*electronics*

MDPI

*Article*

# An Anonymous Device to Device Authentication Protocol Using ECC and Self Certified Public Keys Usable in Internet of Things Based Autonomous Devices

**Bander A. Alzahrani [1,*], Shehzad Ashraf Chaudhry [2], Ahmed Barnawi [1], Abdullah Al-Barakati [1] and Taeshik Shon [3,*]**

[1] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; ambarnawi@kau.edu.sa (A.B.); aaalbarakati@kau.edu.sa (A.A.-B.)

[2] Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Avcılar, 34310 Istanbul, Turkey; sashraf@gelisim.edu.tr or ashraf.shehzad.ch@gmail.com

[3] Department of Cyber Security, Ajou University San 5, Woncheon-Dong, Yeongtong-Gu, Suwon 443-749, Korea

* Correspondence: baalzahrani@kau.edu.sa (B.A.A.); tsshon@ajou.ac.kr (T.S.)

Received: 13 February 2020; Accepted: 17 March 2020; Published: 21 March 2020

check for updates

**Abstract:** Two party authentication schemes can be good candidates for deployment in Internet of Things (IoT)-based systems, especially in systems involving fast moving vehicles. Internet of Vehicles (IoV) requires fast and secure device-to-device communication without interference of any third party during communication, and this task can be carried out after registration of vehicles with a trusted certificate issuing party. Recently, several authentication protocols were proposed to enable key agreement in two party settings. In this study, we analyze two recent protocols and show that both protocols are insecure against key compromise impersonation attack (KCIA) as well as both lack of user anonymity. Therefore, this paper proposes an improved protocol that does not only resist KCIA and related attacks, but also offers comparable computation and communication. The security of proposed protocol is tested under formal model as well as using well known Burrows–Abadi–Needham (BAN) logic along with a discussion on security features. While resisting the KCIA and related attacks, proposed protocol also provides comparable trade-of between security features and efficiency and completes a round of key agreement in just 13.42 ms, which makes it a promising candidate to be deployed in IoT environments.

**Keywords:** Internet of Things; V2V Security; Internet of Vehicles; key compromise impersonation attack; 2PAKA

## 1. Introduction

A Two-Party Authentication Key Agreement Protocol (2PAKA) shares a secret key after authentication for secure communication between two parties. The certificate based 2PAKA can be deployed in Internet of Things (IoT)-based vehicular environments to offer autonomous device to device communication because in such dynamic and fast moving devices network, the interference of some gateway or trusted authority may lead to delay, and such delays may lead to infeasibility of the whole network [1]. In 2PAKA systems, the vehicle, after registering with the trusted certificate generation authority, gets a private and public key pair based credentials of both trusted authority and the requesting vehicle. However, the security and privacy of such schemes remain on stake due to open architecture beneath the communication. Such architecture is shown in Figure 1, involving the smart devices networks and the certificate authority which can also termed as server. Every device in a smart network gets its key pair from certificate authority and then can communicate autonomously without involvement of the authority. In this article the term device and vehicle are used interchangeably as well as server and certificate authority means same.

Diffie & Hellman key exchange protocol [2] was the first approach in this direction. After then, several key exchange protocols [3–6] based on traditional public key infrastructure (PKI) were proposed to avoid man-in-middle (MIM) attack. The use of modular exponentiation in PKI led towards PKI's inapplicability in resource constrained environments like smart phones, smadrcards etc. Therefore, research efforts then have focused on lightweight Elliptic Curve Cryptography (ECC) and some 2PAKA protocols based on ECC [7–9] were proposed. The ECC-based 2PAKA protocols require less computation and storage with same level of security, due to the use of 160 bits key in ECC instead of 1024 bits key in Rivest, Shamir, and Adleman (RSA) algorithm . The ECC-based 2PAKA protocols require a trusted third party, called certificate authority(CA), to manage and generate certificates. It also validates and generates public keys of users.

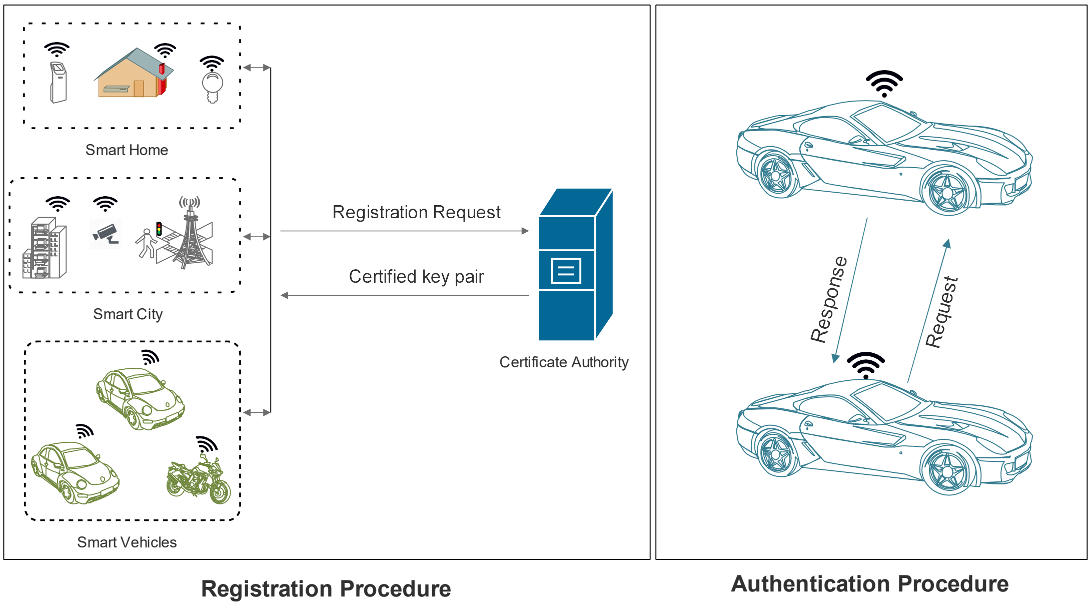


**Figure 1.** Device to Device Authentication Scenario.

In 1989, Gunther et al. [10] proposed a key exchange protocol based on user's identity. The protocol in [10] requires the intervention of certificate authority for establishing a secure channel between two users. In 2000 Saeedina [11] proposed the improvement over Gunther et al.'s identity-based key exchange protocol. The modified scheme overcomes the number of passes to half, and so minimize the communication between the parties. In 2002, Hsieh et al. [12] proposed a slight modification of Saeednia's identity-based key exchange protocol to reduce computation cost. However, Tseng et al. [9] demonstrated that the scheme proposed by Hiesh et al. cannot withstand key compromise impersonation attack (KCIA). Holbl and Welzer [13] proposed two new two-party identity-based authenticated key agreement protocols.The first is based on the protocol of Hsieh et al. to make it immune against KCIA, while the second is an efficient enhancement of Tseng's protocol. Zhang et al. [14] proved that the protocols proposed in [13] cannot resist impersonation attack as well as KCIA. Smart [15] proposed another identity based key agreement protocol using weil pairing. Chen and Kudla [16] and Shim [17] independently purposed authenticated key agreement (AKA) protocols. Sun and Hsieh [18] proved that both the protocols [16,17] are vulnerable to KCIA and man-in-middle (MIM) attacks. Ryu et al. [19] also proposed another protocol and demonstrated that their protocol minimizes the cost of computation and communication and is more efficient than Chen and Kudla's protocol with same security properties. Boyd and Choo [20] showed that the Ryu et al.'s protocol could not achieve the KCIA resilience properties. McCullagh and Barreto [21] claimed that their protocol can be used in either escrow or escrow-less mode. They also described conditions under which users of different key generation centers can agree on a shared secret key. In 2005 Zu-hua et al. [22] proposed bilinear pairing based self-certified protocol using computational Diffie-Hellman assumption. Ni et al. [23] also presented two secure variants of their proposal.

In 2008 Cao et al. [24] put forwarded a new identity-based authentication key agreement protocol and claimed it to achieve forward secrecy. Tsaur [25] also proposed an ECC-based self-certified public key cryptosystem based AKA and their protocol achieved session and public keys in a single step. In 2009 Hölbl and Welzer [26] proposed two new identity-based 2PAKA protocols but their scheme were proved to be vulnerable to key compromise impersonation attacks. Their protocol do not offer provable security. Some other IBC-based 2PAKA protocols using ECC were also proposed [9,11–13,16–21,27–30], these protocols suffer from private key escrow problem because the private key is known as Private Key Generation (PKG) party. If the PKG is malicious with man-in-middle (MIM) attack then the whole protocol is suffered [31].

### *Motivations and Contribution*

In 2015, Islam & Biswas (Islam-Biswas) [31] proposed a self certified ECC based key agreement protocol and claimed that their protocol provides security against all kinds of attacks. Mandal et al. [32] found that their protocol lacks anonymity and is defenseless against replay and clogging attacks [33]. However, in this paper we show that both the protocols of Islam-Biswas and Mandal et al. are insecure against key compromise impersonation attack (KCIA). Moreover, both protocols lack user anonymity. This paper then introduces a new scheme to overcome the insecurities of Islam-Biswas and Mandal et al.'s protocols. The proposed protocol achieves following merits:

1. Proposed protocol resists KCIA and related attacks under the hardness assumption of Elliptic Curve Discrete Logarithm Problem ($ECDLP$).
2. Proposed protocol achieves low computation and communication cost as compared with related secure schemes.

## 2. Fundamentals

This section describes some fundamental concepts relating to Hash Functions, Elliptic Curve Cryptography along with some hard problems. The adversarial model is also defined in this section. Moreover, notation guide is provided in Table 1.

**Table 1.** Notation Guide.

| Notation | Definition |
|---|---|
| $\mathcal{U}_x$,$\mathcal{S}$ | User $x$, Server |
| $\mathcal{D}_a$,$\mathcal{D}_b$ | Device $a$ and Device $b$ |
| $ID_x$, $F_p$ | Identity of $\mathcal{U}_x$, Prime Field |
| $E/F_p$, $G$ | Elliptic Curve over $F_p$, Base Point over $E/F_p$ |
| $K_{Pri}$, $K_{Pub}$ | Private and public key pair of $\mathcal{S}$ |
| $E_{ki}$, $D_{ki}$ | Encryption, Decryption using $ki$ as key |
| $\|\|$, $\oplus$ | Concatenation and Exclusive-Or operations |
| $h(.)$, $H(.)$, $H_i(.)$ | Hash Functions |
| $\overset{?}{=}$ | Equality Checking operator |

### *2.1. Hash Function*

The arbitrary size input $S_a$ to a hash function $H : \{0,1\}^* \rightarrow Z_q^*$ with collision resistant property yields a fixed length value $F_h = H(S_a)$ with following additional pre-requisit properties:

- A slight fluctuation in $S_a$ (the input), there is a massive change in output $F_h = H(S_a)$.

- Computing $F_h$, given $S_a$ is easy; whereas, computing $S_a$, given $F_h$ is a hard problem
- Finding a pair $\{S_a, S_b\}$ such that $H(S_a) = H(S_b)$ is a hard problem and this property is termed as collision resistance property (CRP).

**Definition 1.** *[CRP for Secure Hash] Given $H(.)$, an attacker $\mathcal{A}$ can compute an input pair $\{S_a, S_b\}$ such that $H(S_a) = H(S_b)$ with probability $Advg^{HASH}_{\mathcal{A}}(t) = P[(S_a, S_b) \Leftarrow_r \mathcal{A} : (S_a \neq S_b)$ and $H(S_a) = H(S_b)]$. $\mathcal{A}$ is considered to select the pair at random. The computed advantage is based on polynomial-time t bound arbitrary choices. As per CRP $Advg^{HASH}_{\mathcal{A}}(t) \leq \epsilon$ for $\epsilon > 0$.*

### 2.2. Elliptic Curve Cryptography

Consider $p$ (a very large prime, (160 $bits \leq |p|$), an Elliptic Curve $EC$: $j^2 = i^3 + \alpha i + \beta \mod p$ is a set with finite points $E_p(\alpha, \beta)$. The pair $\{\alpha, \beta\}$ is pragmatically selected to satisfy the relationship $(4\alpha^3 + 27\beta^2) \mod p \neq 0$. The point $W$ multiplication with some chosen scalar $a$ can be computed as $a.W = \{W + W + ......... + W\}$ $a$ times addition repeatedly. All system parameters are chosen from finite field $F_p$; whereas, $EC$ forms an abelian group with point $O$ considered to be at infinity and described as additive identity.

**Definition 2.** *[ Discrete logarithm problem for EC (ECDLP)]* Consider $\{V, W\}$ are two points over $E_p(\alpha, \beta)$ such that $V = aW$, knowing the duo $\{(V = aW, W)\}$, the probability of computing $a$ can be solicited as: $Advg^{ECDLP}_{\mathcal{A}}(t) = P[(\mathcal{A}(V = aW, W) = a : a \in Z_p]$, the experiment is allowed to be conducted by a polynomial-time $t$ bound attacker $\mathcal{A}$. As per $ECDLP$, $Advg^{ECDLP}_{\mathcal{A}}(t) \leq \epsilon$.

**Definition 3.** *[ Diffie Hellman problem for EC (ECDHP)]* Consider $\{V, W, G\}$ are three points over $E_p(\alpha, \beta)$ such that $V = aG$, $W = bG$ and knowing the trio $\{(V = aG, W = bG), G\}$, the probability of computing $X = abG$ can be solicited as: $Advg^{ECDHP}_{\mathcal{A}}(t) = P[(\mathcal{A}(V = aG, W = bG, G) = \{a, b\} : (a, b) \in Z_p]$, the experiment is allowed to be conducted by a polynomial-time $t$ bound attacker $\mathcal{A}$. As per $ECDHP$, $Advg^{ECDHP}_{\mathcal{A}}(t) \leq \epsilon$.

### 2.3. Attacker Model

The authenticated key agreement is achieved over an insecure networks, assuming a strong attacker having many capabilities [34,35]. Some common assumptions related with attackers' capabilities are made as follows:

- The adversary $\mathcal{A}$ is having access to public keys of both parties.
- $\mathcal{A}$ knows public identities of all users of the system.
- $\mathcal{A}$ can control the insecure communication channel, precisely $\mathcal{A}$ can eavesdrop, inject, delete or replay any message, while $\mathcal{A}$ can not have any access to secure channel.

## 3. Review of Islam-Biswas Protocol

In this section, we review Islam-Biswas 2PAKA protocol [31] consisting of three phases: system setup, registration and authenticated key agreement phase, the detail of each phase is as follows:

### 3.1. System setup Phase

In system setup phase, the server ($\mathcal{S}$) initializes the system parameter $\Omega$. Initially $\mathcal{S}$ chooses a security parameter $k \in Z^+$ along with an elliptic curve $E/F_p$, then $\mathcal{S}$ selects a base point $G$ over $E/F_p$. Further $\mathcal{S}$ selects $K_{Pri}$ as his private key and computes $K_{Pub} = K_{Pri}G$ and chooses three one-way hash functions $H_0, H_1, H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$. Finally $\mathcal{S}$ publishes all public parameters $\Omega = \{E/F_p, H_0, H_1, H_2, G, K_{Pub}\}$ and keeps $K_{Pri}$ secret.

### *3.2. Registration Phase*

This phase is executed when a user $\mathcal{U}_a$ wants to register with server. $\mathcal{U}_a$ selects his identity $ID_a$ and a random number $x_a \in_R Z_p^*$, then $\mathcal{U}_a$ computes $X_a = H_0(ID_a \| x_a)G$ and sends $ID_a, X_a$ to $\mathcal{S}$ via some secure channel, which selects $t_a \in_R Z_p^*$ upon receiving a message from $\mathcal{U}_a$. $\mathcal{S}$ then computes $P_a = H_0(ID_a \| t_a)K_{Pub} + X_a$, $r_a = [H_0(ID_a \| t_a) + H_0(ID_a \| P_a)]K_{Pri}$ and $Q_a = P_a + H_0(ID_a \| P_a)K_{Pub}$. $\mathcal{S}$ sends $(ID_a, P_a, r_a)$ to $\mathcal{U}$ via some secure channel and publishes $Q_a$. Upon receiving, $\mathcal{U}_a$ computes his private key $d_a = [r_a + H_0(ID_a \| x_a)]$, the public key of $\mathcal{U}_a$ is $d_a G = Q_a$.

### *3.3. Authenticated Key Agreement Phase*

This phase takes place when two users say $\mathcal{U}_i$ and $\mathcal{U}_j$ want to exchange information and $\mathcal{U}_i$ initiates the process. The following steps as shown in Figure 2 are performed among $\mathcal{U}_i$ and $\mathcal{U}_j$.

IKA 1: $\mathcal{U}_i \rightarrow \mathcal{U}_j : m_j\{ID_i, T_i, R_i\}$
$\mathcal{U}_i$ selects $x \in_R Z_p^*$ and computes $T_i = xQ_i$ & $R_i = H_1(T_i \| d_i Q_j)$, $\mathcal{U}_i$ then sends $ID_i, T_i, R_i$ to $\mathcal{U}_j$.

IKA 2: $\mathcal{U}_j \rightarrow \mathcal{U}_i :\ m_i = \{ID_j, T_j, R_j\}$
$\mathcal{U}_j$ selects $y \in_R Z_p^*$ and computes $T_j = yQ_j$ & $R_j = H_1(T_j \| d_j Q_i)$, $\mathcal{U}_j$ then sends $ID_j, T_j, R_j$ to $\mathcal{U}_i$.

IKA 3: Now the authenticated key is computed as follows:

1. $\mathcal{U}_i$ computes $R_j^* = H_1(T_j \| d_i Q_j)$ and verifies $R_j^* \stackrel{?}{=} R_j$, if not true, $\mathcal{U}_i$ aborts the session, otherwise the key is computed as: $K_i = (xd_i)T_j = xyd_i d_j G$.
2. Similarly $\mathcal{U}_j$ computes $R_i^* = H_1(T_i \| d_j Q_i)$ and verifies $R_i^* \stackrel{?}{=} R_i$, if not true, $\mathcal{U}_j$ aborts the session, otherwise the key is computed as: $K_j = (yd_j)T_i = xyd_i d_j G$.

| *User $\mathcal{U}_i$* | *User $\mathcal{U}_j$* |
|---|---|
| Select $x \in_R Z_p^*$<br>Compute $T_i = xQ_i$<br>Compute $R_i = H_1(T_i \| d_i Q_j)$ | |
| $\xrightarrow{m_x=\{ID_i,T_i,R_i\}}$ | |
| | Select $y \in_R Z_p^*$<br>Compute $T_j = yQ_j$<br>Compute $R_j = H_1(T_j \| d_j Q_i)$ |
| $\xleftarrow{m_y=\{ID_j,T_j,R_j\}}$ | |
| Compute $R_j^* = H_1(T_j \| d_i Q_j)$<br>Check $R_j^* \stackrel{?}{=} R_j$<br>$K = K_i = (xd_i)T_j = xyd_i d_j G$<br>Session key<br>$SK = H_2(ID_i \| ID_j \| T_i \| T_j \| R_i \| R_j \| K)$ | Compute $R_i^* = H_1(T_i \| d_j Q_i)$<br>Check $R_i^* \stackrel{?}{=} R_i$<br>$K = K_j = (yd_j)T_i = xyd_i d_j G$<br><br>$SK = H_2(ID_i \| ID_j \| T_i \| T_j \| R_i \| R_j \| K)$ |

**Figure 2.** Islam-Biswas Key Agreement Protocol.

## 4. Review of Mandal et al.'s Protocol

In this section, we review Mandal et al.'s 2PAKA protocol [32] consisting of three phases: system setup, registration and authenticated key agreement phase. The system setup phase is as it is taken from Islam-Biswas protocol, except Mandal et al. just selected one hash function $H(.)$ instead of three in Islam-Biswas protocol. The detail of other two phases is as follows:

### *4.1. Registration Phase*

This phase is executed when a user $\mathcal{U}_a$ wants to register with server. $\mathcal{U}_a$ selects his identity $ID_a$ and a random number $x_a \in_R Z_p^*$, then $\mathcal{U}_a$ computes $X_a = H(ID_a \| x_a)G$ and sends $\{ID_a, X_a\}$

to $\mathcal{S}$ via some secure channel, which selects $k_a \in_R Z_p^*$ upon receiving a message from $\mathcal{U}_a$. $\mathcal{S}$ then computes $V_a = H(ID_a \| k_a)K_{Pri}$, $TID_a = X_a \oplus V_a$, $W_a = X_a \oplus k_a G$ and $X_{sa} = H(TID_a \| W_a)K_{Pri} \oplus k_a$. $\mathcal{S}$ sends $\{ID_a, TID_a, W_a, X_{sa}\}$ to $\mathcal{U}_a$ via some secure. Upon receiving, $\mathcal{U}_a$ computes his private key $d_a = X_{sa} \oplus H(ID_a \| x_a)$, and public key $Q_a = d_a G$. $\mathcal{U}_a$ checks the validity/correctness of public private key pair as $d_a.G \stackrel{?}{=} [H(TID_a||W_a)K_{Pub} \oplus W_a]$. On successful verification, $\mathcal{U}_a$ keeps $d_a$ secret and publishes $Q_a$.

### 4.2. Authenticated Key Agreement Phase

This phase takes place when two users say $\mathcal{U}_i$ and $\mathcal{U}_j$ want to exchange information and $\mathcal{U}_i$ initiates the process. The following steps as shown in Figure 3 are performed among $\mathcal{U}_i$ and $\mathcal{U}_j$.

MKA 1: $\mathcal{U}_i \rightarrow \mathcal{U}_j : m_i = \{N_i, t_i, C_i\}$

$\mathcal{U}_i$ selects $N_i \in_R Z_p^*$, generate $t_i$ and computes $W_1 = TID_i \oplus W_i$, $Z_i = H(x_i)$, $Key_i = H(d_i Q_j||N_i||t_i)$, $M_1 = H(W_1||Key_i||N_i||Z_1)$ and $Z_1 = Z_i \oplus M_1$. $\mathcal{U}_i$ then compute encryption as : $C_i = E_{Key_i}(TID_i||M_1||Z_1||W_i||N_i||t_i)$ and sends $m_i = \{N_i, t_i, C_i\}$ to $\mathcal{U}_j$.

MKA 2: $\mathcal{U}_j \rightarrow \mathcal{U}_i : m_j = \{N_j, t_j, Z_2, C_j\}$

On receiving a message, $\mathcal{U}_j$ checks the time-stamp freshness and aborts the session if $t_c - t_i \leq \Delta T$, does not hold. Otherwise, $\mathcal{U}_j$ computes $Key_i^{'} = H(d_j Q_i||N_i||t_i)$ and decrypts $C_i$ using key $Key_i^{'}$ to obtain $(TID_i||M_1||Z_1||W_i||N_i||t_i)$. $\mathcal{U}_j$ further computes $W_1^{'} = TID_i \oplus W_i$, $M_1^{'} = H(W_1||Key_i||N_i||Z_1)$ and aborts the session if $M_1^{'} \stackrel{?}{=} M_1$, does not hold. Otherwise, $\mathcal{U}_j$ computes $Z_i^{'} = Z_1 \oplus M_1^{'}$ and selects $N_j \in_R Z_p^*$ and current time-stamp $t_j$ and further computes $Z_j = H(x_j)$, $Z_2 = Z_j \oplus M_1^{'}$, $Key_j = H(Z_i^{'} Z_j||N_j||t_j)$, $W_2 = TID_j \oplus W_j$, $M_2 = H(W_2||Key_j||N_j||Z_2)$. $\mathcal{U}_j$ then computes session key $SK_{xy} = H(TID_i||TID_j||Z_i^{'} Z_j d_j Q_i||key_i^{'}||key_j||M_1^{'}||M_2||N_i||N_j)$ and $C_j = E_{Key_j}(TID_j||M_2||W_j||N_j||t_j)$ and sends back $m_j = \{N_j, t_j, Z_2, C_j\}$ to $\mathcal{U}_j$.

MKA 3: On receiving a message, $\mathcal{U}_i$ checks the time-stamp freshness and aborts the session if $t_c - t_j \leq \Delta T$, does not hold. Otherwise, $\mathcal{U}_i$ computes $Z_j^{'} = Z_2 \oplus M_1$, $Key_j^{'} = H(Z_i Z_j^{'}||N_j||t_j)$ and decrypts $C_j$ using $Key_j^{'}$ to obtain $(TID_j||M_2||W_j||N_j||t_j)$. Further $\mathcal{U}_i$ computes $W_2^{'} = TID_j \oplus W_j$, $M_2^{'} = H(W_2^{'}||Key_j^{'}||N_j||Z_2)$ and aborts the session if $M_2^{'} \stackrel{?}{=} M_2$, does not hold. Otherwise, $\mathcal{U}_i$ considers $\mathcal{U}_j$ is authenticated and computes session key $SK_{xy} = H(TID_i||TID_j||Z_i Z_j^{'} d_i Q_j||key_i||key_j^{'}||M_1||M_2^{'}||N_i||N_j)$.

| *User* $\mathcal{U}_i$ | *User* $\mathcal{U}_j$ |
|---|---|
| Select $N_i \in_R Z_p^*$, Generate $t_i$<br>Compute $W_1 = TID_i \oplus W_i$<br>$Z_i = H(x_i)$<br>$Key_i = H(d_iQ_j\|\|N_i\|\|t_i)$<br>$M_1 = H(W_1\|\|Key_i\|\|N_i\|\|Z_1)$<br>$Z_1 = Z_i \oplus M_1$<br>$C_i = E_{Key_i}(TID_i\|\|M_1\|\|Z_1\|\|W_i\|\|N_i\|\|t_i)$ | |
| $\xrightarrow{m_i=\{N_i,t_i,C_i\}}$ | |
| | Check $t_c - t_x \leq \Delta T$<br>Compute $Key_i^{'} = H(d_jQ_i\|\|N_i\|\|t_i)$<br>$(TID_i\|\|M_1\|\|Z_1\|\|W_i\|\|N_i\|\|t_i) = D_{Keyd_i}(C_i)$<br>$W_1^{'} = TID_i \oplus W_i$<br>$M_1^{'} = H(W_1\|\|Key_i\|\|N_i\|\|Z_1)$<br>Check $M_1^{'} \overset{?}{=} M_1$<br>Compute $Z_i^{'} = Z_1 \oplus M_1^{'}$<br>Select $N_j \in_R Z_p^*$ and Generate $t_j$<br>$Z_j = H(x_j)$<br>$Z_2 = Z_j \oplus M_1^{'}$<br>$Key_j = H(Z_i^{'}Z_j\|\|N_j\|\|t_j)$<br>$W_2 = TID_j \oplus W_j$<br>$M_2 = H(W_2\|\|Key_j\|\|N_j\|\|Z_2)$<br>$SK_{xy} = H(TID_i\|\|TID_j\|\|Z_i^{'}Z_jd_jQ_i\|\|key_i^{'}\|\|key_j\|\|M_1^{'}\|\|M_2\|\|N_i\|\|N_j)$<br>$C_j = E_{Key_j}(TID_j\|\|M_2\|\|W_j\|\|N_j\|\|t_j)$ |
| $\xleftarrow{m_j=\{N_j,t_j,Z_2,C_j\}}$ | |
| Check $t_c - t_j \leq \Delta T$<br>$Z_j^{'} = Z_2 \oplus M_1$<br>Compute $Key_j^{'} = H(Z_iZ_j^{'}\|\|N_j\|\|t_j)$<br>$(TID_j\|\|M_2\|\|W_j\|\|N_j\|\|t_j) = D_{Key_j}(C_j)$<br>$W_2^{'} = TID_j \oplus W_j$<br>$M_2^{'} = H(W_2^{'}\|\|Key_j^{'}\|\|N_j\|\|Z_2)$<br>Check $M_2^{'} \overset{?}{=} M_2$<br>$SK_{xy} = H(TID_i\|\|TID_j\|\|Z_iZ_j^{'}d_iQ_j\|\|key_i\|\|key_j^{'}\|\|M_1\|\|M_2^{'}\|\|N_i\|\|N_j)$ | |

**Figure 3.** Mandal Key Agreement Protocol.

## 5. Weakness of Existing Protocols

In this section, firstly we perform cryptanalysis of Islam-Biswas protocol to show its weaknesses and then we perfom the cryptanalysis of Mandal et al.'s protocol. The following subsections show that both the protocols of Islam-Biswas and Mandal et al. are vulnerable to key compromise impersonation attack, and lack of user anonymity.

### 5.1. Key Compromise Impersonation Attack on Islam-Biswas Protocol

By key compromise impersonation attack, if an active adversary is able to get access to a user's (e.g., $\mathcal{U}_i$) long term private key, then he can masquerade himself as an other user (e.g., $\mathcal{U}_j$) to the victim. In this subsection, we show that Islam-Biswas protocol is vulnerable to key compromise impersonation attack. An active adversary can mount this attack to share a session key with a peer. Let $\mathcal{A}$ be an attacker who wants to impersonate as a legal user $\mathcal{U}_i$ to another legal user $\mathcal{U}_j$. For successful impersonation, the steps performed between $\mathcal{A}$ and $\mathcal{U}_j$ are described as follows:

Step KCI 1: $\mathcal{A}$ computes:

$$T_i^{'} = G \tag{1}$$
$$R_i^{'} = H_1(T_i^{'} \| d_j Q_i) \tag{2}$$

Then $\mathcal{A}$ sends $(ID_i, T_i^{'}, R_i^{'})$ to $\mathcal{U}_j$.

Step KCI 2: Upon receiving the message $\mathcal{U}_j$ selects $y \in_R Z_p^*$, and computes $\mathcal{U}_j$

$$T_j = yQ_j \tag{3}$$
$$R_j = H_1(T_j \| d_j Q_i) \tag{4}$$

Further $\mathcal{U}_j$ sends $(ID_j, T_j, R_j)$ to $\mathcal{U}_i$.

Step KCI 3: $\mathcal{A}$ intercepts the message and computes

$$R_j^* = H_1(T_j \| d_j Q_i) \tag{5}$$

and verifies

$$R_j^* \stackrel{?}{=} R_j \tag{6}$$

Then $\mathcal{A}$ computes:

$$K = K_i^{'} = (d_j)T_j = yd_jG \tag{7}$$
$$SK = H_2(ID_i \| ID_j \| T_i^{'} \| T_j \| R_i^{'} \| R_j \| K) \tag{8}$$

Similarly $\mathcal{U}_j$ computes:

$$R_i^* = H_1(T_i^{'} \| d_j Q_i) \tag{9}$$

and verifies

$$R_i^* \stackrel{?}{=} R_i^{'} \tag{10}$$

If Equation (10) does not hold, $\mathcal{U}_j$ aborts the session, otherwise $\mathcal{U}_j$ believes the party on other side is $\mathcal{U}_i$ and computes:

$$K = K_j = (yd_j)T_i^{'} = yd_jG \tag{11}$$
$$SK = H_2(ID_i \| ID_j \| T_i^{'} \| T_j \| R_i^{'} \| R_j \| K) \tag{12}$$

**Proposition 1.** *In Islam-Biswas protocol, upon execution of key compromise impersonation attack, user $\mathcal{U}_j$ accepts adversary $\mathcal{A}$ as another user $\mathcal{U}_i$ and $\mathcal{A}$ shares the session key with $\mathcal{U}_j$ on behalf of $\mathcal{U}_i$.*

**Proof.** $\mathcal{A}$ initiates the key compromise impersonation attack by computing $T_i^{\prime} = G$ and $R_i^{'} = H_1(T_i^{'} \| d_j Q_i)$, then $\mathcal{A}$ sends $ID_i, T_i^{'}, R_i^{'}$ to $\mathcal{U}_j$, which believes the other party is legal $\mathcal{U}_i$ if Equation (10) holds. $\mathcal{U}_j$ computes $R_i^*$ in Equation (9), which is equal to $R_i^{'}$ computed by $\mathcal{A}$ in Equation (2). Hence $\mathcal{A}$ is believed to be $\mathcal{U}_i$ by $\mathcal{U}_j$. The session key computed by both $\mathcal{A}$ and $\mathcal{U}_j$ is also same, as $\mathcal{A}$ computed session key $SK$ in Equation (8) which is exactly the same as computed by $\mathcal{U}_j$ in Equation (12). Hence, $\mathcal{A}$ has successfully launched KCIA on Islam-Biswas's protocol. □

### 5.2. Key Compromise Impersonation Attack on Mandal et al.'s Protocol

This subsection shows that the protocol of Mandal et al. is also vulnerable to Key Compromise Impersonation Attack (KCIA). Let $\mathcal{A}$ be an attacker who wants to impersonate as a legal user $\mathcal{U}_i$ to another legal user $\mathcal{U}_j$. For successful impersonation, the steps performed between $\mathcal{A}$ and $\mathcal{U}_j$ are simulated as follows:

KCM 1: $\mathcal{A}$ randomly selects $N_a, TID_a, W_a, Z_a \in_R Z_p^*$, generates $t_a$ and computes:

$$W_1 = TID_a \oplus W_a \tag{13}$$
$$Key_a = H(d_j Q_i || N_a || t_a) \tag{14}$$
$$M_1 = H(W_1 || Key_a || N_a || Z_1) \tag{15}$$
$$Z_1 = Z_a \oplus M_1 \tag{16}$$
$$C_a = E_{Key_a}(TID_a || M_1 || Z_1 || W_a || N_a || t_a) \tag{17}$$

$\mathcal{A}$ sends $m_a = \{N_a, t_a, C_a\}$ to $\mathcal{U}_j$.

KCM 2: On receiving a message, $\mathcal{U}_j$ checks the time-stamp freshness and aborts the session if $t_c - t_a \leq \Delta T$, does not hold. $\mathcal{U}_j$ then computes:

$$Key_a^{'} = H(d_j Q_i || N_a || t_a) \tag{18}$$
$$(TID_a || M_1 || Z_1 || W_a || N_a || t_a) = D_{Key_a^{'}}(C_a) \tag{19}$$
$$W_1^{'} = TID_a \oplus W_a \tag{20}$$
$$M_1^{'} = H(W_1 || Key_a || N_a || Z_1) \tag{21}$$

$\mathcal{U}_j$ then checks :

$$M_1^{'} \overset{?}{=} M_1 \tag{22}$$

Upon success, $\mathcal{U}_j$ selects $N_j \in_R Z_p^*$ and $t_j$ and computes:

$$Z_a^{'} = Z_1 \oplus M_1^{'} \tag{23}$$
$$Z_j = H(x_j) \tag{24}$$
$$Z_2 = Z_j \oplus M_1^{'} \tag{25}$$
$$Key_j = H(Z_a^{'} Z_j || N_j || t_j) \tag{26}$$
$$W_2 = TID_j \oplus W_j \tag{27}$$
$$M_2 = H(W_2 || Key_j || N_j || Z_2) \tag{28}$$
$$SK_{xy} = H(TID_a || TID_j || Z_a^{'} Z_j d_j Q_a || key_a^{'} || key_j || M_1^{'} || M_2 || N_a || N_j) \tag{29}$$
$$C_j = E_{Key_j}(TID_j || M_2 || W_j || N_j || t_j) \tag{30}$$

$\mathcal{U}_j$ sends back $m_j = \{N_j, t_j, Z_2, C_j\}$ to $\mathcal{U}_i$.

KCM 3: $\mathcal{A}$ intercepts the messages and computes:

$$Z_j^{'} = Z_2 \oplus M_1 \tag{31}$$
$$(TID_j || M_2 || W_j || N_j || t_j) = D_{Key_j^{'}}(C_j) \tag{32}$$
$$W_2^{'} = TID_j \oplus W_j \tag{33}$$
$$M_2^{'} = H(W_2^{'} || Key_j^{'} || N_j || Z_2) \tag{34}$$

$\mathcal{A}$ then computes session key as:

$$SK_{xy} = H(TID_a||TID_j||Z_a Z_j^{'} d_j Q_i||key_a||key_j^{'}||M_1||M_2^{'}||N_a||N_j) \tag{35}$$

**Proposition 2.** *In Mandal et al.'s protocol, upon execution of key compromise impersonation attack, user $\mathcal{U}_j$ accepts adversary $\mathcal{A}$ as another user $\mathcal{U}_i$ and $\mathcal{A}$ shares the session key with $\mathcal{U}_j$ on behalf of $\mathcal{U}_i$.*

**Proof.** $\mathcal{A}$ initiates the key compromise impersonation attack by computing $W_1, Key_a, M_1, Z_1$ and $C_a$ then $\mathcal{A}$ sends $\{N_a, t_a, C_a\}$ tuple to $\mathcal{U}_j$, which believes the other party is legal $\mathcal{U}_i$ if Equation (22) holds. The security of the protocol relies on the computation of $Key_a$, if $Key_a$ is computed same on both sides, then decryption of $C_a$ on $\mathcal{U}_j$ will be same as computed by $\mathcal{A}$. Therefore, $M_1$ computed in Equation (15) by $\mathcal{A}$ and in Equation (21) by $\mathcal{U}_j$ will also be same. Hence Equation (22) will hold true. $\mathcal{U}_j$ computes $Key_a^{'}$ in Equation (18), which is equal to $Key_a$ computed by $\mathcal{A}$ in Equation (14). Therefore, Equation (22) holds. Hence, $\mathcal{A}$ is believed to be $\mathcal{U}_i$ by $\mathcal{U}_j$. The session key computed by both $\mathcal{A}$ and $\mathcal{U}_j$ is also same, as $\mathcal{A}$ computed session key $SK$ in Equation (35) which is exactly the same as computed by $\mathcal{U}_j$ in Equation (29). Hence, $\mathcal{A}$ has successfully launched KCIA on Mandal et al.'s protocol. □

### *5.3. Lacking User Anonymity*

Both the protocols of Islam-Biswas and Mandal et al., lack user anonymity and privacy. The former did not claim to provide anonymity, whereas, latter claimed to provide it. However, after a careful analysis, it is revealed that their protocol lacks anonymity. Our analysis is simulated as follows:
After computing $W_1, Key_i, M_1, Z_1$ and $C_a$, the $\mathcal{U}_i$ sends $\{N_i, t_i, C_i\}$ tuple to $\mathcal{U}_j$. $\mathcal{U}_j$ after verification of freshness computes:

$$Key_i^{'} = H(d_j Q_i||N_i||t_i) \tag{36}$$

The computation of Equation (36) requires the public key $Q_i$ of the user $\mathcal{U}_i$. However, the received message $\{N_i, t_i, C_i\}$ does not contain any information to identify the requesting user. Therefore, the protocol will not work. The authors in this paper consider it a typographical mistake and the complete request message may be $\{N_i, t_i, C_i, ID_i\}$, because in other case, the protocol is incorrect and cannot complete the authentication process. As per the valid assumption made by authors, the protocol of Mandal et al. does not provide user anonymity.

## 6. Proposed Protocol

This section briefly explains the proposed protocol designed specifically to resist key compromise impersonation attack (KCIA). The proposed protocol is based on ECC and self certified keys and resist all known attacks. The proposed protocol involves two entities: (1) The server is responsible for registration of the devices and assigns certificates to each of the device, the server is assumed to be trusted, (2) the communicating devices after getting certificate from server can establish secure connection with each other without intervention of server or any other party. Following subsections explains the proposed methodology:

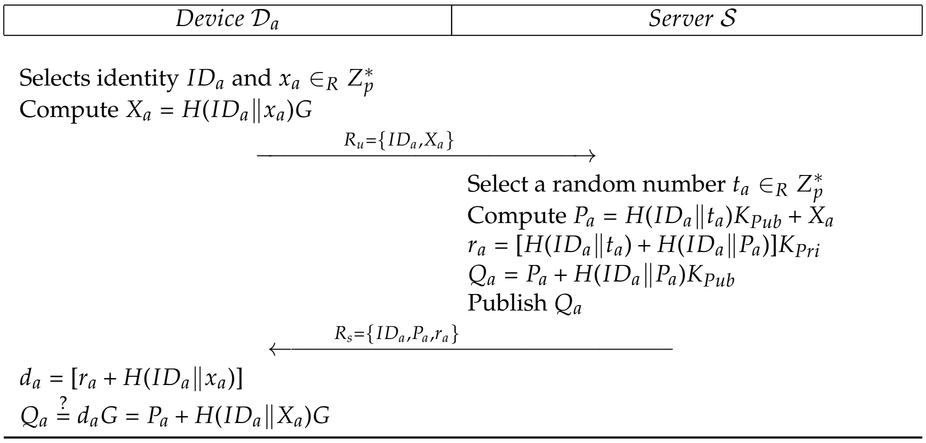
### *6.1. Setup Phase*

In system setup phase, the server ($\mathcal{S}$) initializes the system parameter $\Omega$. Initially $\mathcal{S}$ chooses a security parameter $k \in Z^+$ along with an elliptic curve $E/F_p$, then $\mathcal{S}$ selects a base point $G$ over $E/F_p$. Further $\mathcal{S}$ selects $K_{Pri}$ as his private key and computes $K_{Pub} = K_{Pri}G$ and chooses a one way hash functions $H : \{0,1\}^* \rightarrow \{0,1\}^k$. Finally $\mathcal{S}$ publishes all public parameters $\Omega = \{E/F_p, H, G, K_{Pub}\}$ and keeps $K_{Pri}$ secret.

### 6.2. Registration Phase

This phase is very similar to the corresponding phase of Islam et al.'s protocol and is initiated by a device $\mathcal{D}_a$, when $\mathcal{D}_a$ wants to register with $\mathcal{S}$. $\mathcal{D}_a$ selects his identity $ID_a$ and a random number $x_a \in_R Z_p^*$, then $\mathcal{D}_a$ computes $X_a = H(ID_a\|x_a)G$ and sends $ID_a, X_a$ to $\mathcal{S}$ via some secure channel, which selects $t_a \in_R Z_p^*$ upon receiving a message from $\mathcal{D}_a$. $\mathcal{S}$ then computes $P_a = H(ID_a\|t_a)K_{Pub} + X_a$, $r_a = [H(ID_a\|t_a) + H(ID_a\|P_a)]K_{Pri}$ and $Q_a = P_a + H(ID_a\|P_a)K_{Pub}$. $\mathcal{S}$ sends $(ID_a, P_a, r_a)$ to $\mathcal{D}_a$ via some secure channel and publishes $Q_a$. Upon receiving, $\mathcal{D}_a$ computes his private key $d_a = [r_a + H(ID_a\|x_a)]$, the public key of $\mathcal{D}_a$ is $d_aG = Q_a$. The registration phase is also illustrated in Figure 4. The private key of $\mathcal{D}_a$ can be verified as follows:

$$\begin{aligned} d_aG &= [r_a + H(ID_a\|x_a)]G \\ &= [[H(ID_a\|t_a) + H(ID_a\|P_a)]K_{Pri} + H(ID_a||X_a)]G \\ &= [H(ID_a\|t_a)K_{Pub} + H(ID_a\|P_a)K_{Pub} + H(ID_a||X_a)G \\ &= P_a + H(ID_a||P_a)K_{Pub} \\ &= Q_a \end{aligned} \tag{37}$$



Device $\mathcal{D}_a$ | Server $\mathcal{S}$

Selects identity $ID_a$ and $x_a \in_R Z_p^*$
Compute $X_a = H(ID_a\|x_a)G$

$$\xrightarrow{R_u=\{ID_a,X_a\}}$$

Select a random number $t_a \in_R Z_p^*$
Compute $P_a = H(ID_a\|t_a)K_{Pub} + X_a$
$r_a = [H(ID_a\|t_a) + H(ID_a\|P_a)]K_{Pri}$
$Q_a = P_a + H(ID_a\|P_a)K_{Pub}$
Publish $Q_a$

$$\xleftarrow{R_s=\{ID_a,P_a,r_a\}}$$

$d_a = [r_a + H(ID_a\|x_a)]$
$Q_a \overset{?}{=} d_aG = P_a + H(ID_a\|X_a)G$

**Figure 4.** Proposed registration.

### 6.3. Authenticated Key Agreement Phase

In proposed scheme, a device say $\mathcal{D}_i$ initiates the process to exchange authenticated key with peer say $\mathcal{D}_j$. Following steps as shown in Figure 5 are performed among $\mathcal{D}_i$ and $\mathcal{D}_j$:

PKA 1: $\mathcal{D}_i \rightarrow \mathcal{D}_j : m_{i1} = \{AID_i, \tau_i, \gamma_i, t_i\}$
$\mathcal{D}_i$ selects $x \in_R Z_p^*$, generates $t_i$ and computes $\tau_i = xG$, $\alpha_i = xQ_j$, $AID_i = \alpha_i \oplus ID_i$ and $\gamma_i = H(\alpha_i||\tau_i||ID_i||ID_j||t_i)$. Then $\mathcal{D}_i$ sends $m_{i1} = \{AID_i, \tau_i, \gamma_i t_i\}$ to $\mathcal{D}_j$.

PKA 2: $\mathcal{D}_j \rightarrow \mathcal{D}_i : m_j = \{AID_j, \tau_j, R_j, t_j\}$
On receiving request message, $\mathcal{D}_j$ aborts the session if $t_c - t_i \leq \Delta T$. Otherwise, $\mathcal{D}_j$ computes $\alpha_i = d_j\tau_i$, $ID_i = AID_i \oplus \alpha_i$ and aborts the session if $\gamma_i \neq H(\alpha_i||\tau_i||ID_i||ID_j||t_i)$. Otherwise, $\mathcal{D}_j$ selects $y \in_R Z_p^*$, generates $t_j$ and computes $\tau_j = yG$, $K = K_j = yQ_i + d_j\tau_i$, $AID_j = \alpha_i \oplus ID_j$, $R_j = H(K||\alpha_i||\tau_i||\tau_j||ID_i||ID_j||t_j)$. The $\mathcal{D}_j$ sends $m_j = \{AID_j, \tau_j, R_j, t_j\}$ to $\mathcal{D}_i$.

PKA 3: $\mathcal{D}_i \rightarrow \mathcal{D}_j : m_{i2} = \{R_i\}$
After receiving the reply, $\mathcal{D}_i$ aborts the session if $t_c - t_j \leq \Delta T$. Otherwise, $\mathcal{D}_i$ computes $ID_j = AID_j \oplus \alpha_i$, $K = K_i = xQ_j + d_i\tau_j$ and checks $R_j \overset{?}{=} H(K||\alpha_i||\tau_i||\tau_j||ID_i||ID_j||t_j)$, continues to compute $SK = H(ID_i\|ID_j\|\tau_i\|\tau_j\|K)$ and $R_i = H(SK||ID_i||ID_j||K)$, if the equality holds. The $\mathcal{D}_i$ sends $m_{i2} = \{R_i\}$ to $\mathcal{D}_j$.

PKA 4: $\mathcal{D}_j$ on receiving $m_{i2}$ computes $SK = H(ID_i\|ID_j\|\tau_i\|\tau_j\|K)$ and verifies $R_i \overset{?}{=} H(SK||ID_i||ID_j||K)$. $\mathcal{D}_j$ terminates the session on failure and keeps $SK$ as session key upon success.

| *Device* $\mathcal{D}_i$ | *Device* $\mathcal{D}_j$ |
|---|---|
| Select $x \in_R Z_p^*$ and $t_i$<br>Compute $\tau_i = xG$<br>$\alpha_i = xQ_j$<br>$AID_i = \alpha_i \oplus ID_i$<br>$\gamma_i = H(\alpha_i\|\|\tau_i\|\|ID_i\|\|ID_j\|\|t_i)$ | |
| $\xrightarrow{m_{i1}=\{AID_i,\tau_i,\gamma_i,t_i\}}$ | |
| | Check $t_c - t_i \leq \Delta T$<br>Compute $\alpha_i = d_j\tau_i$<br>$ID_i = AID_i \oplus \alpha_i$<br>$\gamma_i \overset{?}{=} H(\alpha_i\|\|\tau_i\|\|ID_i\|\|ID_j\|\|t_i)$<br>Select $y \in_R Z_p^*$ and $t_j$<br>Compute: $\tau_j = yG$<br>$K = K_j = yQ_i + d_j\tau_i$<br>$AID_j = \alpha_i \oplus ID_j$<br>$R_j = H(K\|\|\alpha_i\|\|\tau_i\|\|\tau_j\|\|ID_i\|\|ID_j\|\|t_j)$ |
| $\xleftarrow{m_j=\{AID_j,\tau_j,R_j,t_j\}}$ | |
| Check $t_c - t_j \leq \Delta T$<br>$ID_j = AID_j \oplus \alpha_i$<br>$K = K_i = xQ_j + d_i\tau_j$<br>$R_j \overset{?}{=} H(K\|\|\alpha_i\|\|\tau_i\|\|\tau_j\|\|ID_i\|\|ID_j\|\|t_j)$<br>$SK = H(ID_i\|ID_j\|\tau_i\|\tau_j\|K)$<br>$R_i = H(SK\|\|ID_i\|\|ID_j\|\|K)$ | |
| $\xrightarrow{m_{i2}=\{R_i\}}$ | |
| | $SK = H(ID_i\|ID_j\|\tau_i\|\tau_j\|K)$<br>$R_i \overset{?}{=} H(SK\|\|ID_i\|\|ID_j\|\|K)$ |

**Figure 5.** Proposed key agreement.

## 7. Security Analysis

In this section the security of proposed protocol under the attack model of automated tool Scyther is performed, backed by the security requirements discussion. This section also provides a security features comparison of the proposed and existing protocols [13,31,32,36,37] in Table 2. Referring to Table 2, only the proposed schemes provide all security features, whereas all other protocols lacks device anonymity. The protocols [13,36,37] are insecure key replication (KRA/KOA) attack, the protocols [13,31,32] are insecure against Key compromise impersonation attack (KCIA). Protocol proposed by Islam-Biswas [31] is also insecure against replay attack. Following subsections provides detailed security analysis and security features provided by the proposed protocol:

**Table 2.** Security Comparison table.

| **Features**→ **Protocols**↓ | $\mathcal{RF}_1$ | $\mathcal{RF}_2$ | $\mathcal{RF}_3$ | $\mathcal{RF}_4$ | $\mathcal{RF}_5$ | $\mathcal{RF}_6$ | $\mathcal{RF}_7$ | $\mathcal{RF}_8$ | $\mathcal{RF}_9$ | $\mathcal{RF}_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Ours | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [13] | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| [36] | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| [37] | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| [31] | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [32] | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |

Note: $\mathcal{RF}_1$: Key Compromise Impersonation Attack; $\mathcal{RF}_2$: device Anonymity; $\mathcal{RF}_3$: Man in Middle Attack; $\mathcal{RF}_4$: Known Key attack; $\mathcal{RF}_5$: Unknown Key Share Attack; $\mathcal{RF}_6$: Perfect Forward Secrecy; $\mathcal{RF}_7$: Known Session Specific Information Attack; $\mathcal{RF}_8$: Key Offset/Replicate Attack; $\mathcal{RF}_9$: No Key Control;$\mathcal{RF}_{10}$: Replay Attack ✓: indicates that the scheme provides or is secure against that feature; ✗: indicates that the scheme does not provide or is insecure against that feature.

### *7.1. Formal Security*

To analyze formally, the security and privacy of the proposed protocol, following oracles are defined:

- $Reveal_h$**:** Execution of this oracle unconditionally yields $S_a$ out of $H(S_a)$.
- $Reveal_{dlp}$**:** Given the pair $\{V = a.W, W\}$, execution of this oracle unconditionally provides $a$.

**Theorem 1.** *The proposed device to device security protocol is secure for $\mathcal{A}$ - an attacker, to expose $ID_a$ of device $\mathcal{D}_a$, the parameter $K = yQ_i + d_j.\tau_i$, the session key $SK = H(ID_i||ID_j||\tau_i||\tau_j||K)$ shared between $\mathcal{D}_a$ and $\mathcal{S}$ under the hardness of ECDLP and hash function is considered as a random oracle.*

**Proof.** $\mathcal{A}$ is considered as an attacker with abilities to compute $ID_a$ of device $\mathcal{D}_a$, secretly computed parameter $K = yQ_i + d_j\tau_j$ and $SK = H(ID_i||ID_j||\tau_i||\tau_j||K)$ between $\mathcal{D}_a$ and $\mathcal{D}_b$. $\mathcal{A}$ simulates the oracles oracles $Reveal_h$ and $Reveal_{dlp}$ for the execution of the algorithmic experiment (Algorithm 1) $EXPE1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$ against the two party device-to-device authenticated key agreement ($2DTDAKA$) protocol. The success probability of $EXPE1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$ can be solicited as $Suc_{ex1} = |P[EXPE1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA} = 1] - 1|$, where the advantage of $\mathcal{A}$ is $Advt1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}(t_f, q_{revH}, q_{revD}) = max_{\mathcal{A}}(Succe_{ex1})$. The maximum allowed queries $\mathcal{A}$ can make are $q_{revH}$ and $q_{revD}$, for each of the oracles $Reveal_h$ and $Reveal_{dlp}$. Referring the simulation of $EXPE1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$, $\mathcal{A}$ can compute $ID_a$, $K$ and $SK$ if $\mathcal{A}$ has the abilities to (i) break one-way property of hash and (ii) Compute the hard $ECDLP$. As per Definition 1, inverting hash is hard problem; likewise, by Definition 2 solving ECDLP is also computationally infeasible for large parameter sizes ($geq$160 bits). Hence, proposed $2DTDAKA$ is unbreakable against disclosure of secretly computed parameter $K$, session key $SK$ and device identity $ID_a$. □

**Algorithm 1** $EXPE1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$

```
Eavesdrop the Request $m_{i1} = \{AID_i, \tau_i, \gamma_i, t_i\}$, Where $AID_i = \alpha_i \oplus ID_i$, $\tau_i = x.G$ and $\gamma_i = H(\alpha_i||\tau_i||ID_i||ID_j||t_i)$
Call $Reveal_{dlp}$ oracle on $\tau_i$ and $G$ and get $x^{'} \leftarrow Reveal_{dlp}(\tau_i, G)$
Compute $\alpha_i^{'} = x^{'}.Q_j$ and $ID_i^{'} = AID_i \oplus \alpha_i^{'}$
Call $Reveal_h$ on $\gamma_i$ and get $(\alpha_i^{''}||\tau_i^{'}||ID_i^{''}||ID_j^{'}||t_i^{'}) \leftarrow Reveal_h(\gamma_i)$
if $(ID_i^{''} = ID_i^{'}$ and $t_i == t_i^{'}$ and $\alpha_i^{'} == \alpha_i^{''}$ ) then
    Accept $ID_i^{'}$ along-with session parameters $x^{'}$ and $\tau_i^{'}$ and
    Eavesdrop Challenge $m_j = \{AID_j, \tau_j, R_j, t_j\}$, where $AID_j = \alpha_i \oplus ID_j$, $\tau_j = y.G$ and $R_j = H(K||\alpha_i||\tau_i||\tau_j||ID_i||ID_j||t_j)$
    Compute $ID_j^{'} = AID_j \oplus \alpha_i^{'}$
    Call $Reveal_h$ on $R_j$ and get $(K^{'}||\alpha_i^{'''}||\tau_i^{''}||\tau_j^{'}||ID_i^{'''}||ID_j^{''}||t_j^{'}) \leftarrow Reveal_h(R_j)$
    if $(ID_j^{'} = ID_j^{'''}$ and $t_j == t_i^{'})$ then
        Accept $K^{'}$ and compute $SK^{'} = H(ID_i^{'}||ID_j^{'}||\tau_i^{'}||\tau_j^{'}||k^{'})$
        Eavesdrop response $m_{i2} = \{R_i\}$
        Call $Reveal_h$ on $R_i$ and get $(SK^{''}||ID_i^{'''}||ID_j^{'''}||K^{''}) \leftarrow Reveal_h(R_i)$
        if $(SK^{'} == SK^{''})$ then
            Accept $SK^{'}$
        else
            return Fail
        end if
    else
        return Fail
    end if
else
    return Fail
end if
```

**Theorem 2.** *The proposed device to device security protocol is secure for $\mathcal{A}$ - an attacker, with access to private key of a registered device $\mathcal{D}_j$, to share a session key SK with $\mathcal{D}_j$ on behalf of another registered device $\mathcal{D}_i$.*

**Proof.** $\mathcal{A}$ having access to private key $d_j$ of registered device $\mathcal{D}_j$ is considered as competent enough to compute, secretly computed parameter $K = yQ_i + d_j\tau_i$ and $SK = H(ID_i||ID_j||\tau_i||\tau_j||K)$ between $\mathcal{A}$ (on behalf of $\mathcal{D}_i$ ) and $\mathcal{D}_b$. $\mathcal{A}$ simulates the oracles $Reveal_h$ and $Reveal_{dlp}$ for the execution of the algorithmic-experiment (Algorithm 2) $EXPE2^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$ against the 2 party device-to-device authenticated key agreement ($2DTDAKA$) protocol. The success probability of $EXPE2^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$ can be solicited as $Suc_{ex2} = |P[EXPE1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA} = 1] - 1|$, where the advantage of $\mathcal{A}$ is $Advt1^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}(t_f, q_{revH}, q_{revD}) = max_{\mathcal{A}}(Succe_{ex2})$. The maximum allowed queries $\mathcal{A}$ can make are $q_{revH}$ and $q_{revD}$, for each of the oracles $Reveal_h$ and $Reveal_{dlp}$. Referring the simulation of $EXPE2^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$, $\mathcal{A}$ can compute $K$ and $SK$ if $\mathcal{A}$ has the abilities to (i) break one-way property of hash and (ii) Compute the hard $ECDLP$. As per Definition 1, inverting hash is hard problem; likewise, by Definition 2 solving ECDLP is also computationally infeasible for large parameter sizes ($\geq$ 160 bits). Therefor, proposed $2DTDAKA$ is unbreakable against disclosure of secretly computed parameter $K$ and session key $SK$, given private key of victim and can resist KCIA. □

**Algorithm 2** $EXPE2^{ECDLP,HASH}_{\mathcal{A},2DTDAKA}$

```
Compute $\tau_i = x.G$, $\alpha_i = x.Q_j$, $AID_i = \alpha_i \oplus ID_i$ and $\gamma_i = H(\alpha_i||\tau_i||ID_i||ID_j||t_i)$
Send $m_{i1} = \{AID_i, \tau_i, \gamma_i, t_i\}$ to $\mathcal{D}_j$
Eavesdrop Challenge $m_j = \{AID_j, \tau_j, R_j, t_j\}$, where $AID_j = \alpha_i \oplus ID_j$, $\tau_j = y.G$ and $R_j = H(K||\alpha_i||\tau_i||\tau_j||ID_i||ID_j||t_j)$
Compute $ID_j = AID_j \oplus \alpha_i$
Call $Reveal_{dlp}$ oracle on $\tau_j$ and get $y' \leftarrow Reveal_{dlp}(\tau_j)$
Compute $K = x.Q_j + d_i.\tau_j = (x.Q_j + y'.Q_i)$
Call $Reveal_h$ on $R_j$ and get $(K'||\alpha_i||\tau_i||\tau_j||ID_i||ID_j||t_j) \leftarrow Reveal_h(R_j)$
if ($K == K'$) then
    Compute $SK = H(ID_i\|ID_j\|\tau_i\|\tau_j\|K)$
    Compute $R_i = H(SK||ID_i||ID_j||K)$
else Send $m_{i2} = \{R_i\}$ to $\mathcal{D}_j$
    return Fail
end if
```

### 7.2. BAN Logic Based Security Analysis

In this section the formal security analysis of the proposed scheme has been done by using Burrows-Abadi-Needham (BAN) logic. We analyze the likelihood of mutual authentication among participants, along with the resistance from session key disclosure by using the BAN logic.

Various rules and principals were presented by Burrows, Abadi and Needham in 1989. If any one of these rules is being violated then the protocol/scheme is considered incorrect. Here are some rules and their descriptions:

**Rule 1: Message Meaning**

$$\frac{P|\equiv P\overset{K}{\longleftrightarrow}Q.P\triangleleft<X>_K}{P|\equiv Q|\sim X}$$

This rule depicts that P believe, and Q one time said that if P believes than secret key K shared with Q and P see that X is encrypted by using key K.

**Rule 2: Nonce Verification**

$$\frac{P|\equiv\#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$$

this rule says that P is believing that Q also believes X, if P is still believing that X is fresh and Q said that X.

**Rule 3: Jurisdiction**

$$\frac{P|\equiv Q \Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$$

We can say that $P$ is believing on $Q$ and also $X$ is valid, if and only if when $P$ is believing that $Q$ has the jurisdiction over $X$.

**Rule 4: Acceptance Conjunction**

$$\frac{P|\equiv\ X, P|\equiv Y}{P|\equiv (X,Y)}$$

If a $P$ believes on $X$ and $X$ believes on $Y$, as a result we can say that $P$ principal believes on both $(X, Y)$ too.

**Rule 5: Freshness Conjunction**

$$\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$$

In this rule we can said that $P$ believing that both $X$ and $Y$ are fresh if and only if when $P$ believe $X$ is still fresh.

**Rule 6: Session Key**

$$\frac{P|\equiv \#(X), P|\equiv Q \equiv X}{P|\equiv\ P \overset{K}{\longleftrightarrow} Q}$$

In the session key rule if a $P$ principal believes on the freshness of session key then also $P$ and then $Q$ also on $X$ believes which is the most important part of the session key. And then $P$ principal also believes that user shares a session key "$K$" with $Q$.

We employ the following notations in verifying the the security properties.

- $\gamma|\equiv \sigma$ : $\gamma$ believes $\sigma$
- $\gamma \lhd \sigma$ : $\gamma$ sees $\sigma$
- $\gamma|\sim \sigma$ : $\gamma$ once said $\sigma$, some time ago.
- $\gamma| \implies \sigma$ : $\gamma$ has got jurisdiction over $\sigma$
- $\#(\sigma)$: The message $\sigma$ is to be taken as fresh.
- $(\sigma)\sigma^{'}$ : The formulae $\sigma$ is hashed in combination with formulae $\sigma^{'}$.
- $(\sigma, \sigma^{'})$ : $\sigma$ or $\sigma^{'}$ being the part of message $(\sigma, \sigma^{'})$.
- $(\sigma, \sigma^{'})_k \to \gamma$ : $\sigma$ or $\sigma^{'}$ is encrypted with symmetric or asymmetric key $K$ of $\gamma$.
- $\gamma \overset{K}{\longleftrightarrow} \gamma^{'}$ : $\gamma$ and $\gamma^{'}$ can securely contact using the shared key $K$.

The following are the assumptions for the BAN logic analysis.

- $A1$: $D_i|\equiv \#(t_i)$
- $A2$: $D_j|\equiv \#(t_j)$
- $A3$: $D_i|\equiv (D_i \overset{SK}{\longleftrightarrow} D_j)$
- $A4$: $D_j|\equiv (D_i \overset{SK}{\longleftrightarrow} D_j)$
- $A5$ : $D_i| \implies K_i$
- $A6$: $D_j| \implies K_i$

The following goals serve as the target for proving this analysis.

- Goal 1: $D_j|\equiv (D_i \overset{SK}{\longleftrightarrow} D_j)$
- Goal 2: $D_j|\equiv D_i|\equiv (D_i \overset{SK}{\longleftrightarrow} D_j)$
- Goal 3: $D_i|\equiv (D_i \overset{SK}{\longleftrightarrow} D_j)$
- Goal 4: $D_i|\equiv D_j|\equiv (D_i \overset{SK}{\longleftrightarrow} D_j)$

The protocol's generic form is illustrated as under.

- M1: $D_i \rightarrow D_j : AID_i,\ \tau_i,\ y_i,\ t_i$
- M2: $D_j \rightarrow D_i : AID_j,\ \tau_j,\ R_j,\ t_j$:
- M3: $D_i \rightarrow D_j : R_i :$

The idealized form of the protocol is designed as follows.

- $M1$: $D_i \rightarrow D_j : \{(IDi)_{a_i},\ x.G,\ (ID_j, t_i)(a_i,\ ID_i),\ t_i\}$
- $M2$: $D_j \rightarrow D_i : \{(ID_j)_{a_i},\ y.G, (a_i,\ \tau_i,\ \tau_j,\ ID_i,\ ID_j,\ t_j)_k,\ t_j\}$
- $M3$: $D_i \rightarrow D_j : \{(ID_i,\ ID_j,\ K)_{SK}\}$

Considering the first and third message of the idealized form:

- $M1$: $D_i \rightarrow D_j : \{(ID_i)_{a_i},\ x.G,\ (ID_j, t_i)_{(ai, ID_i)},\ t_i\}$
- $M3$: $D_i \rightarrow D_j : \{(ID_i,\ ID_j,\ K)_{SK}\}$

By Applying seeing rule, we get,

- $S1$: $D_j \lhd \{(ID_i)_{a_i},\ x.G,\ (ID_j, t_i)_{(ai, IDi)},\ t_i\}$
- $S2$: $D_j \lhd \{(ID_i, ID_j, K)_{SK}\}$

According to S1, S2, A3 and message meaning rule,

- $S3$: $D_j| \equiv \{(IDi)_{a_i},\ x.G,\ (ID_j,\ t_i)_{(a_i,\ ID_i)},\ ti\}$
- $S4$: $D_j| \equiv \{(ID_i,\ ID_j,\ K)_{SK}\}$

According to A1, S3, S4 freshness conjucatenation, and nonce verification rules, we get

- $S5$: $D_j| \equiv D_i| \equiv \{(ID_i)_{ai},\ x.G,\ (IDj,\ t_i)_{(a_i,\ ID_i)}, t_i\}$
- $S6$: $D_j| \equiv D_i| \equiv \{(ID_i,\ ID_j, K)_{SK}\}$

According to A6, S5, S6 and Jurisdiction rule

- $S6$: $D_j| \equiv \{(IDi)_{ai},\ x.G,\ (ID_j, t_i)_{(a_i,\ ID_i)},\ t_i\}$
- $S7$: $D_j| \equiv \{(ID_i,\ ID_j,\ K)_{SK}\}$

According to A3, S6, S7, and session key rule, we get

- $S8$: $D_j| \equiv D_i| \equiv D_i \xleftrightarrow{SK} D_j$ (**Goal 2**)

According to A6, S8, and Jurisdiction rule

- $S9$: $D_j| \equiv D_i \xleftrightarrow{SK} D_j$ (**Goal 1**)

Considering the second idealized form as:

- $M2$: $D_j \rightarrow D_i : \{(ID_j)_{a_i},\ y.G,\ (a_i,\ \tau_i,\ \tau_j,\ ID_i,\ ID_j,\ t_j)_K,\ t_j\}$

By applying seeing rule, we get

- $S10$: $D_i \lhd : \{(IDj)_{a_i},\ y.G,\ (a_i,\ \tau_i,\ \tau_j,\ ID_i,\ ID_j,\ t_j)_K,\ t_j\}$

According to S10, A4 and message meaning rule,

- $S11$: $D_i| \equiv D_j \sim \{(ID_j)_{a_i},\ y.G,\ (a_i,\ \tau_i,\ \tau_j,\ ID_i,\ ID_j,\ t_j)_K,\ t_j\}$

According to A2, S11, freshness conjucatenation, and nonce verification rules we get,

- $S12$: $D_i| \equiv D_j| \equiv \{(ID_j)_{ai},\ y.G,\ (a_i,\ \tau_i,\ \tau_j,\ ID_i,\ ID_j,\ t_j)_K,\ tj\}$

According to A5, S12, and Jurisdiction rule

- $S13$: $D_i| \equiv \{(ID_j)_{ai},\ y.G,\ (a_i,\ \tau_i,\ \tau_j,\ ID_i,\ ID_j,\ t_j)_K, t_j\}$

According to A4, S13, and session key rule, we get

- $S14$: $D_i| \equiv D_j| \equiv D_i \overset{SK}{\longleftrightarrow} D_j$ (**Goal 4**)

According to A5, S14, and Jurisdiction rule

- $S15$: $D_i| \equiv D_j \overset{SK}{\longleftrightarrow} D_i$ (**Goal 3**)

The above BAN logic analysis formally proves that the proposed protocol achieves mutual authentication and the session key SK is mutually established between $D_i$ and $D_j$.

### *7.3. Security Features Analysis*

Following subsections provide a discussion on attack resilience of the proposed protocol:

#### 7.3.1. Key Compromise Impersonation Attack

By KCIA, if an adversary $\mathcal{A}$ gets long private key of a device say $\mathcal{D}_a$ can impersonate himself as anyother device say $\mathcal{D}_b$ of the system to the victim $\mathcal{D}_a$. In proposed protocol if $\mathcal{A}$ gets the long term private key $d_a = r_a + H(ID_a \| x_a)$, cannot impersonate himself as anyother device say $\mathcal{D}_b$ to the victim $\mathcal{D}_a$. To launch KCIA $\mathcal{A}$ can be the initiator or the responder, and for responding role $\mathcal{A}$ can intercept the message $\{(AID_a, \tau_a, \gamma_a, t_a)\}$ sent by the $\mathcal{D}_a$ to $\mathcal{D}_b$. $\mathcal{A}$ cannot compute $\alpha_a = d_b\tau_b$ as it requires private key $d_b$ of $\mathcal{D}_b$. The inability of computing $\alpha_a$ is also extended to compute the identity $ID_a$ of initiator. Moreover, $\mathcal{A}$ cannot compute $K = K_b = yQ_b + d_b\tau_a$ because with known $d_a$ and the public key $Q_b$, finding $yQ_b + d_b\tau_a$ is elliptic curve discrete logarithm (ECDLP)—a hard problem. Hence $\mathcal{A}$ will also fail to compute $R_b$ and $SK$ as both also requires the knowledge of $K$. Similarly, in initiator case, $\mathcal{A}$ can compute $\tau_a = xG$, $\alpha_a = xQ_b$ and $AID_a = \alpha_a \oplus ID_i$ (With supposition that all identities are known to adversary). Similarly, after receiving the return message from $\mathcal{D}_b$, the adversary can also compute $ID_b$, but computing $K = K_a = xQ_b + d_a\tau_b$ is again intractable ECDLP. Therefore, the proposed protocol provides resistance against KCIA.

#### 7.3.2. Device Anonymity

The proposed scheme provides device anonymity and un-traceability [38,39]. In the proposed scheme, $\mathcal{D}_a$ sends his pseudo calculated identity $AID_a = \alpha_a \oplus ID_a$, any adversary just by listening the channel can get this pseudo identity and to compute original identity $ID_a$, the adversary needs to know $\alpha_a$, which is not sent on communication channel. The adversary can get $\tau_a = xG$ but computing $\alpha_a$ from $\tau_a$ needs the private key of the receiver $\mathcal{D}_b$, same private key is required to get the original identity $ID_b$ from pseudo calculated identity $AID_b$. Moreover, the temporary $ID$ is dynamically computed for each session. The proposed scheme provides identity hiding as well as resistance to traceability attack.

#### 7.3.3. Man-in-Middle Attack

For two devices ($\mathcal{D}_a$ and $\mathcal{D}_b$), the proposed protocol exchanges $\tau_a = xG$ and $\tau_b = yG$ and generates $K = xd_bG + yd_aG$ and session key $SK = H(ID_a \| ID_b \| \tau_a \| \tau_b \| K)$ using two private keys $d_a$ and $d_b$, and two session specific parameters $x$ and $y$ generated each participant. Since the devices can authenticate $R_a$ and $R_b$ very easily, a valid session key $SK$ is generated.Therefore, to get authenticated from other side, the attacker $\mathcal{A}$ requires the private key of the $2^{nd}$ participant as well as session specific temporary parameter generated on other side. Even if $\mathcal{A}$ can generate session specific parameter but computing private key out of public key is the hard ECDLP problem and computing $xd_bG + yd_aG$ from $d_aG$ and $d_bG$ is ECC Diffie-Hellman problem (ECDHP), which is also a hard problem. Thus, the proposed protocol provides protection against MIM attack.

### 7.3.4. Known-Key Attacks

Known-key-attack (KKA) is a cryptographic attack in which an adversary can access the ciphertext. Known-key-attacks can be attempted successfully by an adversary when the palintext is related with the ciphertext and the adversary could trace the plaintext by just performing backtracking. A $2PAKA$ protocol holds KKA property if a disclosure of whole or part of previously generated keys occur and such disclosure may not help to generate other past or future session keys. In the proposed protocol, each key is formed using private keys of both interconnected devices as well as their random numbers generated solely for formation of each session key and if an attacker $\mathcal{A}$ by some means gets one or more generated session keys, it may have no advantage in computing any other safe past or future keys and to expose any past or future keys $SK = H(ID_a||ID_b||\tau_a||\tau_b||K)$, $\mathcal{A}$ needs to compute $K$, $R_a$ and $R_b$ which are based on private keys and session specific parameters and are unknown to $\mathcal{A}$. Hence, proposed protocol resists KKS attack.

### 7.3.5. Unknown Key Share Attack (UKS)

By UKS, An entity, say $\mathcal{D}_x$ believes that a correct session key with other device $\mathcal{D}_y$ is accomplished and on other hand another device say $\mathcal{D}_y$ wrongly believes that the key is established with $\mathcal{A}$ instead of $\mathcal{D}_x$. In the proposed protocol, the session key computed on both sides is same and it requires the privates keys as well as identities of both the participants. Therefore, the proposed protocol is secure from UKS.

### 7.3.6. Backward/Forward secrecy

A protocol satisfies forward secrecy [40,41], if the private key of one or more participant but not all or some of the previously generated sessions keys are compromised, it may not effect future sessions keys. Similarly, in a protocol if compromise of current session key or some of the private keys cannot help to expose any previous session key, the protocol is said to be forward secure. The protocol is said to posses perfect forward secrecy if the compromise of all private keys have no effect on previously generated session keys. In the proposed protocol, even if the private keys of both participants are known to an adversary, he cannot compute any previously generated session key due to the inclusion of the session specific random parameters. Hence our device to device AKA provides $PFS$.

### 7.3.7. Known Session Specific Information Attack (KSSIA)

Resistance to KSSIA implies that, the exposure of all session parameters $(x, y)$ to $\mathcal{A}$, may not expose the session key. In the proposed device authentication protocol, both devices $\mathcal{D}_a$ and $\mathcal{D}_b$ compute $SK = H(ID_a||ID_b||\tau_a||\tau_b||K)$. $\mathcal{A}$ can reveal $SK$ if and only if he knows $K = K_a = xQ_b + yQ_a$ or $K = K_b = yQ_a + xQ_b$. Knowing only the pair $(x, y)$ may not help $\mathcal{A}$ to derive $K_a$ or $K_b$. Therefore, the proposed protocol resists KSSIA.

### 7.3.8. Key Off-Set/Replicating Attack

The key replicating attack ($KRA$) is a distinction of MIM attack, where one or more active adversaries intercept and modify the exchanged information between devices $\mathcal{D}_a$ and $\mathcal{D}_b$ in such a way that the modification results into agreement of an incorrect session key. In our proposed protocol the $\mathcal{D}_a$ and $\mathcal{D}_b$ exchange $\tau_a$ and $\tau_b$. $\mathcal{A}$ can modify some values by offset $\epsilon$ and produces $\epsilon\tau_a$ and $\epsilon\tau_b$. Nevertheless, $\mathcal{A}$ remains unable to compute $SK$ that is agreed by $\mathcal{D}_a$ and $\mathcal{D}_b$, as $\mathcal{A}$ requires the knowledge of the private keys $d_a$ and/or $d_b$. Hence, the proposed device to device key is resistance to key off-set/replicating attack ($KOA/KRA$).

### 7.3.9. No Key Control

The session key $SK = H(ID_a||ID_b||\tau_a||\tau_b||K)$ computed between $\mathcal{D}_a$ and $\mathcal{D}_b$ contains equal share of both participants, i.e., both participants add their session parameters as well as their private keys.

Therefore, none of the participant has any control on session key formation and proposal provides No Key Control (NKC) property.

#### 7.3.10. Replay Attack

Our proposed protocol is free from replay attack (RA). Any adversary can replay any old message say $\{AID_a, \tau_a, \gamma_a, t_a\}$ exchanged between to legal devices. However, the timestamp $t_a$ is also a part of message in plain text as well as hidden in $\gamma_a$. The receiver can easily detect the freshness and discard the message in case it is replayed. Same is the case, if against any request, the adversary replays an old reply message say $\{AID_j, \tau_j, R_j, t_j\}$, the initiator will easily detect the replay and will discard the message.

## 8. Performance Analysis

This section shows the comparative performance measure of the proposed protocol with existing protocols [13,31,32,36,37] in terms of computation and communication efficiency. Following notations and their running time computed by Kilinic and Yanik [42] on a Dual CPU $E2200$ with 2.20 GHz speed and with 2048 MB of RAM, were used for computation cost analysis:

- $T_{exp} \approx 3.85$ ms: Cost of modular exponentiation
- $T_{em} \approx 2.226$ ms: Cost of Point multiplication over ECC
- $T_{ea} \approx 0.0288$ ms: Cost of Point multiplication over ECC
- $T_h \approx 0.0023$ ms: Cost of hash function
- $T_{pb} \approx 5.811$ ms: Cost of bilinear pairing operation
- $T_{ed} \approx 0.0046$ ms: Cost of symmetric encryption

Table 3 shows a comprehensive performance comparisons; referring the table, the proposed scheme completed the key exchange process by performing $6T_{em} + 2T_{ea} + 8T_h$ operations and with running time $\approx 13.42$ ms. Mandal et al.'s protocol accomplished the same with $4T_{em} + 11T_{syd} + 12T_h$ operations and a running time of $\approx 8.9822$ ms. The protocol proposed by Islam-Biswas completed it in $\approx 13.3698$ ms by performing $6T_{em} + 6T_h$ operations. Wang et al.'s protocol performed $2T_{bp} + 4T_{em}$ operations and completed the authentication process in $\approx 20.5262$ ms. Holbl-Walzer protocols [13] accomplished authentication in $8T_{bp}$ and $6T_{bp}$ respectively with running time $\approx 46.48$ ms and $\approx 34.866$ ms respectively. The proposed protocol funished the authentication with slight higher computation time as compared with Mandal et al. and Islam-Biswas protocols, whereas it was efficient as compared with other related protocols. For communication cost, we considered an ECC point of size 160 bits, the output of hash function (SHA-1) is 160 bits and for simplicity identity was also taken as 160 bit, with timestamps of 32 bits length. The communication cost of the proposed protocol was just 168 bytes in comparison with Mandal et al.'s 252 bytes, Islam-Biswas's 120, Ni et al.'s 132 bytes, Wang et al.'s 66 bytes and Holbl-Walzer's 258 bytes. The communication cost of the proposed protocol was less than Mandal et al. and Holbl-Wazler's protocols and more than Islam-Biswas, Ni et al. and Wang et al.'s protocols. Therefore, the proposed protocol achieved a good trad-off between computation and communication efficiencies.

**Table 3.** Communication and Computation cost.

| Protocol | Bytes Exchanged | Computation Cost | Running Time |
|---|---|---|---|
| Holbl-Walzer I [13] | 258 | $8T_{bp}$ | 46.48 ms |
| Holbl-Walzer II [13] | 258 | $6T_{bp}$ | 34.866 ms |
| Wang et al. [36] | 66 | $2T_{bp} + 4T_{em}$ | 20.5262 ms |
| Ni et al. [37] | 132 | $2T_{bp} + 2T_{em} + 2T_{exp}$ | 23.7742 ms |
| Islam-Biswas [31] | 120 | $6T_{em} + 6T_h$ | 13.3698 ms |
| Mandal et al. [32] | 252 | $4T_{em} + 11T_{syd} + 12T_h$ | 8.9822 ms |
| Proposed | 168 | $8T_{em} + 2T_{ea} + 6T_h$ | 13.42 ms |

## 9. Conclusions

In this paper, we have simulated key compromise impersonation attack (KCIA) on two recent ECC and self certified public key based authentication protocols. It has been shown that both the protocols of Islam-Biswas and Mandal et al. are not only insecure against KCIA, but also lacking anonymity. We then proposed an improved protocol to resist KCIA and related known attacks and to provide anonymity and related important security features. Proposed scheme is tailored to work in IoT-based fast moving vehicular networks and does not require involvement of a third party for sharing a key between two smart vehicles. The security of proposed scheme is analyzed through formal and informal methods. Although, proposed protocol accomplishes the authentication with slight high computation and communication costs as compared with related protocols but it provides resistance against all known attacks and encompasses all required security features. Hence, the proposed protocol is best suited for key exchange in device to device using certificates.

**Author Contributions:** B.A.A. wrote the initial draft as well as revision and BAN logic analysis of the proposed scheme. S.A.C. conceptualized the idea and performed cryptanalysis and designed the new scheme. A.B., and A.A.-B. performed security and efficiency analysis. T.S. performed formal analysis, proof read and supervised the whole process. All authors have read and agreed to the published version of the manuscript.

**Funding:** This Project was funded by the Deanship of Scientific Research (DSR), at King Abdulaziz University, Jeddah, under grant no. (RG-7-611-40). The authors, therefore, acknowledge with thanks DSR for technical and financial support. This research was supported by Energy Cloud R&D Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2019M3F2A1073386).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chen, C.M.; Xiang, B.; Liu, Y.; Wang, K.H. A secure authentication protocol for internet of vehicles. *IEEE Access* **2019**, *7*, 12047–12057. [CrossRef]
2. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
3. Chen, T.H.; Lee, W.B.; Chen, H.B. A round-and computation-efficient three-party authenticated key exchange protocol. *J. Syst. Softw.* **2008**, *81*, 1581–1590. [CrossRef]
4. Lu, R.; Cao, Z. Simple three-party key exchange protocol. *Comput. Secur.* **2007**, *26*, 94–97. [CrossRef]
5. Phan, R.C.W.; Yau, W.C.; Goi, B.M. Cryptanalysis of simple three-party key exchange protocol (S-3PAKE). *Inf. Sci.* **2008**, *178*, 2849–2856. [CrossRef]
6. Chen, C.M.; Wang, K.H.; Yeh, K.H.; Xiang, B.; Wu, T.Y. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3133–3142. [CrossRef]
7. Pu, Q.; Zhao, X.; Ding, J. Cryptanalysis of a three-party authenticated key exchange protocol using elliptic curve cryptography. In Proceedings of the International Conference on Research Challenges in Computer Science, ICRCCS'09, Shanghai, China, 28–29 December 2009; pp. 7–10.
8. Tan, Z. An Enhanced Three-Party Authentication Key Exchange Protocol Using Elliptic Curve Cryptography for Mobile Commerce Environments. *J. Commun.* **2010**, *5*, 436–443. [CrossRef]
9. Tseng, Y.M. An efficient two-party identity-based key exchange protocol. *Informatica* **2007**, *18*, 125–136.
10. Günther, C.G. An identity-based key-exchange protocol. In Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, 10–13 April 1989; pp. 29–37.
11. Saeednia, S. Improvement of Günther's identity-based key exchange protocol. *Electron. Lett.* **2000**, *36*, 1535–1536. [CrossRef]
12. Hsieh, B.; Sun, H.; Hwang, T.; Lin, C. An improvement of Saeednia's identity-based key exchange protocol. *Inf. SecuR. Conf.* **2002**, *2002*, 41–43.
13. Hölbl, M.; Welzer, T. Two improved two-party identity-based authenticated key agreement protocols. *Comput. Stand. Interfaces* **2009**, *31*, 1056–1060. [CrossRef]
14. Zhang, S.; Cheng, Q.; Wang, X. Impersonation attack on two identity-based authenticated key exchange protocols. In Proceedings of the 2010 WASE International Conference on Information Engineering, Beidaihe, China, 14–15 August 2010.

15. Smart, N. Identity-based authenticated key agreement protocol based on Weil pairing. *Electron. Lett.* **2002**, *38*, 630–632. [CrossRef]
16. Chen, L.; Kudla, C. Identity based authenticated key agreement protocols from pairings. In Proceedings of the 16th IEEE Computer Security Foundations Workshop, Pacific Grove, CA, USA, 30 June–2 July 2003; pp. 219–233.
17. Shim, K. Efficient ID-based authenticated key agreement protocol based on Weil pairing. *Electron. Lett.* **2003**, *39*, 653–654. [CrossRef]
18. Sun, H.M.; Hsieh, B.T. Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings. *IACR Cryptol. EPrint Arch.* **2003**, *2003*, 113.
19. Ryu, E.K.; Yoon, E.J.; Yoo, K.Y. An efficient ID-based authenticated key agreement protocol from pairings. In *International Conference on Research in Networking*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 1458–1463.
20. Boyd, C.; Choo, K.K.R. Security of two-party identity-based key agreement. In Proceedings of the International Conference on Cryptology in Malaysia, Kuala Lumpur, Malaysia, 28–30 September 2005; pp. 229–243.
21. McCullagh, N.; Barreto, P.S. A new two-party identity-based authenticated key agreement. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 14–18 February 2005; pp. 262–274.
22. Shao, Z.-H. Efficient authenticated key agreement protocol using self-certified public keys from pairings. *Wuhan Univ. J. Nat. Sci.* **2005**, *10*, 267–270.
23. Ni, L.; Chen, G.; Li, J.; Hao, Y. Strongly secure identity-based authenticated key agreement protocols. *Comput. Electr. Eng.* **2011**, *37*, 205–217. [CrossRef]
24. Cao, X.; Kou, W.; Yu, Y.; Sun, R. Identity-based authentication key agreement protocols without bilinear pairings. *IEICE Trans. Fundam.* **2008**, *12*, 3833–3836. [CrossRef]
25. Tsaur, W.J. Several security schemes constructed using ECC-based self-certified public key cryptosystems. *Appl. Math. Comput.* **2005**, *168*, 447–464. [CrossRef]
26. Hölbl, M.; Welzer, T.; Brumen, B. An improved two-party identity-based authenticated key agreement protocol using pairings. *J. Comput. Syst. Sci.* **2012**, *78*, 142–150. [CrossRef]
27. Chen, L.; Cheng, Z.; Smart, N.P. Identity-based key agreement protocols from pairings. *Int. J. Inf. Secur.* **2007**, *6*, 213–241. [CrossRef]
28. Choo, K.K.R.; Boyd, C.; Hitchcock, Y.; Maitland, G. On session identifiers in provably secure protocols. In Proceedings of the International Conference on Security in Communication Networks, Amalfi, Italy, 8–10 September 2004; pp. 351–366.
29. Li, S.; Yuan, Q.; Li, J. Towards Security Two-part Authenticated Key Agreement Protocols. *IACR Cryptol. EPrint Arch.* **2005**, *2005*, 300.
30. Wang, S.; Cao, Z.; Choo, K.K.R.; Wang, L. An improved identity-based key agreement protocol and its security proof. *Inf. Sci.* **2009**, *179*, 307–318. [CrossRef]
31. Islam, S.H.; Biswas, G. Design of two-party authenticated key agreement protocol based on ECC and self-certified public keys. *Wirel. Pers. Commun.* **2015**, *82*, 2727–2750. [CrossRef]
32. Mandal, S.; Mohanty, S.; Majhi, B. Cryptanalysis and Enhancement of an Anonymous Self-Certified Key Exchange Protocol. *Wirel. Pers. Commun.* **2018**, *99*, 863–891. [CrossRef]
33. Khatwani, C.; Roy, S. Security Analysis of ECC Based Authentication Protocols. In Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 12–14 December 2015; pp. 1167–1172.
34. Chaudhry, S.A.; Shon, T.; Al-Turjman, F.; Alsharif, M.H. Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems. *Comput. Commun.* **2020**, *153*, 527537. [CrossRef]
35. Mansoor, K.; Ghani, A.; Chaudhry, S.A.; Shamshirband, S.; Ghayyur, S.A.K.; Mosavi, A. Securing IoT-Based RFID Systems: A Robust Authentication Protocol Using Symmetric Cryptography. *Sensors* **2019**, *19*, 4752. [CrossRef]
36. Wang, S.; Cao, Z.; Cao, F. Efficient Identity-based Authenticated Key Agreement Protocol with PKG Forward Secrecy. *Int. J. Netw. Secur.* **2008**, *7*, 181–186.
37. Ni, L.; Chen, G.; Li, J.; Hao, Y. Strongly secure identity-based authenticated key agreement protocols in the escrow mode. *Sci. China Inf. Sci.* **2013**, *56*, 1–14. [CrossRef]

38. He, D.; Kumar, N.; Khan, M.K.; Wang, L.; Shen, J. Efficient Privacy-Aware Authentication Scheme for Mobile Cloud Computing Services. *IEEE Syst. J.* **2018**, *12*, 1621–1631. [CrossRef]
39. Zhang, L.; Zhang, Y.; Tang, S.; Luo, H. Privacy Protection for E-Health Systems by Means of Dynamic Authentication and Three-Factor Key Agreement. *IEEE Trans. Ind. Electron.* **2018**, *65*, 2795–2805. [CrossRef]
40. Hussain, S.; Chaudhry, S.A. Comments on "Biometrics-Based Privacy-Preserving User Authentication Scheme for Cloud-Based Industrial Internet of Things Deployment". *IEEE Internet Things J.* **2019**, *6*, 10936–10940. [CrossRef]
41. Ghani, A.; Mansoor, K.; Mehmood, S.; Chaudhry, S.A.; Rahman, A.U.; Najmus Saqib, M. Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key. *Int. J. Commun. Syst.* **2019**, *32*, e4139. [CrossRef]
42. Kilinc, H.H.; Yanik, T. A survey of SIP authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1005–1023. [CrossRef]