Key Agreement Between User and Drone With Forward Unlinkability in Internet of Drones

JAE YEOL JEONG\textsuperscript{1}, JIN WOOK BYUN\textsuperscript{2,} (Member, IEEE), AND IK RAE JEONG\textsuperscript{1}
\textsuperscript{1}School of CyberSecurity, Korea University, Seongbuk-gu, Seoul 02841, South Korea
\textsuperscript{2}Department of Information and Communication, Pyeongtaek University, Pyeongtaek-si, Gyeonggi-do 450-701, South Korea

Corresponding authors: Jin Wook Byun (jwbyun@ptu.ac.kr) and Ik Rae Jeong (irjeong@korea.ac.kr)

This work was supported as part of Military Crypto Research Center (UD210027XD) funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD), and a Korea University Grant.

\section{ABSTRACT}
Many applications are equipped to utilize drones to provide various services to users in Internet of Drones (IoD) environments. In such applications, it is necessary to make a session key between a drone and a user to establish an authenticated and secure channel. It is also desirable to provide strong anonymity to increase user(drone) privacy. To provide robust anonymity, a protocol has to provide both pseudonymity and unlinkability. If a protocol provides only pseudonymity without unlinkability, user(drone) privacy could be breached by analyzing communication frequency or user(drone) movement. On the other hand, we consider drone capture attacks in the IoD, because if a drone is captured, the secret information of the drone could be revealed. To minimize the damage against the capturing attacks, a key exchange should provide the forward unlinkability as well as forward secrecy. Forward unlinkability means that even though the secret information of the drone is revealed, the unlinkability is guaranteed. In the paper, we suggest the first key agreement protocols providing both pseudonymity and forward unlinkability, whereas previous key agreement protocols provide only pseudonymity and unlinkability.

\section{INDEX TERMS}
Internet of Drones (IoD), anonymity, pseudonymity, unlinkability, authentication, key agreement, forward secrecy.

\section{I. INTRODUCTION}
Owing to the rapid development of unmanned aerial technology, various services based on the Internet of Drones (IoD) have been developed in real life \cite{1}. Since contemporary drones are equipped with high-performance sensors, IoD is proving useful in a number of ways, including during disasters, for military operations, and transportation processes. For instance, unmanned drones have been used in COVID-19 management systems, in which the drones fly over a certain area (or quarantined area) and send alert messages to the gathered people. And agricultural drones can be used for farming \cite{2}. Drones are also useful in collecting several kinds of data and these collected data can be stored in blockchains \cite{3}. As illustrated in Fig. 1, IoD consists of drones, users, and the control server.

In the IoD, it is necessary to authenticate the communicating parties and protect the messages exchanged between the parties from adversaries. For authentication and confidentiality between a user and a drone, we need an authenticated key exchange protocol to make a session key. With the session key, the user and the drone can create a secure channel to protect the messages from adversaries. Recently, authentication with an efficient key exchange

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{IoD.png}
\caption{IoD environment.}
\end{figure}
between users and drones has become a popular research topic in the IoD settings [4]–[7].

In some applications, pseudonymity and unlinkability can be crucial. For example, during a military operation, soldiers can utilize information collected by drones. Drones can move around the operation field to collect information, and soldiers also independently move the field. Whenever a soldier needs additional information, they are able to establish a session key and communicate with the drone. In this scenario, we have to assume that an adversary is eavesdropping on the communications between the drones and the soldiers.

In a key exchange protocol, we can use pseudonyms for users and drones to provide privacy. We note that pseudonymity alone without unlinkability does not provide strong anonymity. For instance, if an adversary can observe the same pseudonym of a user(drone), it can also trace the frequency of communications or the movements of the user(drone). This could eventually trace back to the real identity of the user(drone), and thus break user(drone) privacy. Therefore, in the above scenario, the knowledge of the movements of the drones and the soldiers due to the lack of unlinkability could lead to a failed military operation.

In the IoD environment, it is possible to capture a drone and analyze the drone to extract some useful information. This attack is called drone capturing attack [8]. In a key exchange protocol the drone capturing attacks make an adversary extract the long-term secret key stored in a drone, and use the secret key to break the secrecy of the session keys or unlinkability of the sessions. Therefore, a key exchange should provide the forward unlinkability to minimize the damage against the capturing attacks. Forward unlinkability means that even though the secret information of the drone is revealed, the unlinkability is guaranteed.

### A. RELATED WORKS AND CONTRIBUTIONS

Wireless Sensor Networks (WSN), Internet of Things (IoT), and Internet of Drones (IoD) environments have similar properties and security requirements. Therefore, a key exchange protocol developed for WSN or IoT can be usable with minor modifications in the IoD environment.

In 2013, Xie et al. proposed a chaotic, maps-based, three-party password-authentication key agreement scheme [9]. However, in 2015, Lee et al. found security flaws in Xie et al.’s scheme [10]. In order to solve these security flaws, Lee et al. proposed a three-party-authenticated key agreement scheme based on chaotic maps without a password table [10]. However, the chaotic maps are less efficient than symmetric key-based encryption or hash functions [11].

In 2014, Turkanovic et al. proposed an efficient authenticated key exchange scheme in WSN [12]. Unfortunately, Turkanovic et al.’s scheme was later found to be insecure due to numerous attack types, such as man-in-the-middle, smart card, and sensor node impersonation attacks. To overcome the weaknesses of this scheme, Farash et al. proposed a new scheme in a heterogeneous WSN [13]. However, in 2016 Amin et al. showed that the Farash et al.’s scheme is vulnerable to a known session-specific temporary information attack, an off-line password guessing attack using a stolen smart card, a new smart card issue attack, and a user impersonation attack [14]. In 2017, Tai et al. designed a new authentication scheme in the same setting as the experiments conducted by Farash et al [15]. But this scheme is also not secure against other attack types, such as privileged-insider, password-guessing, man-in-the-middle, and replay attacks. Also in 2017, Jiang et al. observed that Amin et al.’s scheme in [14] is neither secure against smart card loss attacks nor known session-specific temporary information attacks [16].

In IoT settings, some ECC-based key exchange schemes already exist [17], [18]. In 2019, Wazid et al. suggested device authentication and a key management protocol for IoT devices, cloud servers, and edge nodes in edge-based IoT environments [19].

In 2018, Wazid et al. proposed a novel and lightweight remote user authentication and key agreement scheme in the IoD setting [5]. This scheme enables a valid user to directly receive collected data from a drone after establishing a session key via a key agreement protocol. In 2019, Srinivas et al. proposed a more efficient scheme than that proposed by Wazid et al. [6]. However, the schemes in [5] and [6] are not secure against other legal users, since a legal user can calculate session keys established by other legal users. In 2020, Zhang et al. suggested a lightweight authenticated key agreement protocol using only a hash function and a bitwise XOR operation [7]. Overall, the schemes in [5]–[7] provide pseudonymity, but they do not provide unlinkability as pseudonyms are revealed and do not change between sessions.

There are already some authentication and communication protocols for drones. In 2020, Yazdinejad et al. proposed a drone authentication scheme using a shared ledger for the blockchain [20]. This scheme enables the drone information to be recorded in the blockchain when the drone is certified by the drone controller in a certain zone. If this is accomplished, the drone does not need to be re-certified, even if it needs to move to another zone due to the information registered in the blockchain. In 2020, Sharma et al. proposed a similar scheme, in which a drone can query to deploy sensor nodes to collect information [21].

There exist some works to make key exchange protocols that satisfy unlinkability in various environments such as IoT and ad-hoc networks. However, all of the existing protocols do not satisfy forward unlinkability. That is, if a secret information of a communicating party is revealed, it is possible to check whether the same communicating party participated in the two different sessions.

In 2015, Buttnner and Huss proposed a new anonymous authentication key exchange protocol for vehicular ad hoc networks [22]. Even though the protocol tries to achieve the unlinkability using the Elliptic Curve Integrated Encryption Scheme (ECIES) and the ring signature, it is obvious that we can verify the identity of a communicating party if the party’s
private key is revealed. Therefore, it does not satisfy forward unlinkability.

In 2017, Shin and Kwon proposed an anonymous authentication key exchange protocol between a user and an IoT gateway in an IoT environment [23]. This protocol allows users and the IoT App Server to make a session key. However, this protocol does not satisfy forward unlinkability, because the pseudonyms of the same user is linkable if the server’s secret key is revealed.

In 2021, Li et al. proposed an anonymous authentication key exchange protocol between vehicles using homomorphic encryption [24]. A vehicle first tries to anonymously authenticate itself to RSU using pseudonyms and tickets, and then vehicles try to anonymously authenticate each other. However, it is easy to check whether any two tickets belong to the same vehicle. Therefore, the protocol does not satisfy forward unlinkability.

In 2021, Khan et al. proposed an anonymous authentication key exchange protocol between drones and users in the IoD environment. In our proposed schemes, users and drones are registered to the control server. Our first key agreement protocol makes a drone initiate a key exchange protocol with a user. Our second key agreement protocol makes a user initiate a key exchange protocol with a drone.

Our two key agreement protocols use only symmetric cryptographic primitives such as symmetric encryption, MAC, and hash functions. To provide pseudonymity and forward unlinkability, the two key agreement protocols use different pseudonyms and secret keys in different sessions.

Our key agreement protocols also provide forward secrecy. Usually, forward secrecy is provided using asymmetric cryptographic primitives such as the ephemeral Diffie-Hellman key exchange which are expensive operations compared to the symmetric cryptographic primitives. Even though our protocols use only symmetric cryptographic primitives, our protocols provide 1-less full forward secrecy which is a variant of the full forward secrecy.

### II. CRYPTOGRAPHIC PRIMITIVES

$H : \{0, 1\}^* \rightarrow \{0, 1\}^\theta$ is a cryptographically secure hash function, where $\theta$ is a security parameter.

**Definition 1 (SUF Secure MAC [26]):** We consider MAC $\text{Mac} = (\text{KeyGen}, \text{Mac}, \text{Vrfy})$. To define SUF security for $\text{Mac}$, we suppose an adversary $\mathcal{F}$ that can access MAC generation oracle $\text{Mac}_{\theta}(\cdot)$ and MAC verification oracle $\text{Vrfy}_{\theta}(\cdot)$. Then, SUF security is defined by the following experiment:

$$\begin{align*}
\text{Exp}^{\text{SUF}}_{\text{Mac}, \mathcal{F}}(\theta) \\
\text{mk} \leftarrow \text{KeyGen}(1^\theta)
\end{align*}$$

$\mathcal{F}$ can access oracles $\text{Mac}_{\theta}(\cdot)$ and $\text{Vrfy}_{\theta}(\cdot)$, and we define an advantage of $\mathcal{F}$ as follows:

$$\text{Adv}^\text{SUF}_{\mathcal{F}} = \Pr[\text{Vrfy}_{\theta}(m, \tau) = 1 \text{ and } (m, \tau) \text{ is not in the set of query-response pairs}].$$

A $\text{Mac}$ is SUF secure if the advantage of any probabilistic polynomial-time adversary $\mathcal{F}$ is negligible in terms of the security parameter $\theta$.

**Definition 2 (LoR-CPA Secure Symmetric Encryption [26]):** For our study, we consider a symmetric encryption scheme $\text{SE} = (\text{Key}, E, D)$, and we suppose that an adversary $\mathcal{D}$ can access encryption oracle $E_{\text{sk}}(LR_{\cdot}, \cdot, b))$. The left-or-right chosen-plaintext attack (LoR-CPA) security for $\text{SE}$ is defined by the following experiment:

$$\begin{align*}
\text{Exp}^{\text{LoR-CPA}}_{\text{SE}, \mathcal{D}}(\theta) \\
\text{sk} \leftarrow \text{Key}(1^\theta) \\
b' \leftarrow D_{\text{sk}}(E_{\text{sk}}(LR_{\cdot}(\cdot, b))))(\theta)
\end{align*}$$

The advantage of $\mathcal{D}$ is defined as

$$\text{Adv}^\text{LoR}_{\mathcal{D}} = \Pr[\text{Exp}^{\text{LoR-CPA}}_{\text{SE}, \mathcal{D}}(\theta) = 1] - \Pr[\text{Exp}^{\text{LoR-CPA}}_{\text{SE}, \mathcal{D}}(\theta) = 1],$$

where $\theta$ is a security parameter.

$\text{SE}$ is LoR-CPA secure if the advantage of any probabilistic polynomial-time adversary $\mathcal{D}$ is $1/2 + \epsilon(\theta)$, where $\epsilon(\theta)$ is a negligible function.

### III. SECURITY MODEL

**Full forward secrecy** means that even if all the secret keys of the communicating parties are disclosed, the session keys before the disclosure are still secure. The full forward secrecy of a key agreement protocol is defined by an experiment. In the experiment, an adversary $\mathcal{A}$ asks Initiate, Send, Reveal, Corrupt, and Test queries, and it receives the messages according to the protocol description. We assume that $U_i$ denotes users, $D_j$ as drones, and $S$ as the server, where $i \in [1, n]$ and $j \in [1, m]$. Each entity $P_i$, where $P_i \in \{U_i, D_j, S\}$, may have multiple instances.

- An Initiate$(P_i, k)$ query is used to instigate a key agreement protocol, where $P_i \in \{U_i, D_j\}$. $P_i$ returns the first message as its response according to the protocol description.
- A Send$(P_i, k, m)$ query is used to send message $m$ to party $P_i$’s $k$-th instance, where $P_i \in \{U_i, D_j, S\}$. After receiving $m$, $P_i$ returns a message as its response according to the protocol description.
- A Reveal$(P_i, k)$ query is used to get a session key made in party $P_i$’s $k$-th instance, where $P_i \in \{U_i, D_j\}$.
- A Corrupt$(P_i, k)$ query is used to get the long-term secret key of party $P_i$, where $P_i \in \{U_i, D_j\}$. $P_i$ returns its long-term secret key as its response.
A Test($P_i, k$) query is used to define the advantage of an adversary, where $P_i \in \{U_i, D_j\}$. $P_i$ flips a coin $\sigma \in \{0, 1\}$. If $\sigma = 1$, $P_i$ returns a real session key of the $k$-th instance. Otherwise, $P_i$ returns a random value. We note that this query is valid only when the $k$-th instance of $P_i$ is fresh (defined below).

The $k$-th session of party $P_i$, $P_i \in \{U_i, D_j\}$, is fresh if the following conditions hold:
1) If Corrupt($P_i$) or Corrupt($P_j$) has been asked, the $k$-th session of party $P_i$ was made before Corrupt($P_i$) and/or Corrupt($P_j$) query, where $P_i$, $P_j (\neq P_i) \in \{U_i, D_j\}$.
2) Reveal($P_i, k$) has not been asked.
3) If the two instances calculated the same session key, they are matching instances. Reveal($P_i, k$) has not been asked if the $k$-th instance of $P_i$ and the $\ell$-th instance of $P_j$ are matching instances, where $P_j \in \{U_i, D_j\}$.

To terminate the experiment, the adversary $A$ outputs $\sigma'$ to guess $\sigma$ and stops. The advantage of $A$ is defined by

$$\text{Adv}^\text{KE}_A = Pr[\sigma = \sigma'].$$  

A key agreement protocol is “secure” if the advantage of any probabilistic polynomial-time adversary $A$ is $1/2 + \epsilon(\theta)$ in the security parameter $\theta$, where $\epsilon(\theta)$ is a negligible function.

We note that our security model deals with various attack types, such as impersonation, drone capture, and compromised user device attacks.

**1-Less Full Forward Secrecy:** We define 1-less full forward secrecy which is a variant of the full forward secrecy. In the experiment of the 1-less full forward secrecy, an adversary can test the instances of $P_i$ except the ones which are made after the second to last instance having a matching instance of $P_j$ and before Corrupt($P_i$) or Corrupt($P_j$) is asked. The difference between the full forward secrecy and the 1-less full forward secrecy is depicted in Fig. 2.

![FIGURE 2. forward secrecy and 1-less forward secrecy.](image)

**IV. OUR KEY AGREEMENT PROTOCOLS**

In this section, we outline our key agreement protocols, which require registration protocols.

In the registration protocols, a user(drone) registers its identity, pseudonym, and a secret key to the control server which are shared between the user(drone) and the server.

All protocols throughout this paper are explained using the following common notations.

![FIGURE 3. User registration protocol.](image)

![FIGURE 4. Drone registration protocol.](image)
B. OUR KEY AGREEMENT PROTOCOLS

During execution of our protocols, each communicating party should check the followings.

1) There should be only one unfinished instance in a user(drone). That is, if a user(drone) has to make session keys with several other drones(users), each session for each session key in the user(drone) is executed sequentially, not concurrently.

2) A party checks the timeout for a protocol message in a session. That is, if a party has not received a protocol message before timeout, the party stops and exits the session.

Our first key agreement protocol, a drone-initiated key agreement protocol, makes a session key between $U_i$ and $D_j$ as follows.

1) To begin a protocol, $D_j$ first sends $PD_j$ to $U_i$.

2) After receiving $PD_j$, $U_i$ decrypts $v_i$ with the password $pw_i$ and obtains $PU_i[α_i]$. Then, $U_i$ calculates encryption key $e_k = H(α_i||0)$ and MAC key $mk_i = H(α_i||1)$. $U_i$ randomly selects $r_{ui}$, $PU_i'$ and $α_i'$, and it generates ciphertext $c_i = E_{ek}(U_i[α_i'][PD_j][r_{ui}])$ and MAC value $τ_i = Mac_{mk_i}(PU_i'[S][c_i])$. $U_i$ then sends $PU_i'[S][c_i][τ_i]$ to $S$.

3) After receiving $PU_i'[S][c_i][τ_i]$, $S$ finds $U_i$ such that $PU_i \in \{PU_i^0, PU_i^1\}$. $S$ calculates the two keys $e_k, mk_i$ with the corresponding $α_i'$ or $α_i''$, decrypts $c_i$, and obtains $U_i[PU_i'[α_i'][PD_j][r_{ui}]]$. If $Vrfy_{mk_i}(PU_i'[S][c_i][τ_i]) = 1$, $S$ finds $D_j$ such that $PD_j \in \{PD_j^0, PD_j^1\}$. Then, $S$ calculates the two keys $e_k, mk_j$ with the corresponding $β_j'$ or $β_j''$ and makes ciphertext $c_j = E_{ek}(PU_i'[α_i''][PD_j][r_{uj}])$ and MAC value $τ_j = Mac_{mk_j}(PU_i'[S][c_j])$. $S$ sends $PD_j[τ_j][c_j]$ to $D_j$.

4) After receiving $PD_j[τ_j][c_j]$, $D_j$ calculates the two keys $e_k, mk_j$. $D_j$ decrypts $c_j$ and obtains $PU_i[r_{uj}]$. If $Vrfy_{mk_j}(PD_j[τ_j][c_j]) = 1$, $D_j$ randomly selects $r_{dj}, PD_j^0$ and $PD_j^1$. $D_j$ makes ciphertext $d_j = E_{ek}(PD_j^0)[β_j''][r_{dj}]$ and MAC value $τ_j = Mac_{mk_j}(PD_j^0[β_j''][r_{dj}])$, and then it sends $PD_j[τ_j][d_j]$ to $S$. $D_j$ generates the session key $sk_{ij} = H(PU_i'[PD_j][r_{uj}][r_{dj}][1])$ and another MAC key $mk_{ij} = H(PU_i'[PD_j][r_{uj}][r_{dj}][0])$.

5) After receiving $PD_j'[S][d_j][τ_j]$, $S$ decrypts $d_j$ and obtains $PD_j'[β_j''][r_{dj}][σ_j]$. If $Vrfy_{mk_j}(PD_j'[S][d_j][τ_j]) = 1$, $S$ makes ciphertext $d_j = E_{ek}(PD_j'[β_j''][r_{dj}][σ_j])$ and MAC value $τ_j = Mac_{mk_j}(PD_j'[S][d_j][σ_j])$, and then it sends $S[PU_i'[d_j][σ_j]]$ to $U_i$. $S$ stores $(U_i, PU_i^0, PU_i^1), α_i'' = α_i', α_i'' = α_i', (D_j, PD_j^0, PD_j^1) \in \{PD_j^0, PD_j^1\}, β_j'' = β_j', β_j'' = β_j')$.

6) After receiving $S[PU_i'[d_j][σ_j]], U_i$ decrypts $d_j$ and obtains $PD_j'[β_j''][r_{dj}][σ_j]$. If $Vrfy_{mk_j}(S[PU_i'[d_j][σ_j]]) = 1$, $U_i$ calculates MAC key $mk_{ij} = H( PU_i'[PD_j][r_{uj}][rdj][0] )$ and session key $sk_{ij} = H( PU_i'[PD_j][r_{uj}][rdj][1] )$. $U_i$ makes MAC value $δ_{ij} = Mac_{mk_{ij}}(PU_i'[PD_j][r_{uj}][rdj][0])$ and sends $PU_i'[PD_j][δ_{ij}]$ to $D_j$. $U_i$ stores $v_i = E_{H(U_i[pw])}(PU_i'[α_i'])$.

7) After receiving $PU_i'[PD_j][δ_{ij}], D_j$ checks if $Vrfy_{mk_{ij}}(PU_i'[PD_j][δ_{ij}]) = 1$. If it is valid, $D_j$ stores $(D_j, PD_j^0, β_j')$.

Our second key agreement protocol, a user-initiated key agreement protocol is similar to our drone-initiated key agreement protocol except that an initiator is a user instead of a drone as shown in Fig. 6.

V. SECURITY AND EFFICIENCY ANALYSES

Theorem 1: The drone-initiated key agreement protocol is 1-less-full-forward-secure in the random oracle model if $Mac$ is SUF secure and $SE$ is LoR-CPA secure.

The proof of Theorem 1 appears in the appendix.

Theorem 2: The user-initiated key agreement protocol is 1-less-full-forward-secure in the random oracle model, if $Mac$ is SUF secure and $SE$ is LoR-CPA secure.

The proof of Theorem 2 is similar to Theorem 1, and so we have omitted this from our paper.

A. ANONYMITY AND FORWARD UNLINKABILITY

The most pressing issue in the IoD environment is to protect the privacy of both the users and the drones. To provide anonymity, many previous key exchange schemes considered pseudonymity and unlinkability for the users and the drones, but forward unlinkability has not been considered in a key exchange protocol.

Our key agreement protocols provide not only pseudonymity but also forward unlinkability to ensure strong anonymity. In a session of our key agreement protocols, user and drone create new pair of pseudonym and secret key, stored as $(PU_i^0, α_i^0)$ and $(PD_j^0, β_j^0)$, respectively, for the next session which are independently selected of the current pair of pseudonym and secret key, stored as $(PU_i^0, α_i^0)$ and $(PD_j^0, β_j^0)$, respectively. Because $(PU_i^0, α_i^0)$ and $(PU_i^0, α_i^0)$ are independent, the protocol messages of any two sessions of user $U_i$ are unlinkable. Similarly, the protocol messages of any two sessions of drone $D_j$ are unlinkable, since $(PD_j^0, β_j^0)$ and $(PD_j^0, β_j^0)$ are independent. We compare anonymity among related key exchange protocols in Table 1.

### Table 1. Pseudonymity and forward unlinkability of related schemes.

| Scheme | Pseudonymity | Forward Unlinkability | LoR-CPA security | SUF security |
|--------|--------------|-----------------------|------------------|-------------|
| TRH [3] | Yes          | Yes                   | No               | Yes         |
| FTCH [13] | Yes        | Yes                   | No               | Yes         |
| ABER [14] | Yes        | Yes                   | No               | Yes         |
| TCL [15] | Yes        | Yes                   | No               | Yes         |
| ZMH [16] | Yes        | Yes                   | No               | Yes         |
| CWSK [17] | Yes        | Yes                   | No               | Yes         |
| WDONG [18] | Yes       | Yes                   | No               | Yes         |
| DWRDP [19] | Yes       | Yes                   | No               | Yes         |
| WDSK [19] | Yes       | Yes                   | No               | Yes         |
| SDVR [20] | Yes       | Yes                   | No               | Yes         |
| ZHLC [21] | Yes       | Yes                   | No               | Yes         |
| BH [22] | Yes        | Yes                   | No               | Yes         |
| SK [23] | Yes        | Yes                   | No               | Yes         |
| LQVIA [24] | Yes      | Yes                   | No               | Yes         |
| KOM [25] | Yes       | Yes                   | No               | Yes         |
**B. EFFICIENCY**

Due to limited resources, computational and communication costs are crucial factors in the IoD environments. To make efficient key agreement protocols, our two key agreement protocols use only symmetric cryptographic primitives, such as hash functions, symmetric encryption, and MACs.

Usually, full forward secrecy is provided using asymmetric cryptographic primitives such as the ephemeral Diffie-Hellman key exchange which are expensive operations compared to the symmetric cryptographic primitives. For instance, an ECC multiplication for the ephemeral Diffie-Hellman key exchange in ECC is 100 times more expensive than the symmetric primitives [28].

Khan et al.’s scheme uses only symmetric cryptographic primitives and provides partial forward secrecy [25]. That is, if only the secret key of a user is disclosed, the session key is still secure. However, if the secret key of the server is disclosed, the session key is not secure.

Even though our protocols use only symmetric cryptographic primitives, our protocols provide 1-less full forward secrecy which is a variant of the full forward secrecy. Therefore, our protocols are the most efficient ones among the key exchange protocols providing a variant of full forward secrecy.

Our protocols are a little bit inefficient with respect to the number of rounds, but the most efficient ones among the key exchange protocols providing full forward secrecy with respect to the computational cost. In Table 2, we analyzed the number of rounds, the total size of messages, and the total computations with respect to a user, a server, and a drone.
VI. CONCLUSION

In this paper, we proposed two authenticated key agreement protocols between drones and users in the IoD environment.

Our first key agreement protocol makes a drone start a key exchange protocol. Our second key agreement protocol makes a user start a key exchange protocol.
Our key agreement protocols are the first to provide both pseudonymity and forward unlinkability for users and drones in the IoD environment. Moreover, our key agreement protocols provide a variant of full forward secrecy.

In our protocols a party should execute several sessions sequentially. As a future research, it would be interesting to construct a key agreement protocol with forward unlinkability and forward secrecy, where a party can run several sessions concurrently.

**APPENDIX. PROOF OF THEOREM V.1**

Let $\mathcal{A}$ be a polynomial-time adversary against the drone-initiated key agreement protocol. Then, we show that $\mathcal{A}$’s advantage is bounded as follows:

$$\text{Adv}^{KE}(\theta, n, m, q_s) \leq \frac{1}{2} + (n + m)q_s \cdot (\text{Adv}^{SUF} + \text{Adv}^{LoR}),$$

where $n$ is the number of users, $m$ is the number of drones, $q_s$ is the maximum number of sessions for a party, and $\theta$ is a security parameter.

$\mathcal{A}$’s advantage is from the following two cases:

- **Case 1.** There are forged MACs with respect to $\tau$ or $\epsilon$ made by $\mathcal{A}$.
- **Case 2.** There is no forged MAC with respect to $\tau$ or $\epsilon$.

We bound the advantage from each case in the following lemmas.

**Lemma 1:** The advantage from Case 1 is bounded as

$$\text{Adv}^{KE, \text{Case 1}} \leq (n + m)q_s \cdot \text{Adv}^{SUF}.$$  \hspace{1cm} (5)

**Lemma 2:** The advantage from Case 2 is bounded as

$$\text{Adv}^{KE, \text{Case 2}} \leq \frac{1}{2} + (n + m)q_s \cdot \text{Adv}^{LoR}.$$  \hspace{1cm} (6)

Therefore, the advantage of $\mathcal{A}$ is bounded as follows:

$$\text{Adv}_{\mathcal{A}, \text{game} 0} = \text{Adv}^{KE, \text{Case 1}}_{\mathcal{A}, \text{game} 0} + \text{Adv}^{KE, \text{Case 2}}_{\mathcal{A}, \text{game} 0}$$

$$\leq \frac{1}{2} + (n + m)q_s \cdot \text{Adv}^{SUF}$$

$$+ (n + m)q_s \cdot \text{Adv}^{LoR}$$

$$= \frac{1}{2} + (n + m)q_s \cdot (\text{Adv}^{SUF} + \text{Adv}^{LoR})$$  \hspace{1cm} (7)

Next, we prove the above two lemmas. To do so, we define the following games:

- **game$_0$: game$_0$** is the original game defined in the experiment for key agreement protocols.

- **game$_1$: game$_1$** is the same as game$_0$ except that $ru_1$ of $U_1$ is replaced by random $ru_1^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_1$ of $U_1$ has not changed. $\alpha_1'$ of $U_1$ in $c_i$ is also replaced by random $\alpha_1^{*e}$ if $\alpha_1$ of $U_1$ has not changed.

- **game$_2$: game$_2$** is the same as game$_1$ except that $ru_1$ of $U_1$ is replaced by random $ru_1^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_1$ of $U_1$ has not changed.

- **game$_k$: game$_k$** is the same as game$_{k-1}$ except that $ru_1$ of $U_1$ is replaced by random $ru_1^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_1$ of $U_1$ has changed $k - 1$ times. $\alpha_1'$ of $U_1$ in $c_i$ is also replaced by random $\alpha_1^{*e}$ if $\alpha_1$ of $U_1$ has changed $k - 1$ times.

- **game$_{q_1}$: game$_{q_1}$** is the same as game$_{q_1-1}$ except that $ru_1$ of $U_1$ is replaced by random $ru_1^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_1$ of $U_1$ has changed $q_s - 1$ times. $\alpha_1'$ of $U_1$ in $c_i$ is also replaced by random $\alpha_1^{*e}$ if $\alpha_1$ of $U_1$ has changed $q_s - 1$ times.

- **game$_{q_1+i}$: game$_{q_1+i}$** is the same as game$_{q_1+i-1}$ except that $ru_2$ of $U_2$ is replaced by random $ru_2^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_2$ of $U_2$ has not changed. $\alpha_2$ of $U_2$ in $c_i$ is also replaced by random $\alpha_2^{*e}$ if $\alpha_2$ of $U_2$ has not changed $q_s - 1$ times.

- **game$_{n-1}q_1+1$: game$_{n-1}q_1+1$** is the same as game$_{n-1}q_1$ except that $ru_1$ of $U_1$ is replaced by random $ru_1^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_n$ of $U_n$ has not changed. $\alpha_n$ of $U_n$ in $c_i$ is also replaced by random $\alpha_n^{*e}$ if $\alpha_n$ of $U_n$ has not changed.

- **game$_{(n-1)}q_1+k$: game$_{(n-1)}q_1+k$** is the same as game$_{(n-1)}q_1+k-1$ except that $ru_2$ of $U_2$ is replaced by random $ru_2^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_n$ of $U_n$ has changed $k - 1$ times. $\alpha_n'$ of $U_n$ in $c_i$ is also replaced by random $\alpha_n^{*e}$ if $\alpha_n$ of $U_n$ has changed $k - 1$ times.

- **game$_{n-q_1}$: game$_{n-q_1}$** is the same as game$_{n-q_1-1}$ except that $ru_1$ of $U_1$ is replaced by random $ru_1^{*}$ in ciphertexts $c_i$ and $c_j$ if $\alpha_n$ of $U_n$ has changed $q_s - 1$ times. $\alpha_n$ of $U_n$ in $c_i$ is also replaced by random $\alpha_n^{*e}$ if $\alpha_n$ of $U_n$ has changed $q_s - 1$ times.

- **game$_{n-q_1+1}$: game$_{n-q_1+1}$** is the same as game$_{n-q_1}$ except that $rd_1$ of $D_1$ is replaced by random $rd_1^{*}$ in ciphertexts $d_i$ and $d_j$ if $\beta_1$ of $D_1$ has not changed. $\beta_1$ of $D_1$ in $d_i$ is also replaced by random $\beta_1^{*e}$ if $\beta_1$ of $D_1$ has not changed.

- **game$_{n-q_1+2}$: game$_{n-q_1+2}$** is the same as game$_{n-q_1+1}$ except that $rd_1$ of $D_1$ is replaced by random $rd_1^{*}$ in ciphertexts $d_i$ and $d_j$ if $\beta_1$ of $D_1$ has changed one time.
\( \beta'_1 \) of \( D_1 \) in \( d_j \) is also replaced by random \( \beta'^*_1 \) if \( \beta_1 \) of \( D_1 \) has changed once.

- \( \text{game}_{n,q+k}: \text{game}_{n,q+k} \) is the same as \( \text{game}_{n,q+k-1} \) except that \( r_d_1 \) of \( D_1 \) is replaced by random \( r_{d_1}' \) in ciphertexts \( d_j \) and \( d_j \) if \( \beta_1 \) of \( D_1 \) has changed \( k - 1 \) times. \( \beta'_1 \) of \( D_1 \) in \( d_j \) is also replaced by random \( \beta'^*_1 \) if \( \beta_1 \) of \( D_1 \) has changed \( k - 1 \) times.

- \( \text{game}_{(n+1),q}: \text{game}_{(n+1),q} \) is the same as \( \text{game}_{(n+1),q-1} \) except that \( r_d_1 \) of \( D_2 \) is replaced by random \( r_{d_1}' \) in ciphertexts \( d_i \) and \( d_j \) if \( \beta_2 \) of \( D_2 \) has not changed. \( \beta'_2 \) of \( D_2 \) in \( d_j \) is also replaced by random \( \beta'^*_2 \) if \( \beta_2 \) of \( D_2 \) has not changed.

- \( \text{game}_{(n+2),q}: \text{game}_{(n+2),q} \) is the same as \( \text{game}_{(n+2),q-1} \) except that \( r_d_2 \) of \( D_2 \) is replaced by random \( r_{d_2}' \) in ciphertexts \( d_i \) and \( d_j \) if \( \beta_2 \) of \( D_2 \) has not changed \( k - 1 \) times. \( \beta'_2 \) of \( D_2 \) in \( d_j \) is also replaced by random \( \beta'^*_2 \) if \( \beta_2 \) of \( D_2 \) has not changed.

- \( \text{game}_{(n+m-1),q_1}: \text{game}_{(n+m-1),q_1} \) is the same as \( \text{game}_{(n+m-1),q_1-1} \) except that \( r_d_m \) of \( D_m \) is replaced by random \( r_{d_m}' \) in ciphertexts \( d_i \) and \( d_j \) if \( \beta_m \) of \( D_m \) has not changed. \( \beta'_m \) of \( D_m \) in \( d_j \) is also replaced by random \( \beta'^*_m \) if \( \beta_m \) of \( D_m \) has not changed.

- \( \text{game}_{(n+m),q_1}: \text{game}_{(n+m),q_1} \) is the same as \( \text{game}_{(n+m),q_1-1} \) except that \( r_d_m \) of \( D_m \) is replaced by random \( r_{d_m}' \) in ciphertexts \( d_i \) and \( d_j \) if \( \beta_m \) of \( D_m \) has changed \( k - 1 \) times. \( \beta'_m \) of \( D_m \) in \( d_j \) is also replaced by random \( \beta'^*_m \) if \( \beta_m \) of \( D_m \) has changed \( k - 1 \) times.

- \( \text{game}_{(n+1),q} \) is given oracles \( \text{Mac} \) and \( \text{Vrfy} \) in the MAC scheme experiment, and uses the oracles to make and verify MACs that are supposed to be generated and verified with \( m_k \) or \( m_k^r \) in the randomly selected instance. A more concrete description of \( \mathcal{F} \) is as follows:

1. \( \mathcal{F} \) is given oracles \( \text{Mac} \) and \( \text{Vrfy} \) in the MAC scheme experiment, and uses the oracles to make and verify MACs that are supposed to be generated and verified with \( m_k \) or \( m_k^r \) in the randomly selected instance. A more concrete description of \( \mathcal{F} \) is as follows:

   - For Send: If this query is made for the \( k^* \)-th instance of the target party, the oracles \( \text{Mac} \) and \( \text{Vrfy} \) are used to generate and verify MACs that are supposed to be generated and verified with \( m_k \) or \( m_k^r \).

2. If each oracle query of \( A \), \( \mathcal{F} \) answers it the same as in \( \text{game}_0 \) except for the following:

   - If a forged MAC appears with respect to the target instance, \( \mathcal{F} \) outputs the forged MAC and message pair, and then quits. Otherwise, \( \mathcal{F} \) stops when \( A \) stops.

3. If a forged MAC appears in \( \text{game}_0 \), except that \( r_d_1 \) of \( D_1 \) is replaced by random \( r_{d_1}' \) in ciphertexts \( d_j \) and \( d_j \) if \( \beta_1 \) of \( D_1 \) has changed \( q_1 - 1 \) times. \( \beta'_1 \) of \( D_1 \) in \( d_j \) is also replaced by random \( \beta'^*_1 \) if \( \beta_1 \) of \( D_1 \) has changed \( q_1 - 1 \) times.

Proof of Lemma 1: If a forged MAC appears in \( \text{game}_0 \), we can construct an algorithm \( \mathcal{F} \) that breaks the SUF security of the underlying MAC scheme Mac.
1) For Send: If this query is made to the instance of $U_j$, and $\alpha_0$ of $U_j$ has changed $k^* - 1$ times, $D$ queries $(U_j || PU'_j || \alpha'_j || PD_j || ru'_j, U_j || PU'_j || \alpha'_j || PD_j || ru'_j)$ to encryption oracle $E_{sk}(LR(\cdot, \cdot), b)$ to get $c_j$. And $D$ queries $(PU'_j || ru'_j, PU'_j || ru'_j)$ to the encryption oracle to get $c_j$. 

If this query is not to the target instance, $D$ behaves the same way as in $\text{game}_j$.

2) For Test: This is the same as in $\text{game}_0$. Note that a coin $\sigma$ is flipped for the Test query.

3) If $A$ outputs $\sigma'$ and $\sigma = \sigma'$, $D$ outputs 1. The advantage of $D$ in the LoR-CPA game is bounded as follows:

$$\text{Adv}^{\text{LoR}}_D = \Pr[D = 1|b = 0] - \Pr[D = 1|b = 1]$$

$$\geq \Pr[\sigma = \sigma'|\text{game}_{\ell-1}] - \Pr[\sigma = \sigma'|\text{game}_j]$$

$$\geq \text{Adv}^{KE, \text{Case}2}_{A,\text{game}_{\ell-1}} - \text{Adv}^{KE, \text{Case}2}_{A,\text{game}_j}.$$ (17)

Similarly, with the advantage difference $\text{Adv}^{KE, \text{Case}2}_{A,\text{game}_{\ell-1}} - \text{Adv}^{KE, \text{Case}2}_{A,\text{game}_j}$ for $\ell \in [n \cdot q_s + 1, (n \cdot m) \cdot q_s]$, we can construct a protocol $D$, which breaks the LoR-CPA security of the underlying encryption scheme $SE$.

$D$ is given the encryption oracle $E_{sk}(LR(\cdot, \cdot), b)$ in the LoR-CPA experiment and uses the encryption oracle to make $d_I$ and $d_I$ for the target instances. We show a more concrete description of $D$ as follows, if $\ell = n \cdot q_s + (j^* - 1) \cdot q_s + k^*$, where $j^* \in [1, m]$ and $k^* \in [1, q_s]$:

1) For Send: If this query is made to the instance of $D_j$ and $\beta_j$ of $D_j$ has changed $k^* - 1$ times, $D$ queries $(PD_j || \beta_j || ru'_j || rd_j, PD_j || \beta_j || ru'_j || rd_j)$ to encryption oracle $E_{sk}(LR(\cdot, \cdot), b)$ to get $d_I$. And $D$ queries $(PD_j || \beta_j || ru'_j || rd_j, PD_j || ru'_j || rd_j)$ to the encryption oracle to get $d_I$. 

If this query is not to the target instance, $D$ behaves the same way as in $\text{game}_j$.

2) For Test: This is the same as in $\text{game}_0$. Note that a coin $\sigma$ is flipped for Test query.

3) If $A$ outputs $\sigma'$ and $\sigma = \sigma'$, $D$ outputs 1. Proof of Claim 2.2: Claim 2.2 is obvious from the fact that $A$ cannot get any information about $ru_i$ and $rd_j$ used to make a session key $sk_{i,j}$ for the Test session since $ru'_i$ and $rd_j$ are encrypted in ciphertexts for all instances instead of $ru_i$ and $rd_j$. Therefore,

$$\text{Adv}^{KE, \text{Case}2}_{A,\text{game}_{0+1+2}} = \Pr[\sigma = \sigma'] = \frac{1}{2}.\quad (18)$$

REFERENCES

[1] M. Gharibi, R. Boutaba, and S. L. Wielandter, “Internet of Drones,” IEEE Access, vol. 4, pp. 1148–1162, 2016.

[2] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, T. R. Gadekallu, and G. Srivastava, “SP2F: A secured privacy-preserving framework for smart agricultural unmanned aerial vehicles,” Comput. Netw., vol. 187, Mar. 2021. Art. no. 107819.

[3] R. Ch. G. Srivastava, T. Reddy Gadekallu, P. K. R. Maddikunta, and S. Bhattacharya, “Security and privacy of UAV data using blockchain technology,” J. Inf. Secur. Appl., vol. 55, Dec. 2020, Art. no. 102670.

[4] D. He, S. Chan, and M. Guizani, “Drone-assisted public safety networks: The security aspect,” IEEE Commun. Mag., vol. 55, no. 8, pp. 218–223, Aug. 2017.

[5] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. C. Rodrigues, “Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of Drones deployment,” IEEE Internet Things J., vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[6] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, “TCAALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment,” IEEE Trans. Veh. Technol., vol. 68, no. 7, pp. 6903–6916, Oct. 2019.

[7] Y. Zhang, D. He, L. Li, and B. Chen, “A lightweight authentication and key agreement scheme for Internet of Drones,” Comput. Commun., vol. 154, pp. 455–464, Oct. 2020.

[8] M. Wazid, A. K. Das, and J. H. Lee, “Authentication protocols for the internet of drones: Taxonomy, analysis and future directions,” J. Ambient Intell. Humanized Comput., pp. 1–10, 2018.

[9] Q. Xie, J. Zhao, and X. Yu, “Chaotic maps-based three-party password-authenticated key agreement scheme,” Nonlinear Dyn., vol. 74, no. 4, pp. 1021–1027, Aug. 2013.

[10] C.-C. Lee, C.-T. Li, S.-T. Chiu, and Y.-M. Lai, “A new three-party-authenticated key agreement scheme based on chaotic maps without password table,” Nonlinear Dyn., vol. 79, no. 4, pp. 2485–2495, Mar. 2015.

[11] C.-T. Li, C.-C. Lee, C.-Y. Weng, and S.-J. Chen, “A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-healthcare systems,” J. Med. Syst., vol. 40, no. 11, p. 233, Sep. 2016.

[12] M. Turkanović, B. Brumen, and M. Hölzl, “A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion,” Ad Hoc Netw., vol. 20, pp. 96–112, Sep. 2014.

[13] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölzl, “An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment,” Ad Hoc Netw., vol. 36, pp. 152–176, Jan. 2016.

[14] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, “Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks,” Comput. Netw., vol. 101, pp. 42–62, Jun. 2016.

[15] W. L. Tai, Y. F. Chang, and W. H. Li, “An IoT notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks,” J. Inf. Secur. Appl., vol. 34, pp. 133–141, Jun. 2017.

[16] Q. Jiang, S. Zeadally, J. Ma, and D. He, “Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks,” IEEE Access, vol. 5, pp. 3376–3392, 2017.

[17] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E. J. Yoon, and K. Y. Yoo, “Secure signature-based authenticated key establishment scheme for future IoT applications,” IEEE Access, vol. 5, pp. 3028–3043, 2017.

[18] M. Wazid, A. K. Das, V. Odeku, N. Kumar, M. Conti, and M. Jo, “Design of secure user authenticated key management protocol for generic IoT networks,” IEEE Internet Things J., vol. 5, no. 1, pp. 269–282, Feb. 2018.

[19] M. Wazid, A. K. Das, S. Shetty, J. J. Rodrigues, and Y. Park, “LDAKM-EloT: Lightweight device authentication and key management mechanism for edge-based IoT deployment,” Sensors, vol. 19, no. 24, p. 5539, Dec. 2019.

[20] A. Yazdinejad, R. M. Parizi, A. Dehghatanha, H. Karimipour, G. Srivastava, and M. Aledhari, “Enabling drones in the Internet of Things with decentralized blockchain-based security,” IEEE Internet Things J., vol. 8, no. 8, pp. 6406–6415, Apr. 2021.

[21] B. Sharma, G. Srivastava, and J. C.-W. Lin, “A bidirectional congestion control transport protocol for the Internet of Drones,” Comput. Commun., vol. 153, pp. 102–116, Mar. 2020.

[22] C. Buttner and S. A. Huss, “A novel anonymous authenticated key agreement protocol for vehicular ad hoc networks,” in Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), Feb. 2015, pp. 259–269.

[23] S. Shin and T. Kwon, “Two-factor authenticated key agreement supporting unlinkability in 5G-integrated wireless sensor networks,” IEEE Access, vol. 6, pp. 11229–11241, 2018.

[24] X. Li, J. Liu, M. S. Obaidat, P. Vijayakumar, Q. Jiang, and R. Amin, “An unlinkable authenticated key agreement with collusion resistant for VANETs,” IEEE Trans. Veh. Technol., vol. 70, no. 8, pp. 7992–8006, Aug. 2021.
[25] H. Khan, B. Dowling, and K. M. Martin, “Pragmatic authenticated key agreement for IEEE Std 802.15.6,” Int. J. Inf. Secur., pp. 1–19, 2021.
[26] J. Katz and Y. Lindell, Introduction to Modern Cryptography, 2nd ed. Boca Raton, FL, USA: CRC Press, 2014.
[27] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, “Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment,” IEEE Internet Things J., vol. 5, no. 6, pp. 4900–4913, Dec. 2018.
[28] V. Odelu, A. K. Das, and A. Goswami, “An efficient biometric-based privacy-preserving three-party authentication with key agreement protocol using smart cards,” Secur. Commun. Netw., vol. 8, no. 18, pp. 4136–4156, Dec. 2015.

JAE YEOL JEONG received the B.S. degree in mathematics and the M.S. degree in information security from Korea University, Seoul, Korea, in 2010 and 2013, respectively, where he is currently pursuing the Ph.D. degree with the School of Cybersecurity. His current research interests include cancelable iris templates and authenticated key exchange.

JIN WOOK BYUN (Member, IEEE) received the B.S. degree from the Department of Computer Science and the M.S. and Ph.D. degrees from the Graduate School of Information Security, Korea University, in 2001, 2003, and 2006, respectively. He has joined a Faculty Member at the Department of Information and Communication, Pyeongtaek University, in 2008. His research interests include the design of cryptographic protocol, authenticated key exchange, and database security.

IK RAE JEONG received the B.S. and M.S. degrees in computer science and the Ph.D. degree in information security from Korea University, Seoul, Korea, in 1998, 2000, and 2004, respectively. From June 2006 to February 2008, he was a Senior Engineer at the Electronics and Telecommunications Research Institute (ETRI), Korea. He is currently a Faculty Member with the School of Cybersecurity, Korea University. His current research interests include cryptography, theoretical computer science, blockchain, and biometrics.