the order of G be n. The following problems are commonly used in the security analysis of many cryptographic protocols [4,19].

2.2.1 Computational Diffie-Hellman (CDH) problem

Given a generator P of G and (aP, bP) for unknown $a, b \in_R Z_n^*$, the solution of the CDH problem is to compute abP. For convenience, we define the function cdh as $cdh(aP, bP) = abP$.

2.2.2 Decisional Diffie-Hellman (DDH) problem

Given a generator P of G and (aP, bP, cP) for unknown $a, b, c \in_R Z_n^*$, the solution of the DDH problem is to decide whether the equation $abP = cP$ holds.

2.2.3 Gap Diffie-Hellman (GDH) problem

Given a generator P of G and (aP, bP) for unknown $a, b \in_R Z_n^*$, and an oracle O_{ddhp}, the task of GDH problem is to compute abP, where O_{ddhp} is a decision oracle that on input (aP, bP, cP), answers 1 if $cdh(aP, bP) = cP$; and answers 0, otherwise.

The GDH assumption states that the probability of any polynomial-time algorithm to solve the GDH problem is negligible.

3. SECURITY PROPERTIES FOR TWO-PARTY AUTHENTICATED KEY AGREEMENT PROTOCOLS

The following security properties are commonly required for two-party authenticated key agreement protocols[19, 20]:

3.1 Known Key Security (KKS)

Each run of a key agreement protocol between two parties A and B should produce a unique session key. A protocol should not become insecure if the adversary has learned some of the previous session keys.

3.2 The Key off-set attack (KOA)

An adversary can off-set the agreed session key by an exponent α, which is unknown to both A and B

3.3 Resistance to Disclosure of Ephemeral Secrets (DES)

The protocol should be resistant to the disclosure of ephemeral secrets. The disclosure of an ephemeral secret should not compromise the security of other sessions.

3.4 Partial (or Weak) Forward Secrecy (WFS)

An attacker who knows the private keys of all parties, but is not actively involved in choosing ephemeral keys during the sessions of interest, should not be able to determine previously established session keys.

3.5 Resistance to Key-Compromise Impersonation (KCI) Attacks

If the private key of a user A is compromised, the attacker should not be able to impersonate another user B to A.

3.6 No key control (NKC)

Both participants A and B have an input into the session key neither participant can force the full session key to be a preselected value.

3.7 Resistance to Unknown Key-Share (UKS) Attacks

It should be impossible to coerce A into thinking it is sharing a key with B, when it is actually sharing a key with another (honest) user C (and C correctly thinks the key is shared with A).

4. OVERVIEW ON HAIYAN SUN[1] PROTOCOL

4.1 Set up

This algorithm takes a security parameter k as an input and returns the system parameters and the master secret key. Given k, the KGC does the following steps.

1) The security center chooses a k-bit prime p and determines the tuple $\{F_p, E/F_p, G, P\}$ as defined in Section 2.1.
2) KGC chooses the master secret key $s \in Z_n^*$ and computes the master public key $P_{pub} = s.P$

3) KGC chooses two cryptographic secure hash functions $H_1 : \{0,1\}^* \times G \rightarrow Z_q^*$ and $H_2 : \{0,1\}^{*2} \times G^{10} \rightarrow \{0,1\}^k$
4) KGC publishes the system-wide parameters (params) and keeps the master key s secret.

$params = \{F_p, E/F_p, G, P, P_{pub}, H_1, H_2\}$

4.2 Partial –Private –Key –Extract

This algorithm takes a master key, a user's identifier, system parameters as inputs, and returns the user's ID-based private key. The KGC does the following steps.

1) KGC chooses a random number $r_i \in Z_n^*$, computes $R_i = r_i.P$ and $h_i = H_1(ID_i, R_i)$.
2) KGC computes $s_i = r_i + h_i.s$ and sends $\{s_i, R_i\}$ to the rs through a secret channel.

The user's partial private key is s_i, the user checks his secret key validation by testing the equation $s_i.P = R_i + h_i.P_{pub}$. The private key is valid if the equation holds.

4.3 Set- Secret- Value

The user with ID_i picks randomly $x_i \in Z_n^*$, computes $P_i = x_i.P$ and sets x_i as his secret value.

4.4 Set-Public-Key

The user with identity ID_i has the public key as $P_i = x_i.P$, P_i the public key

4.5 Key Agreement

Suppose that A and B want to establish a session key, they perform the following steps:

1) A chooses a random number $a \in Z_q^*$ and computes $T_A = a.P$, then sends (T_A, R_A, ID_A) to the entity B.
2) After B receives (T_A, R_A, ID_A), B chooses a random number $b \in Z_n^*$ and computes $T_B = b.P$, then sends (T_B, R_B, ID_B) to the entity A. Then, B computes $PK_A = R_A + h_A.P_{pub}$. The session key is computed as
$K = H_2(ID_A \| ID_B \| T_A \| T_B \| P_A \| P_B \| Z_1 \| Z_2 \| Z_3 \| Z_4 \| Z_5 \| Z_6)$
where, $Z_1 = (s_B + x_B)(PK_A + P_A)$,
$Z_2 = (s_B + 2x_B)(PK_A + 2P_A)$,
$Z_3 = (s_B + b)(PK_A + T_A)$, $Z_4 = (s_B - b)(PK_A - T_A)$, $Z_5 = (b + x_B)(P_A + T_A)$ and
$Z_6 = (b + 3x_B)(3P_A + T_A)$
3) Upon receiving (T_B, R_B, ID_B), A computes $PK_B = R_B + h_B.P_{pub}$ and its session key
$K = H_2(ID_A \| ID_B \| T_A \| T_B \| P_A \| P_B \| Z_1 \| Z_2 \| Z_3 \| Z_4 \| Z_5 \| Z_6)$
where, $Z_1 = (s_A + x_A)(PK_B + P_B)$,
$Z_2 = (s_A + 2x_A)(PK_B + 2P_B)$,
$Z_3 = (s_A + a)(PK_B + T_B)$, $Z_4 = (s_A - a)(PK_B - T_B)$, $Z_5 = (a + x_A)(P_B + T_B)$ and
$Z_6 = (a + 3x_A)(3P_B + T_B)$.

Both entities A and B are supposed to compute the same session key.

4.6 Weakness on Haiyan Sun[2] Protocol

The key off-set attack (KOA) is defined by Blake-Wilson [21] as follows: an adversary can off-set the agreed session key by an exponent α, which is unknown to both A and B. All the key agreement protocols [2, 3, 4, 19] without key confirmation are vulnerable to this attack. From the attack, the adversary does not gain any knowledge about the agreed session key, but two entities generate a wrong session key. This is a violation of the key integrity property which indicates that any accepted session key should depend only on inputs from the protocol participants. Now, in Figure 1, how this attack works in Haiyan Sun's protocol is demonstrated.

5. THE PROPOSED ELLIPTIC CURVE KEY AGREEMENT PROTOCOL

The proposed protocol consists of three phases: the registration phase, the transfer phase and the key generation phase. The proposed protocol is described in Figure 2.

5.1 The Registration Phase

- The security center chooses a k -bit prime p and determines the tuple $\{F_p, E/F_p, P\}$ as defined in Section 2.1(notations) then publishes this tuple
- Each entity A, B choose his /her secret key and compute the corresponding public keys as follow:
- For entity A, A chooses $x_A \in [p-1]$ where x_A is the secret key and computes the public key $P_A = x_A.P$
- For entity B, B chooses $x_B \in [p-1]$ where x_B is the secret key and computes the public key $P_B = x_B.P$

5.2 The Transfer Phase

- The entity A chooses a random number $a \in [p-1]$ and computes $T_A = a.P$ then sends T_A to the entity B
- The entity B chooses a random number $b \in [p-1]$ and computes $T_B = b.P$ then sends T_B to the entity A.
- Upon receiving T_B, entity A computes the session key as follows: $K_{AB}^1 = a.T_B + x_A.P_B$ and $K_{AB}^2 = a.T_B$.Then A computes the secret session key $sk = H_1(T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$, the authentication message $AM_{Ask} = H_1(T_A \| T_B \| sk)$ and sends (T_A, AM_{Ask}) to B
- Upon receiving T_A, entity B computes the session key as follows: $K_{BA}^1 = b.T_A + x_B.P_A$ and $K_{BA}^2 = b.T_A$ then computes the secret session key $sk = H_1(T_A \| T_B \| K_{BA}^1 \| K_{BA}^2)$, the authentication message $AM_{Bsk} = H_1(T_A \| T_B \| sk)$ and sends (T_B, AM_{Bsk}) to A. B checks the validation of the shared secret key. If $AM_{Ask} = AM_{Bsk}$ holds, B accept the session key sk, otherwise sends an authentication-failed message to A.

- Upon receiving the entity A (T_B, AM_{Bsk}): A checks the validation of the shared secret key. If $AM_{Ask} = AM_{Bsk}$ holds, A accept the session key sk, otherwise sends an authentication-failed message to B.

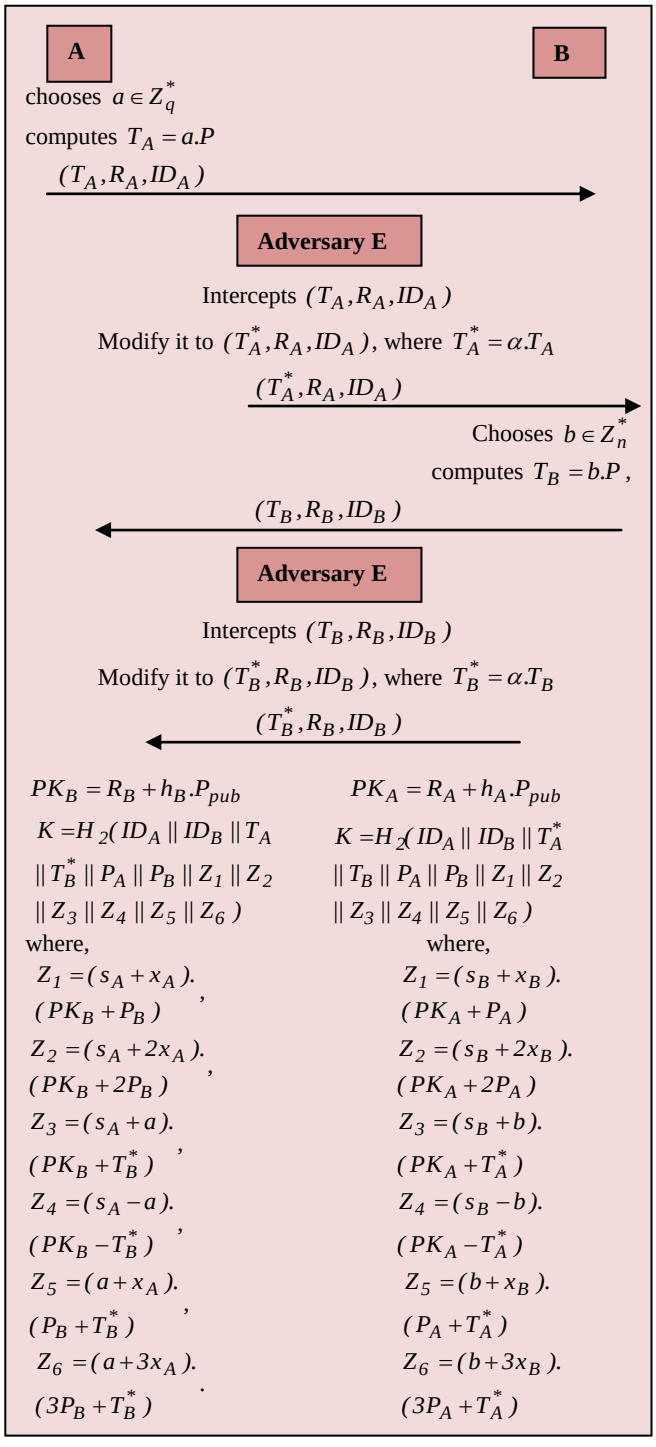

Fig 1: Key off-set attack in Haiyan Sun[1] Protocol

$K_{AB}^1 = a.T_B + x_A.P_B = a.b.P + x_A.x_B.P$ and $K_{BA}^1 = b.T_A + x_B.P_A = b.a.P + x_B.x_A.P = K_{AB}^1$. Also $K_{AB}^2 = a.T_B = a.b.P = K_{BA}^2 = b.a.P$, therefore the secret session key is equal.

$sk = Hash(T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$

$= Hash(T_A \| T_B \| K_{BA}^1 \| K_{BA}^2)$

Also, $AM_{Ask} = AM_{Bsk}$ as both entities share the same secret key.

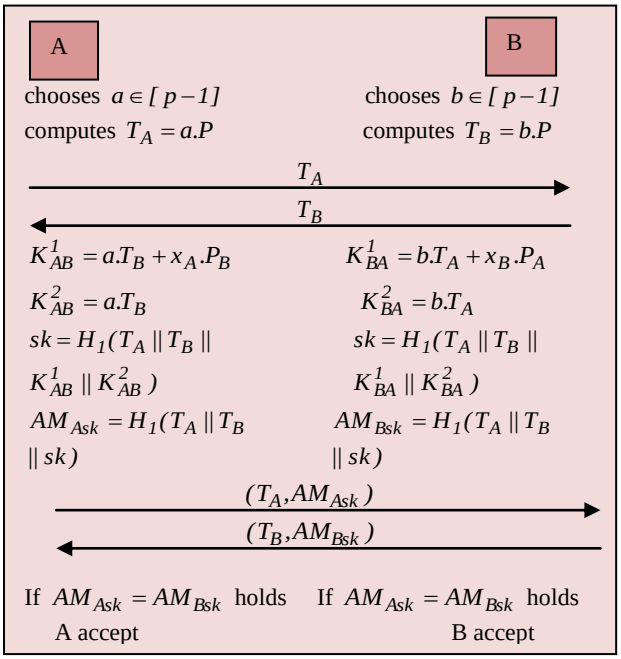

Fig 2: The proposed key agreement Protocol

6. THE PROPOSED CERTIFICATELESS KEY AGREEMENT PROTOCOL

In this section, the above proposed key agreement protocol is extended to be an identity based key agreement protocol. The proposed identity based protocol consists of six polynomial time algorithms. They are described as follows. The proposed protocol described in figure 3.

6.1 Set up

This algorithm takes a security parameter k as an input and returns the system parameters and the master secrete key. Given k KGC does the following steps.

1) The security center chooses a k -bit prime p and determines the tuple $\{F_p, E/F_p, G, P\}$ as defined in Section 2.1(notations)
2) KGC chooses the master secret key $s \in Z_n^*$ and computes the master public key $P_{pub} = s.P$
3) KGC chooses two cryptographic secure hash functions $H_1 : \{0,1\}^* \times G \to Z_n^*$ and $H_2 : \{0,1\}^* \times G \to Z_n^*$
4) KGC publishes as system parameters and keeps the master key s secret.

$params = \{F_p, E/F_p, G, P, P_{pub}, H_1, H_2\}$

6.2 Set- Secret- Value

The user with ID_i picks randomly $x_i \in Z_n^*$, computes $P_i = x_i.P$ and sets x_i as his secrete value.

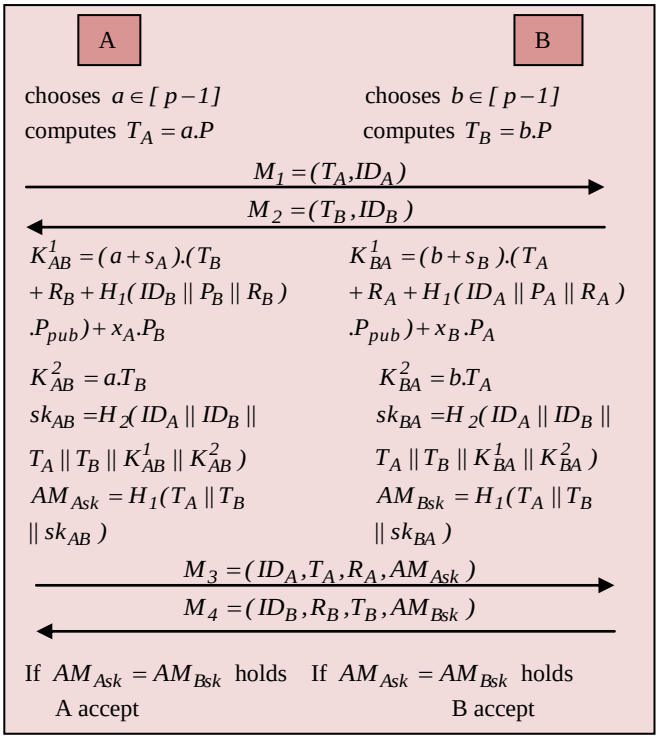

Fig 3: The proposed certificateless KA Protocol

6.3 Partial –Private –Key –Extract

This algorithms takes master key , a user's identifier, system parameters as inputs , and returns the user's ID-based private key. KGC does the following steps.

1) KGC chooses a random number $r_i \in Z_n^*$, computes $R_i = r_i.P$ and $h_i = H_1(ID_i, R_i, P_i)$.
2) KGC computes $s_i = r_i + h_i.s$ and sends $\{s_i, R_i\}$ to the users through secrete channel.

The user's partial private key is s_i, the user checks his secret key validation by testing the equation $s_i.P = R_i + h_i.P_{pub}$. The private key is valid if the equation holds.

6.4 Set-Private-Key

The user with identity ID_i has the secrete key pair $sk_i = (x_i, s_i)$

6.5 Set-Public-Key

The user with identity ID_i has the public key pair $pk_i = (P_i, R_i)$

6.6 Key -Agreement

Let an entity A with identity ID_A has the private key $sk_A = (x_A, s_A)$ and public key $pk_A = (P_A, R_A)$ and an entity B has the private key $sk_B = (x_B, s_B)$ and the public key $pk_B = (P_B, R_B)$ wants to establish a session key , then they can do the following:

1) A choose a random number $a \in Z_n^*$ and computes $T_A = a.P$, then sends $M_1 = (T_A, ID_A)$ to the entity B
2) After B receives M_1, B choose a random number $b \in Z_n^*$ and computes $T_B = b.P$, then sends $M_2 = (T_B, ID_B)$ to the entity B
3) A computes $K_{AB}^1 = (a + s_A).(T_B + R_B + H_1(ID_B \| P_B \| R_B).P_{pub}) + x_A.P_B$ and $K_{AB}^2 = a.T_B$ then $sk_{AB} = H_2(ID_A \| ID_B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$, the authentication message $AM_{Ask} = H_1(T_A \| T_B \| sk_{AB})$ and sends $M_3 = (ID_A, T_A, R_A, AM_{Ask})$ to B
4) B computes $K_{BA}^1 = (b + s_B).(T_A + R_A + H_1(ID_A \| P_A \| R_A).P_{pub}) + x_B.P_A$ and $K_{BA}^2 = b.T_A$ then $sk_{BA} = H_2(ID_A \| ID_B \| T_A \| T_B \| K_{BA}^1 \| K_{BA}^2)$, the authentication message $AM_{Bsk} = H_1(T_A \| T_B \| sk_{BA})$ and sends $M_4 = (ID_B, R_B, T_B, AM_{Bsk})$ to A. B checks the validation of the shared secret key. If $AM_{Ask} = AM_{Bsk}$ holds, B accept the session key sk_{BA} , otherwise sends an authentication-failed message to A.

The shared secret keys agree because

$K_{AB}^1 = (a + s_A).(T_B + R_B + H_1(ID_B \| P_B \| R_B).P_{pub}) + x_A.P_B$

$K_{AB}^1 = (a + s_A).(b.P + r_B.P + H_1(ID_B \| P_B \| R_B).s.P) + x_A.P_B$

$K_{AB}^1 = (a + s_A).(b + r_B + H_1(ID_B \| P_B \| R_B).s).P + x_A.P_B$

$K_{AB}^1 = (a + s_A).(b + s_B).P + x_A.x_B.P$

$K_{AB}^1 = ((a + s_A).(b + s_B) + x_A.x_B).P = K_{BA}^1$ where

$K_{BA}^1 = (b + s_B).(a.P + r_A.P + H_1(ID_A \| P_A \| R_A).s.P) + x_B.P_A$

$K_{BA}^1 = (b + s_B).(a + r_A + H_1(ID_A \| P_A \| R_A).s).P + x_B.x_A.P$

$K_{BA}^1 = ((b + s_B).(a + s_A) + x_B.x_A).P = K_{AB}^1$

And $K_{AB}^2 = a.T_B = a.b.P = K_{BA}^2 = b.T_A = b.a.P$

The agreed session key can be computed as

$sk_{AB} = H_2(ID_A \| ID_B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$

$= sk_{BA} = H_2(ID_A \| ID_B \| T_A \| T_B \| K_{BA}^1 \| K_{BA}^2)$

7. SECURITY ANALYSIS

7.1 Known Key Security (KKS)

In the proposed protocol both entities A and B computes the session key as follow: $sk_{AB} = H_2(ID_A \| ID_B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$ security of which is depends on the secrecy of $K_{AB}^1 = (a + s_A).(T_B + R_B + H_1(ID_B \| P_B \| R_B).P_{pub}) + x_A.P_B = (a + s_A).(b + s_B).P + x_A.x_B.P$

However, if an attacker knows the session ephemeral secrets *a* and *b* , he cannot computes the session key sk_{AB} He can

generate the session key if he knows $x_A.x_B.P$, $s_A.s_B.P$, $a.s_B.P$ and $b.s_A.P$ but this is impossible due to difficulties of solving the CDHP problem. Therefore, the proposed protocol is strong against Known Key Security attack.

7.2 Key off-set attack (KOA)

In the proposed protocol, after sending the user A the message $M_1 = (T_A, ID_A)$ to B. An attacker E modifies it to $M_1 = (T_A^*, ID_A)$ where $T_A^* = \alpha.T_A$. The user B receives T_A^* and computes: $K_{BA}^{1*} = (b + s_B).(T_A^* + R_A + H_1(ID_A \| P_A \| R_A).P_{pub}) + x_B.P_A$, $K_{BA}^{2*} = b.T_A^*$, $sk_{BA}^* = H_2(ID_A \| ID_B \| T_A^* \| T_B \| K_{BA}^{1*} \| K_{BA}^{2*})$. Then computes authentication message $AM_{sk}^* = H_1(T_A^* \| T_B \| sk_{BA}^*)$, $M_4^* = (ID_B, T_B, R_B, AM_{sk}^*)$ and sends M_4^* to A. Again, An attacker E modifies it to T_B^* where $T_B^* = \alpha.T_B$ but does not change the $AM_{sk}^* = H_1(T_A^* \| T_B \| sk_{BA}^*)$ because he has no ability to compute it without B's secret key. Now the user A computes $K_{AB}^{1*} = (a + s_A).(T_B^* + R_B + H_1(ID_B \| P_B \| R_B).P_{pub}) + x_A.P_B$ and the session key: $K_{AB}^{2*} = a.T_B^*$, $sk_{AB}^{**} = H_2(ID_A \| ID_B \| T_A \| T_B^* \| K_{AB}^{1*} \| K_{AB}^{2*})$, $AM_{sk}^{**} = H_1(T_A \| T_B^* \| sk_{AB}^{**})$ which does not equal $AM_{sk}^* = H_1(T_A^* \| T_B \| sk_{BA}^*)$ and therefore, user A rejects the session key agreement and sends an authentication-failed message to B. Thus, the key off-set attack is not possible against the proposed protocol.

7.3 Resistance to Disclosure of Ephemeral Secrets (DES)

If any session key is exposed to an attacker it does not mean that other session keys are also exposed. In the proposed protocol, the agreed session key sk depends on two random ephemeral secrets a,b and these are generated in each session. The only way to derive a and b is from T_A and T_B but due to the difficulties of ECDLP problem , it is impossible. So driving one session key does not allow the attacker to gain the knowledge about other session keys.

7.4 Perfect Forward Secrecy(WPFS)

If the attacker compromise the secret keys of A and B , he cannot recover any past session keys. The attacker E may compute the session key sk if he knows $K_{AB}^1 = ((a + s_A).(b + s_B) + x_A.x_B).P$. Suppose an attacker compromise x_A and x_B, he may compute $x_A.x_B.P$, but not $a.b.x_A.x_B.P$ as a and b are unknown to E .He can try to derive a and b from T_A and T_B, but this is impossible due to difficulties of solving the CDHP problem. Moreover, E tries to derive $a.x_B.P$ from $(T_A, P_B) = (a.P, x_B P)$ and $b.x_A.P$ from $(T_B, P_A) = (b.P, x_A.P)$ directly and then $a.b.x_A.x_B.P$ from them. However, these are also not possible due to hardness of CDHP problem. Thus, the perfect forward security and PKG forward security are preserved in the proposed protocol.

7.5 Resistance to Key-Compromise Impersonation (KCI) Attacks

Suppose that $A's$ secret key x_A is disclosed by an attacker, and then he tries to impersonate B to A to obtain the resulting session key. The attacker intercepts the $A's$ message $M_1 = (T_A, ID_A)$ and then computes $T_B = b'.P$ (b' is selected by the attacker) but he cannot computes $K_{BA}^1 = ((b + s_B).(a + s_A) + x_B.x_A).P$, because with known T_A, random number b and $A's$ secret key, he cannot computes K_{BA}^1 without the knowledge of a, s_A, s_B, x_B. Thus, generating a session key by an attacker is impossible and the proposed protocol is strong against the key-compromise impersonation attacks

7.6 No key control (NKC)

In the proposed protocol, both participants A and B have an input into the session key neither participant can force the full session key to be a preselected value. The session key in our protocol is determined jointly by both participants A and B. Thus $sk_{AB} = H_2(ID_A \| ID_B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$ depends on T_A , T_B, R_B, P_{pub} and P_B where $K_{AB}^1 = (a + s_A).(T_B + R_B + H_1(ID_B \| P_B \| R_B).P_{pub}) + x_A.P_B$, $K_{AB}^2 = a.T_B$ and these are generated by A and B respectively. No way for any user to control the session key only by himself.

7.7 Resistance to Unknown Key-Share (UKS) Attacks

Following the proposed protocol, the session key is generated not only using K_{AB}, also the identities ID_A, ID_B of the participants and other session dependent tokens T_A and T_B. The proposed protocol is secure against unknown key-share attack.

7.8 The Comparative Study

The proposed identity based key agreement protocol is compared with the protocols by Debiao He [4] (2011) , Debiao He [3] (2012) and Haiyan Sun [2] (2013) from the computational cost , number of exchanged messages and the key offset attack where $T_{EC\text{-}mult}$ is the time required for executing multiplication operation on elliptic curve E, $T_{EC\text{-}add}$ is the time required for executing addition operation on elliptic curve E and T_h is the time required for executing one way dispersed row function operation. Table 1 specify this comparative study in details.

Table.1. the proposed protocol is compared with the protocols in [2, 4 ,3]

The protocol	Computational cost	Key off-set attack (KOA)	No of exchanged messages
Debiao He [4] (2011)	$5T_{EC\text{-}mult} + 4T_{EC\text{-}add} + 2T_h$	No	2
Debiao He [3] (2012)	$5T_{EC\text{-}mult} + 3T_{EC\text{-}add} + 2T_h$	No	2
Haiyan Sun [2] (2013)	$10T_{EC\text{-}mult} + 7T_{EC\text{-}add} + 1T_h$	No	2
The proposed protocol	$5T_{EC\text{-}mult} + 3T_{EC\text{-}add} + 2T_h$	Yes	4

8. IMPLEMENTATION

8.1 Domain Parameter Specification

In this section, the elliptic curve and the domain parameters proposed are specified in the following way.

The elliptic curve equation specified in section 2.2. where the parameters

p : is the prime specifying the base field.

a,b : are the coefficients of the equation $y^2 = x^3 + ax + b$ defining the elliptic curve.

$G=(x,y)$: is the base point, i.e., a point in E of prime order, with x and y being its x- and y-coordinates, respectively.

q : is the prime order of the group generated by G.

8.2 Sample Run and its Results

8.2.1 Parameters

p 768849563970453442208097466290016490930379502009430552037356014450315161977 51

a 566981876053261100436272283961783460771206145394752141093868281887638841399 93

b 175772324973218388410756977897945202629504260589230845670468523006333254389 02

x 632437297495623333552922435503129703347781755710547265870953816236271441147 86

y 382186150937535238931222779640308103875854055397726025815578318874857179979 75

q 768849563970453442208097466290016490927375317844145295387555190630635363590 79

s 531456252600089409860660556691090785271146020341382606497902963412365278996 16

P_{pub} {773955272622261281505002627296549212958259238484808867339922169776988198408, 655303213116586075170847239742133317390702038195676208013861119836184508806 9}

8.2.2 Set- Secret- Value

ID_A 716876822973309685053679354228999693204146621015418869652664491989891928349 04

ID_B 719875351925388004776163443793932794988112778206413961004990303433011878010 69

x_A 369709170576049953921631160892745389737846388993741055444226160726869174344 64

x_B 673880757059829922205494716631909377018724965248458914851286295471857466369 07

P_A {772233176565615167596376690589476860967509418415196940616540317716645405119 9 ,449575966017293842144243795274946366200540146453744569973187697373528198317509}

P_B {787703484270028815543966489688285063271143586658709442648391707035657171555 8, 522177263058339921671103539800707065942548047486769390144248410411138374836 48}

8.2.3 Partial –Private –Key –Extract

R_A {314169454325786797840651879609148627570600779440487417999003585410137230578 86, 763318529437577440784779303997459587879644738876762360751065211105754521024 13}

R_B {321174742753041001090213921502670566773986213568100275292925104817701968046 28, 389957626176998814366521073332596443797867574511051338119258314179426213274 36}

h_A 374982247

h_B 353954023

s_A 237285327057485321088463778415436249057354435306404186183260597382688060263 76

s_B 719280106400288117079612494898716807569140220568438767269966359650242133295 24

8.2.4 Key –Agreement

For entity A

T_A {723081766906889475837353000808181994279628910856346151533017806348669598413, 395748470839953908265919587919536130609557713694317811425026172031292410999 95}

K_{AB}^1 {366212463214993221122040586846141504525960806286601287876718501257807056066 75,24679290580813554409735580716781467943531543260882775004281522300813647 96}

K_{AB}^2 {432832149704838055997454945957323111429923550031451020611194565912617063069 51, 342508123567697958488398602786701706910431273981876452527844412973908112325 56}

sk_{AB} 1074612477

AM_{AB} 525601355

For entity B

T_B {425649091864075935185735748971166335095198054401947282524759385003285591595 36, 506865709964088117395535128012953827596317834516840543280402723652406949972 510}

K_{BA}^1 {366212463214993221122040586846141504525960806286601287876718501257807056066 75, 246792905808135544097355807167814679435311767543260882775004281522300813647 96}

K_{BA}^2 {432832149704838055997454945957323111429923550031451020611194565912617063069 51 ,342508123567697958488398602786701706910431273981876452527844412973908112325 56}

sk_{BA} 1074612477

AM_{BA} 525601355

It obvious from the Implementation that both entities share the same session key

9. CONCLUSION

In this paper, a security flaw of Haiyan Sun's protocol has been identified and then a modification has been proposed to design a new paring-free key agreement protocol based on elliptic curve cryptography. A certificateless key agreement protocol without bilinear pairings is proposed which is an extension to the first proposed one. The security analysis of the proposed paring free this key agreement protocol has been discussed. It is found that the proposed protocol achieves all security requirements and avoids the key-off set attack. The performance of the proposed protocol is compared with other protocols and it is found that the proposed protocol is requires computations with improved security properties. The proposed protocol has been implemented using the Mathematica(7) program.

10. REFERENCES

[1] F. Ahmed and Dalia Elkamchouchi ," *A New Efficient Protocol For Authenticated Key Agreement*", International Journal of Computer and Communication Engineering, Vol. 2, No. 4, July 2013.

[2] H. Sun, Q. Wen, H. Zang and Z. Jin, " *A Strongly Secure Pairing –Free Certificateless Authenticated Key Agreement Protocol For Low –Power Devices*", ISSN 1392-124X, ISSN 2335-884X (online) Information Technology and control , 2013,Vol.42 ,No. 2.

[3] D. He , S. Padhye and J. Chen," *An Efficient Certificateless Two-Party Authenticated Key Agreement Protocol*", Computers & Mathematics with Applications , Volume 64, Issue 6, September 2012, Pages 1914–1926

[4] D. He, and Y. Chen, " *An Efficient Certificateless Authenticated Key Agreement Protocol Without Bilinear Pairings*", Mathematical and Computer Modelling, 54 (11-12), 3143-3152, 2011

[5] A. Shamir, "*Identity-Based Cryptosystems And Signature Protocols*", Proc. CRYPTO1984, LNCS, vol.196, 1984, pp.47–53.

[6] S. Al-Riyami, and K.G. Paterson, "*Certificateless Public Key Cryptography*", Proceedings of ASIACRYPT 2003, LNCS 2894, Springer-Verlag, 2003, pp. 452–473.

[7] Z. Shao," *Efficient Authenticated Key Agreement Protocol Using Self-Certified Public Keys From Pairings*", Wuhan University Journal of Natural Sciences, 10(1):267-270, 2005.

[8] S. Wang, Z. Cao, and X. Dong, *"Certificateless Authenticated Key Agreement Based On The MTI/C0 Protocol*", Journal of Information and Computational Science 3 (2006) 575–581.

[9] Y. Shi, and J. Li, " *Two-Party Authenticated Key Agreement In Certificateless Public Key Cryptography*", Wuhan University Journal of Natural Sciences 12 (1) (2007) 71–74.

[10] T. Mandt, and C. Tan, " *Certificateless Authenticated Two-Party Key Agreement Protocols*", in the Proceedings of the ASIAN 2006, LNCS, vol. 4435, Springer-Verlag, 2008, pp. 37–44.

[11] C. Swanson, "*Security In Key Agreement: Two-Party Certificateless Schemes*", Master Thesis, University of Waterloo, 2008.

[12] G. Lippold, C. Boyd, and J. Nieto, "*Strongly Secure Certificateless Key Agreement*", In Pairing 2009, pages 206-230.

[13] L. Zhang, F. Zhang, Q. Wua, and J. Domingo-Ferrer, "*Simulatable Certificateless Two-Party Authenticated Key Agreement Protocol*", Information Sciences 180 (2010) 1020–1030. 16

[14] L. Chen, Z. Cheng, and N.P. Smart, "*Identity-Based Key Agreement Protocols From Pairings*", Int. J. Inf. Secur., 6(2007) pp.213–241.

[15] M. Geng and F. Zhang, "*Provably Secure Certificateless Two-Party Authenticated Key Agreement Protocol Without Pairing*", In International Conference on Computational Intelligence and Security, pages 208-212, 2009.

[16] M. Hou and Q. Xu," *A Two-Party Certificateless Authenticated Key Agreement Protocol Without Pairing*", In 2nd IEEE International Conference on Computer Science and Information Technology, pages 412-416, 2009.

[17] G. Yang, and C. Tan," *Strongly Secure Certificateless Key Exchange Without Pairing*", In the 6th ACM Symposium on Information, Computer and Communications Security, 71-79, 2011.

[18] D. He, J. Chen, and J. Hu," *A Pairing-Free Certificateless Authenticated Key Agreement Protocol*", International Journal of Communication Systems, DOI: 10.1002/dac.1265, 2011.

[19] Y. Kim , Y. Kim , Y. Choe and H. Chol O, " *An Efficient Bilinear Pairing-Free Certificateless Two-Party Authenticated Key Agreement Protocol In The Eck Model*", KISU-MATH-2013-E-R-016: Version 4

[20] S. H. Islama and G. P. Biswas, " *An Improved Pairing-Free Identity-Based Authenticated Key Agreement Protocol Based On ECC*", International Conference on Communication Technology and System Design 2011.

[21] S. Blake-Wilson, D. Johnson, and A. Menezes, " *Key Agreement Protocols And Their Security Analysis*", Proc. of the 6th IMA International Conference on Cryptography and Coding, LNCS, Springer-Verlag, 1997; 1335:30−45