IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2020, accepted February 7, 2020, date of publication February 17, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974381

An Efficient and Provably Secure Certificateless Key-Encapsulated Signcryption Scheme for Flying Ad-hoc Network

MUHAMMAD ASGHAR KHAN[1], INSAF ULLAH[2], SHIBLI NISAR[3], FAZAL NOOR[4], IJAZ MANSOOR QURESHI[5], FAHIM ULLAH KHANZADA[6], AND NOOR UL AMIN[2]

[1]Department of Electrical Engineering, Hamdard University, Islamabad 44000, Pakistan
[2]Department of Information Technology, Hazara University Mansehra, Dhodial 21120, Pakistan
[3]Department of Electrical Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[4]Faculty of Computer and Information Systems, Islamic University of Madinah, Medina 40411, Saudi Arabia
[5]Department of Electrical Engineering, Air University, Islamabad 44000, Pakistan
[6]Descon Engineering Limited, Lahore 54000, Pakistan

Corresponding author: Muhammad Asghar Khan (khayyam2302@gmail.com)

ABSTRACT A Flying Ad-hoc Network (FANET) consists of Unmanned Aerial Vehicles (UAVs) tasked to handle the communication jobs in a multi-hop ad-hoc fashion. Unlike its predecessors, i.e. Mobile Ad-hoc Networks (MANETs) and Vehicular Ad-hoc Networks (VANETs), a FANET promises uninterrupted connectivity, especially during events that are temporary and stipulate a massive audience reach. However, usually, the participating UAVs in a FANET environment are resource-constrained and are, therefore, prone to cyber-attacks. In order to resolve the issue and to enable a secure communication between the UAVs and the Base Station (BS), we propose a Certificateless Key-Encapsulated Signcryption (CL-KESC) scheme. The scheme is based on the concept of Certificateless Public Key Cryptography (CL-PKC). Since CL-PKC is immune to key escrow problems and thus one of the major drawbacks of the Identity-based Public Key Cryptography (ID-PKC) is addressed. Unfortunately, the existing construction models of CL-KESC rely on elliptic curve-based operations, which are computationally expensive for small UAVs. To counter the issue, in this paper, we present a new construction model of CL-KESC based on Hyperelliptic Curve Cryptography (HECC). HECC is an advanced version of the elliptic curve and is characterized by smaller parameter and key size. The key size stretches to a maximum of 80-bits, as opposed to the elliptic curve that demands a 160-bits key size. The proposed scheme proved to be superior, chiefly in terms of security and performance, as demonstrated by the results obtained from the security verification and by carrying out comparative analysis with the existing counterparts.

INDEX TERMS UAVs, FANET, certificateless signcryption, key-encapsulation, hyperelliptic curve, AVISPA.

I. INTRODUCTION

Flying Ad-hoc Network (FANET) is an emerging phenomenon that, smartly, helps realize a rapidly deployable, self-configurable and flexible communication network for data transmission between the Unmanned Aerial Vehicles (UAVs) and the Base Station (BS). Such a network is primarily composed of UAVs acting as communicating entities connected in a multi-hop ad-hoc networking fashion [1]. This feature eliminates the need for deployment of complex hardware in each of the UAVs. Furthermore, in case one of the UAV communication links breaks down, there is no disconnection with the BS since an ad-hoc network is already deployed between the UAVs [2]. Compared to its predecessors, i.e. Mobile Ad-hoc Networks (MANETs) and Vehicular Ad-hoc Networks (VANETs), the FANET can be used to ensure ubiquitous connectivity, particularly during mission-critical disaster-rescue operations [3]. However, most of the applications involved in the a FANET system are based on real-time scenarios. That is, the users are usually

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khandaker.

interested in retrieving real-time information from the UAVs connected to a specific region. It is possible only if the users are permitted to directly access real-time information from the UAVs inside a FANET environment rather than the BS. This results in a security security breach, which might deteriorate the effectiveness of an implemented solution in the FANET system.

Two major aspects of a security system, confidentiality and authentication, need to be addressed. In general, the answer to confidentiality and authentication lies in encryption and digital signature respectively. In case when both methods are required simultaneously, the sign-then-encrypt approach is utilized mostly. However, the stringent constraints associated with small UAVs, such as limited on-board energy and restricted computational capability, do not permit complex cryptographic operations [4]. Moreover, performing computationally intensive tasks on a UAV may result in slow response time which can, in turn, deteriorate the battery's lifetime and hence compromise the mission's success. Fortunately, such an impediment can be mitigated by employing an amalgamated scheme, named 'signcryption' that is aimed at offloading the demanding computational tasks from UAVs [5]. It is a public-key cryptosystem which performs the functions of digital signature and encryption simultaneously in a single logic step. Signcryption is, also, far more efficient and cost-effective than both the encryption and digital signature. Besides, due to its lower cost than the alternates involving signature followed by encryption, signcryption is appropriate for the resource-constrained environments such as Flying Ad-hoc Networks (FANET). However, when 'signcryption' scheme is applied directly to the messages carrying large chunks of data, its performance reduces significantly. Inspired by the concept of hybrid encryption, Dent [6] proposed Signcryption Key Encapsulation Mechanism (SC-KEM) in order to improve the practical use of signcryption. The SC-KEM construction approach is utilized step-wise as follows:

1) A random session key is encapsulated by a signcryption key encapsulation algorithm.

2) The data is encrypted by the very session key using a symmetric encryption algorithm.

3) Finally, both encapsulated the session key and the ciphertext are transmitted over the insecure channel.

In public key cryptosystems, two basic approaches, Public Key Infrastructure (PKI) and Identity-Based Cryptography (IBC), are used to authenticate the public keys. In the PKI environment, maintaining a trustworthy unforgeable link between identity of a participant and its public key is an essential prerequisite [7]. This further stipulates the need of a signature Certificate Authority (CA) that assigns the link a unique signature [8]. The CA bounds the public key as the identity of a participant with certificates. Some of the shortcomings typical of the PKI approach are certificate distribution, storage and manufacturing difficulties [9]. On the other hand, an identity-based cryptosystem [10] is used to reduce the cost of public key management; however, it suffers from the private key escrow problem [11,12] because the trusted third-party Private Key Generator [PKG] has firsthand information about the participants' private keys. In 2003, Al-Riyami and Paterson [13] proposed the certificateless cryptosystem to address the key escrow problem. In a certificateless cryptosystem a participant's private key is composed of two parts: the partial-private key and a secret value. The partial private key is generated by the trusted third-party Key Generation Center (KGC), while secret value is selected by the participant. Similarly, a participant's public key is also based on two parts i.e. the participant's identity information and the public key corresponding to the secret value. Since the public key does not require a certificate, the cost of public key management is significantly reduced. Furthermore, the KGC does not have any firsthand information about the participant's secret value; therefore, the scheme does not suffer from the key escrow problem.

The first certificateless signcryption scheme was proposed by Barbosa and Farshim [14]. The scheme incorporates salient features of certificateless encryption as well as the digital signature in a simultaneous manner. Lippold *et al*. [15] proposed direct construction for Certificateless Key Encapsulation (CL-KEM) that makes use of a certificateless public key encryption scheme of a pattern similar to that of KEM. Li *et al.* [16] proposed Certificateless Signcryption Tag - KEM (CLSC-TKEM), scheme that combines the ideas of SC-TKEM and CL-PKC. The primary advantage of the CLSC-TKEM is the elimination of costs incurred due to certificate management and key escrow problem. CLSC-TKEM, also, upholds the benefits offered by SC-TKEM. In CLSC-TKEM, the cryptographic operations, such as key encapsulation based on CL-PKC, can be performed only when an authentic user possesses the partial private key and the secret value. Moreover, the special structure of CL-PKC allows a user to perform the key decapsulation operation of CLSC-TKEM. Doing so does not necessitate verification of the sender's public key via a public key certificate. CLSC-TKEM is, thus, an efficient scheme since the key encapsulation and the digital signature functions are supported without the need of a certificate management infrastructure.

Normally, the security and efficiency of the aforementioned signcryption schemes are based on some computationally hard problems e.g., Rivest-Shamir-Adleman (RSA) cryptography, Bilinear pairing and elliptic curve cryptosystems. The RSA cryptography [17], [18] is based on a large factorization problem, which utilizes a 1024-bits long key, parameter, certificate and identity [19].This is not suitable for resource-constrained networks, or FANET in this case, due to the lack of onboard processing resources on small UAVs. Furthermore, bilinear pairing is 14.31 times worse than RSA [20] due to huge pairing and map-to-point function computation. In order to counter the deficiencies of RSA and bilinear pairing, a new type of cryptography called elliptic curve was introduced [21]. In the elliptic curve cryptography, the prominent characteristics such as parameter, public key, private key, identity and certificate are of smaller size.

Moreover, the security hardiness and efficiency of the scheme is based on 160-bit small keys, as opposed to bilinear pairing and RSA [22]. Despite, the 160-bit key is not suitable and affordable for resource-hungry devices. Thus, a new type, the generalization of elliptic curve, called hyperelliptic curve, was proposed [23]. The hyperelliptic curve features security same as that of the elliptic curve, bilinear pairing and RSA; it uses a 80-bit key, identity and certificate size [24], [25]. The hyperelliptic curve is deemed to be a better choice for energy-constrained devices.

A. AUTHOR'S MOTIVATIONS AND CONTRIBUTIONS

A comprehensive literature review of the existing key-encapsulated certificateless signcryption schemes was carried out. The schemes incur high computational and communication costs since they are based on hard problems, such as elliptic curve. Secondly, such schemes are not tested using AVISPA, Scyther and other security validation tools. Therefore, small devices that have limited computational power cannot handle them. Such restriction, for effective resolution, demands a solution that is characterized by low computational costs. The state-of-the-art key-encapsulated certificateless schemes need to be harnessed for coming forth with cryptographic solutions that pose no threat to the battery lifetimes of the resource-constrained UAVs.

Aspired from such motivations, a new scheme, named Certificateless Key-Encapsulated Signcryption (CL-KESC) scheme, has been proposed for FANET. The scheme utilizes the concept of hyperelliptic curve and is characterized by smaller key-size. Moreover, in an uncompromising manner, it does offer the security features promised by the elliptic curve model.

The research work undertaken is distinguished by following outstanding attributes:

- A secure and efficient scheme, namely Certificateless Key-Encapsulated Signcryption (CL-KESC) scheme, has been proposed for a FANET environment.
- The CL-KESC scheme makes use of the hyperelliptic curve and addresses the limitations posed by the resource-constrained elements.
- The proposed scheme is shown to be resistant against various attacks through informal security analysis as well as through the formal security verification using the AVISPA tool [43].
- The proposed scheme is also compared with major existing counterparts and it is shown that ours approach provides better efficiency in terms of computational cost as well as communication cost.

B. STRUCTURE OF THE PAPER

The rest of the paper is organized as follows. Section II contains a discussion about the related work. An overview of the underlying concepts and related definitions is provided in section III. The proposed Certificateless Key-Encapsulated Signcryption (CL-KESC) model is presented in section IV. CL-KESC scheme is provided in section V. Formal, provable and informal security analyses are carried out in section VI. Section VII compares the work with existing solutions and presents a comparative analysis. In section VIII, we provide an application scenario for the proposed scheme. Finally, section IX contains a critical reflection and the concluding thoughts.

II. RELATED WORK

The topic of security and privacy issues related to FANET have not, so far, received ample attention in the scientific literature. Therefore, the issues need to be investigated thoroughly. The primary security mechanisms for FANET emphasize on authenticity, confidentiality and integrity of the data by following the principles of cryptography. A well-designed data protection mechanism can significantly reduce the probability of the data getting compromised, irrespective of the devilish technique involved. In the literature, we have come across some studies dedicated to investigating the data protection issues for UAV networks.

A certificate-based encryption communication scheme for an MBN-UAV network is proposed by Kong *et al.* [26]. The scheme uses a negotiated session key in order to support and authenticate the identity of end devices. Only then, the message encrypted with a symmetric key is transmitted. However, as a drawback, since the scheme only entertains a secure end-to-end communication, it fails to support the broadcast of encrypted messages. Therefore, in order to establish multiple individual session keys, the involved devices are required to dedicate ample computational resources.

In a secure communication scheme proposed by He *et al* [27], the requirement of an online centralized authority is waived off. The UAVs, themselves, manage the area and the authorized devices can obtain a broadcast key. The scheme is characterized by employing a hierarchical identity-based broadcast encryption and pseudonym mechanism in which the devices can, anonymously, perform broadcasting of the encrypted messages and decryption of the legal ciphertext. The work done seconds the notion that the very scheme, satisfactorily, addresses four important security concerns: confidentiality, authentication, partial privacy-preservation and resistance to Denial of Service (DoS) attacks. However, it inherits a limitation in the registration phase. That is, the concern of finding a hash value's preimage still persists.

Won *et al.* [28], [29] proposed a suite of cryptographic protocols for drones and smart objects. The protocols deal with three communication scenarios, viz., one to- one, one-to-many and many-to-one. In the first scenario, i.e. 'one-to-one', the efficient encapsulation mechanism, a certificateless signcryption tag key, backs the authenticated key agreement in addition to providing non-repudiation and user revocation. The 'one-to-many' scenario involves a certificateless multi-recipient encryption scheme, which allows a UAV to transmit privacy-intensive data to multiple smart objects. Lastly, UAVs are able to collect data from multiple smart

objects in the 'many-to-one' communication scenario. The protocol, however, finds it difficult to transmit a multitude of encrypted messages and, at the same time, assures the privacy of end devices. Such novel cryptographic mechanisms are efficient and secure. However, they are supposed to be used in group communication where nodes have equal computational capability.

Semal *et al*. [30] proposed a Certificateless Group Authenticated Key Agreement (CL-GAKA) scheme to address the topic of secure communication between untrusting parties. The scheme claimed to have facilitated the provision of data integrity, confidentiality and authenticity for an environment that involves UAVs communicating with each other. The solution is, however, limited to the cases where the number of UAVs remains unchanged and there is a little probability of new entrants/leavers. Besides, the work also fails to address the problem of misbehavior from the authenticated elements.

A novel approach to mitigate the broadcast storm problem during the dissemination of interest packets is proposed by Barka *et al.* [31] The approach is based on a trust-aware monitoring communication architecture for flying named data networking. It makes use of the inter-UAV communication trust for checking the data authenticity on a particular UAV without disturbing the desired level of security. However, data privacy and caching policies are not taken into consideration in the proposed scheme.

To resist against the physical capturing of drones with minimum exposure of confidential data, Bae and Kim [32] proposed a saveless-based key management and delegation system for a multi-drone system. Nevertheless, the proposed scheme is not compatible with devices having limited on-board energy, or, in this case, UAVs, because the process of key renewal suffers significantly due to scarce available energy.

Seo *et al.* [33] proposed a pairing-free approach for the drone-based surveillance applications. The approach suffers from the problem of user revocation when a physical attack occurs. Therefore, the intruder(s) can access the current as well as the future information of the drones.

In order to overcome the problem of forward secrecy in drones, Liu *et al.* [34] proposed two constructions schemes. The schemes are shown to achieve better performance. However, this approach is based on elliptic curve and, therefore, suffers from high computational costs. Moreover, the proposed scheme is not validated through formal security analysis.

Zhou *et al.* [20] proposed a certificateless key-insulated generalized signcryption scheme without bilinear pairings to resolve the issues of private key exposure. The simulation results provide information about the efficiency analysis of the proposed scheme, and such scheme, based on cloud systems, is considered to be more suitable for the user communication. Similarly, the proposed scheme is also based on the concept of elliptic curve, which is computationally very expensive for small UAVs. Besides, the scheme is not validated through AVISPA, Scyther or any other formal security verification tool.

In 2018, Reddy et al. [35], with the aim to improve computational performance and communication efficiency, presented a pairing-free key insulated signature scheme in identity-based setting. Later, in 2019, Xiong *et al.* [36] also proposed a pairing-free scheme and a provably secure Certificateless Parallel Key-Insulated Signature (CL-PKIS) scheme for securing the communication in an Industrial Internet of Things (IIoT) environment. However, these approaches are also based on the concept of elliptic curve and, therefore, suffer from high computational cost. Moreover, the proposed schemes are not validated through formal security analysis.

III. PRELIMINARIES

In this section, a brief introduction to some major basic foundational concepts, along with the formal definitions, is presented.

A. HYPER ELLIPTIC CURVE CRYPTOSYSTEMS (HECC)

Hyperelliptic curves are a special class of algebraic curves that can be viewed as generalizations of the Elliptic Curve Cryptosystems (ECC) [37]. A hyperelliptic curve [38] is defined over curves whose genus is greater than 1 as shown in Fig.1. The curve with genus of value 1 is, commonly, known as elliptic curve. The vairable 'g' is the genus of curve over F_q, the set of finite fields of order 'q'. For the group order of the field F_q, for genus one, we will need a field F_q with following value $|F_q|$ g. $\log_2$ q $\approx 2^{160}$. For a curve with genus two, we will need a field F_q with following value $|F_q| \approx 2^{80}$. Similarly, for curves with genus three, 54 bits long operands are needed [39].

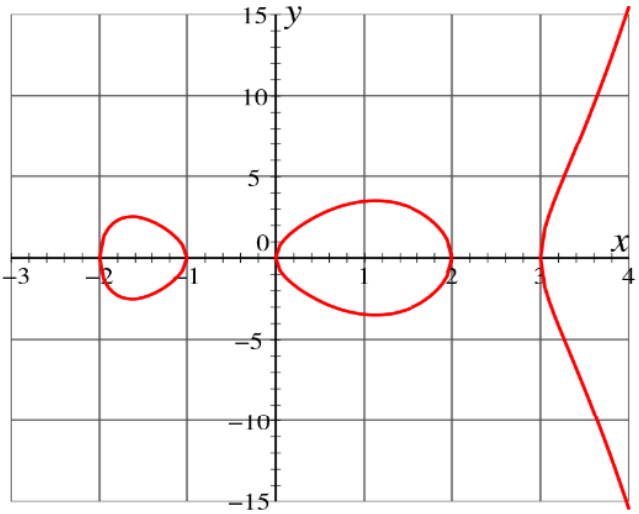


FIGURE 1. Hyper elliptic curve of genus two i.e. g=2 [38].

Let F be a finite field, and let $\bar{F}$ be the algebraic closure of F. A hyperelliptic curve C of genus $g > 1$ over F is a set of solutions $(x, y) \in F \times F$ to the equation of the curve C: $y^2 + h(x)y = f(x)$. Such a curve is said to be non-singular if

there are no pairs of $(x, y) \in \bar{F} \times \bar{F}$ which, at the same time, satisfy the equation of the curve C and the following partial differential equations: $2y + h(x) = 0$ and $h'(x)y - f'(x) = 0$. The polynomial $h(x) \in F[u]$ is of degree g and $f(x) \in F[u]$ is a monic polynomial of degree $2g + 1$. For odd characteristic it suffices to let $h(x) = 0$ and to have f(x) as a square free entity.

B. COMPLEXITY ASSUMPTIONS

While conducting the analysis, we have made following assumptions:

- Fq is a finite field with the order q, where $q \approx 2^{80}$
- D is a divisor of the hyper elliptic curve (HEC), which is the finite sum of points as:

$$D = \sum_{P_i \in HEC} m_i p_i, \quad m_i \in \text{Fq}.$$

1) HYPER ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM (HECDL) ASSUMPTION

For HECDLP, we make following suppositions:

a) ϑ belongs to {1,2,3, ..., q−1}
b) Probability computation ϑ from $\mathfrak{N} = \vartheta \cdot D$ is negligible.

2) HYPER ELLIPTIC CURVE COMPUTATIONAL DIFFIE-HELLMAN (HECCDH) ASSUMPTION

For HECCDH, we make following suppositions:

1) ϑ and Q belongs to {1,2,3, ..., q−1}
2) Probability computation ϑ and Q from $\acute{\Gamma} = \vartheta.\mathrm{Q}.D$ is negligible.

C. DEFINITIONS

A Certificateless Key-Encapsulated Signcryption (CL-KESC) scheme contains a tuple of seven algorithms: SetUp, Contestant Secret Value Generation (CSVG), Partial Private Key Generation (PPKG), Contestant Full Key Generation (CFKG), Symmetric Key Generation (SKG), Certificateless Encapsulation (CLEN) and Certificateless De-Encapsulation (CLDEN). The notations used in the proposed scheme are illustrated in Table 1.

1) SETUP

The Key Generation Center (KGC) runs this algorithm and it involves taking a security parameter $\acute{d}$ as input and generates a master secret key $\mathcal{w}$, a public key u and a set of public parameters ($\mathcal{D}$, H_a, H_b, H_c, $|F_q| \approx 2^{80}$, F, $\bar{F}$, C, g). The master secret key $\mathcal{w}$ is kept secret; whereas, the set of public parameters and master public key u is made available publicly by the KGC.

2) CONTESTANT SECRET VALUE GENERATION (CSVG)

This algorithm is run by each contestant and it involves considering the information of public parameters ($\mathcal{D}$, H_a, H_b, $|F_q| \approx 2^{80}$, F, $\bar{F}$, C, g) and the contestants' identity $\mathrm{ID}_{\mathcal{C}}$ as inputs in order to select the secret value $\alpha_{\mathcal{C}}$.

TABLE 1. Notations used.

Notation	Description
β_{en}, γ_{en}	Partial private key pair of encapsulations
β_{dn}, γ_{dn}	Partial private key pair of de-encapsulations
Υ_{en}, α_{en}	Full private key pair of encapsulations
Υ_{dn}, α_{dn}	Full private key pair of encapsulations
β_{en}, δ_{en}	Public key pair of encapsulations
β_{dn}, δ_{dn}	Public key pair of de-encapsulations
$\mathcal{C}$	Ciphertext
$\mathcal{m}$	Plaintext
$\mathcal{D}$	Divisor of hyperelliptic curve
$\mathcal{w}$	Master secret key
$\mathcal{t}$	Arbitrary tag
IDen	Identity of the encapsulation
IDdn	Identity of the de encapsulation
F	Finite field of hyperelliptic curve
$\bar{\mathrm{F}}$	Algebraic closure of F.
ψ	Encapsulated tuple
H_a, H_b	Hash Functions (SHA-512)
K	Secret key
$\oplus$	Means encryption and decryption
$\mathcal{S}_{\mathcal{k}}$	Shared secret key

3) PARTIAL PRIVATE KEY GENERATION (PPKG)

The KGC generates a partial private key pair ($\beta_{\mathcal{C}}$, $\gamma_{\mathcal{C}}$) for each contestant in response to following inputs: the identity $\mathrm{ID}_{\mathcal{C}}$ of each contestant and a set of public parameters. The partial private key pair is then transmitted to contestants with unsafe network.

4) CONTESTANT FULL KEY GENERATION (CFKG)

This algorithm is also run by each contestant. It involves taking a partial private key pair ($\beta_{\mathcal{C}}$, $\gamma_{\mathcal{C}}$), secret value $\alpha_{\mathcal{C}}$, a set of public parameters and an identity $\mathrm{ID}_{\mathcal{C}}$ of each contestant. Then, the CFKG algorithm produces their full private ($\Upsilon_{\mathcal{C}}$, $\alpha_{\mathcal{C}}$) and public key ($\beta_{\mathcal{C}}$, $\delta_{\mathcal{C}}$) pairs.

5) SYMMETRIC KEY GENERATION (SKG)

This algorithm is run by the sender to obtain symmetric key K and internal state information that is not known to the receiver. It basically takes a de-encapsulation identity ID_{rp}, a set of public parameters and a de-encapsulation public key δ_{dn} as inputs.

6) CERTIFICATELESS ENCAPSULATION (CLEN)

This algorithm is executed by the sender and it takes the following information as inputs: sender and recipient identity (ID_{en}, ID_{dn}); a set of public parameters; recipient public key δ_{dn}; a fresh nonce Non; arbitrary tag t; sender private key pair (Υ_{en}, α_{en}); and secret key K. As an outcome, it produces the encapsulated tuple $\psi = (\mathcal{C}, r, \mathcal{S}, \Omega)$ for recipient.

7) CERTIFICATELESS DE-ENCAPSULATION (CLDEN)

This algorithm is executed by the receiver and it takes the following information as inputs: the encapsulated tuple $\psi = (\mathcal{C}, r, \mathcal{S}, \Omega)$; a sender and recipient identity (ID_{en}, ID_{dn}); a set of public parameters; encapsulated public key δ_{en}; a fresh nonce Non; de-encapsulated private key pair (Υ_{dn}, α_{dn}) and public key δ_{dn}. Then, it proceeds with performing the de-encapsulation and verification process.

IV. PROVABLE SECURITY MODEL FOR CL-KESC SCHEME

The two types of adverseries are classified as Type 1, represented as TA_1, and type 2, represented as TA_2. The adversary TA_1 can maliciously replace the public key of the contestant. However, it has no access to the master secret key. The TA2 can act otherwise. That is, TA2 can access the master key and generate the partial private key. However, it cannot replace the public key. The adversary's queries and the subsequent responses are classified as following seven oracles:

A. CREATE-CONTESTENT-ORACLE

This oracle involves taking the contestant ID as an input. Then, the input is checked in the list. If it is already available in list then PK_c of the corresponding ID is retrived. In case of its unavailability, the corresponding values are assigned as follows: Public key $PK_c = (\beta_{\mathcal{C}}, \delta_{\mathcal{C}})$; and Private key $PR_c = (\Upsilon_{\mathcal{C}}, \alpha_{\mathcal{C}})$. Further, the set ($PK_c$, PR_c) is appended to the list and the value of PK_c is returned.

B. REVEAL-CONTESTENT-PARTIAL-PRIVATE-KEY-ORACLE

In this case, the user ID is searched in the list. If the ID is available, the value of $\gamma_{\mathcal{C}}$ is retrieved. Else, the null symbol $\perp$ is retrieved.

C. REVEAL-CONTESTENT-SECRETE-VALUE-ORACLE

Here the user ID is searched from the list. If it is available the variable $\alpha_{\mathcal{C}}$ is retrieved. Else the null symbol $\perp$ is returned.

D. REPLACE-PUBLIC-KEY-ORACLE

This oracle selects a random number instead of the contestant public key. After receiving the target ID, it replaces the contestant public key in a list with such randomely selected number.

E. SYMMETRIC-KEY-GENERATION AND ENCAPSULATION-ORACLE

This oracle is used to obtain following information:

a.) Symmetric key K, ψ
b.) Internal state information that is not known to the receiver
c.) Sender and recipient identity, (id_{en}, id_{dn})
d.) A set of public parameters
e.) Recipient public key, δ_{dn}
f.) A fresh nonce, Non
g.) Arbitrary tag, t
h.) Sender private key, PR_{en}
i.) Secret key, K

As an outcome, it produces the encapsulated text K and ψ.

F. CERTIFICATELESS DE-ENCAPSULATION -ORACLE

This oracle takes the following information as inputs: the encapsulated tuple ψ; a sender and recipient identity (ID_{en}, ID_{dn}); a set of public parameters; encapsulated public key δ_{en}; a fresh nonce Non; de-encapsulated private key PR_{dn}; and public key δ_{dn}. Then, it proceeds with performing the de-encapsulation and verification process.

1) SECURITY NOTIONS

This sub section is dedicated to discuss two main security requirements: confidentiality and unforgeability.

a: CONFIDENTIALITY

Defnition 1: A Certificateless Key Encapsulation scheme can be secured from the ciphertext indistinguishability adaptive ciphertext attacks (IND-CL-KESC-CCA2) if there is no intruder who can win the games 1 and 2 by making use of the polynomial bounded time.

Game 1: This game is played for the purpose of upholding confdentiality and is actualy based on the ciphertext indistinguishability adaptive ciphertext attacks (in contrast to TA_1).

Initialization: Given the security parameter k, the setup step is processed by challenger to produce master secret key, master public key and a set of public parameters. The master secret key is kept secret,;hereas, the set that includes public parameters and master public key is made publicly available.

Phase I: TA_1 can probe for the aforementioned seven oracles adaptively.

Challenge: In this section, the TA_1 submits the sender and recipient identities (ID_{en}, ID_{dn}), alongwith two different same size messages $\mathcal{M}_0$ and $\mathcal{M}_1$. Then, the challenger picks a random bit $\partial \in \{0,1\}$ and applies the Certificateless Key Encapsulation Algorithm on $\mathcal{M}_\partial$ for producing a Certificateless Key Encapsulation text ψ^* for TA_1.

Phase II: Just as in Phase 1, the TA_1 probes for the same oracle query adaptively, ignoring the Reveal-Contestent-Partial-Private-Key-Oracle and

Reveal-Contestent-Secrete-Value-Oracle. It futher excepts Certificateless De-encapsulation-Oracle with (ψ^*,ID_{en}, ID_{dn}) unless the public key of ID_{en}, ID_{dn} has been altered.

Output: Finally, TA_1 results in $\mathcal{M}_{\partial}{}'$ as the answer of Certificateless key Encapsulation text ψ^*. In case $\partial' = \partial$ then TA_1 emerges as winner of the game 1.

Game 2: This game is played for the purpose of maintaining confdentiality and it is actualy based on the ciphertext indistinguishability adaptive ciphertext attacks in contrast to TA_2.

Initialization: Given the security parameter k, the setup step is processed to produce master secret key, master public key and the set of public parameters. The master secret key is kept secret; whereas, the set that contains public parameters and master public key is made publicly available.

Phase I: TA_2 can probe for the aforementioned seven oracle adaptively.

Challenge: In this section, the TA_2 transmits sender and recipient identities (ID_{en}, ID_{dn}), along with two different same-sized messages $\mathcal{M}_0$ and $\mathcal{M}_1$. Then, the challenger picks a random bit $\partial \epsilon$ {0,1} and applies Certificateless key Encapsulation algorithm on $\mathcal{M}_{\partial}$ for producing a Certificateless Key Encapsulation text ψ^* for TA_2.

Phase II: Just as in Phase 1, the TA_2 probes for the same oracle query adaptively, ignoring the Certificateless De-encapsulation-Oracle as in the earlier case.

Output: Finally, TA_2 results in $\mathcal{M}_{\partial}{}'$ as the answer of Certificateless key Encapsulation text ψ^*. If $\partial' = \partial$ then TA_2 wins the game 2.

2) UNFORGEABILITY

Defnition 2: A Certificateless Key Encapsulation scheme can be secured from EUF-CMA, if there is no intruder, who can win the games 3 and 4 utilizing some polynomial bounded time.

Game 3: This game is played for testing the property of unforgeability and it is actualy based on the EUF-CMA.

Initialization: Given the security parameter k, the setup step is processed to produce master secret key, the master public key and a set of public parameters. The master secret key is kept secret; whereas, the set of public parameters and the master public key is made available publicly.

***Queries*:** In this section TF_1 probes for the following oracle queries adaptively: Create-Contestent-Oracle, Reveal-Contestent-Partial-Private-Key-Oracle, Reveal-Contestent-Secrete-Value-Oracle, Replace Public-Key-Oracle, Symmetric-Key-Oracle, Certificateless Encapsulation–Oracle and Certificateless De-encapsulation –Oracle.

Forgery: At the end, TF_1 computes ψ^* for m, ID^*_{en}, ID^*_{dn} utilizing the Certificateless Encapsulation–Oracle. If ψ^* is genuine then TF_1 prospers in game 3. However, the TF_1 is restricted from directing the queries such as Reveal-Contestent-Partial-Private-Key-Oracle and Reveal Contestent-Secrete-Value-Oracle.

Game 4: This game is played for demonstrating the unforgeability property and is based on EUF-CMA.

Initialization: Given the security parameter k, the setup step is processed to produce master secret key, master public key and a set of public parameters. Along with the master secret key, the challenger sends the TF_2 public parameter(s) and a master public key.

Queries: In this section TF_2 probes for the following oracle queries adaptively:

Create-Contestent-Oracle, Reveal-Contestent-Secrete-Value-Oracle, Symmetric-Key –Oracle, Certificateless Encapsulation–Oracle and Certificateless De-encapsulation –Oracle.

Forgery: In the end, TF_2 computes ψ^* for m, ID^*_{en} and ID^*_{dn} utilizing the Certificateless Encapsulation–Oracle. If ψ^* is genuine then TF_1 stands victorious in game 3. However, here, it is obligatory for the TF_2 to not have lodged the appeal of Contestent-Secrete-Value-Oracle throughout the game 4.

V. PROPOSED CERTIFICATELESS KEY-ENCAPSULATION SIGNCRYPTION SCHEME

A. NETWORK MODEL

An attempt to deploy the proposed scheme must be followed by due consideration to the following assumptions:

1)Each of the UAVs and the GS are connected with Wi-Fi-based ad-hoc networks.

2) The GS presumes the role of administrator and commands the course of UAVs.

3) Then, the UAVs adopt the flight path while flying, and execute the command(s) are issued by the GS.

4) The UAVs are located in the proximity of GS before the flight; therefore, the initial key generation and distribution are secured.

Depending on the application, FANET can be deployed in different settings. The UAVs can, then, be equipped with all the necessary gadgets such as cameras, IMU, sensors, GPS unit and data storage devices, including on-board processing units, flight controller, and short-range radio transceivers (i.e. Wi-Fi). The backbone UAV is mounted with dual band(s) of Wi-Fi, i.e. 802.11b and 802.11n, supposedly operating on 2.4 GHz and 5 GHz frequencies respectively. There are cogent reasons for choosing these technology standards. For instance, they operate in the unlicensed spectrum and they offer reasonable data rate and coverage along with no strict LOS. In addition, they can be easily integrated with small-sized UAV. Furthermore, 802.11n is best suited for air-to-air link, whereas 802.11b is suitable for air-to-ground link [3]. The GS is connected with the backbone UAV via IEEE 802.11b (Wi-Fi), whereas the member UAVs are connected with each other and the backbone UAV through 802.11n, thus paving way for an ad-hoc network architecture [41] as shown in Fig.2.

As depicted in Fig.3, our main system components are the UAV, KGC and GS. UAVs are the main communicating entities in the FANET system. To identify the target, multiple small UAVs are integrated as a team to collect sensing data using specialized onboard sensors.

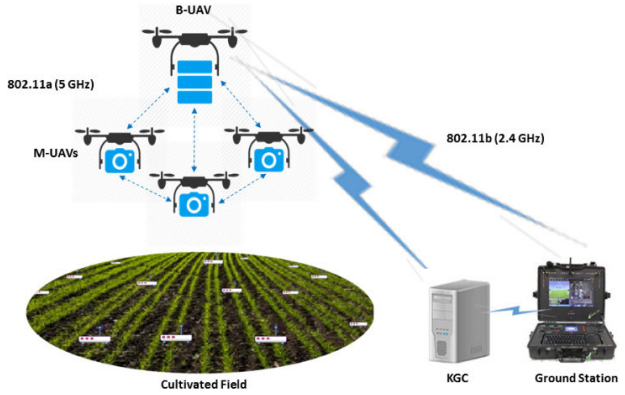


FIGURE 2. Sample architecture of FANET system.

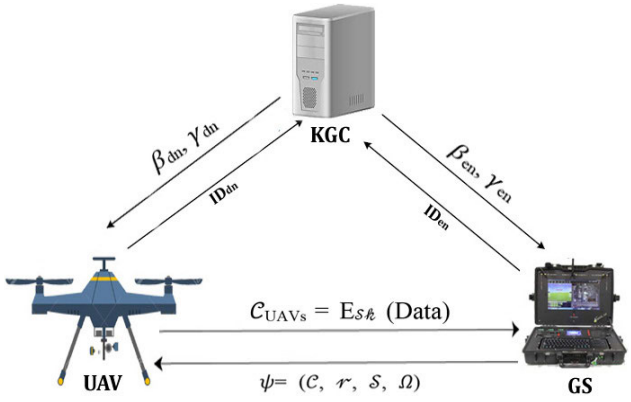


FIGURE 3. Main Components of FANET system.

Key Generation Center (KGC) is responsible for generating all the public parameters, partial private, master secret key and public key authentication. The users are likely required to share the data collected by multiple UAVs to obtain the information required to locate and contain the source. In the scheme, access control can only be controlled by the KGC (i.e., acting as a trusted party). The GS is the administrator that issues various commands and sets the course of the UAVs.

B. THREAT MODEL

The widely recognized Dolev-Yao (DY) threat model [42] is used in the proposed scheme. According to the DY model, an insecure public channel (open channel) is used for communication between any two parties; and the end-point entities have an untrustworthy nature. Therefore, the system is prone to eavesdropping of exchanged messages and deletion/modification attempts by the attacker. Besides, since the UAVs may roam around in unattended hostile areas, there does exist the probability of them getting physically captured. This may lead to leakage of precious data from the UAV's memory. The KGC, on the other hand, is a fully trusted entity.

C. CONSTRUCTION OF CERTIFICATELESS KEY-ENCAPSULATION SIGNCRYPTION SCHEME

Eight algorithms are considered for constructing the proposed scheme [55]. Each of them is explained as follows:

1) SETUP

Step 1: Given a security parameter $\acute{d}$, the KGC selects a master secret key as: {1,2,3, ... , q−1}.

Step 2: The KGC calculates a master public key u using the relation $u = w.\mathcal{D}$ and selects the set of public parameters $(\mathcal{D}, H_a, H_b, |F_q| \approx 2^{80}, F, \bar{F}, C, g)$.

Step 3: The KGC keeps a master secret key w and publishes the set of public parameters $(\mathcal{D}, H_a, H_b, H_c, |F_q| \approx 2^{80}, F, \bar{F}, C, g)$ and master public key u.

2) CONTESTANT SECRET VALUE GENERATION (CSVG)

Given the set of public parameters $(\mathcal{D}, H_a, H_b, |F_q| \approx 2^{80}, F, \bar{F}, C, g)$, the contestant with identity $ID_{\mathcal{C}}$ randomly selects a secret value $\alpha_{\mathcal{C}}$ from the set {1,2, 3 . . . . . , q−1} and computes the public value as $\delta_{\mathcal{C}} = \alpha_{\mathcal{C}}.\mathcal{D}$.

3) CONTESTANT PARTIAL PRIVATE KEY GENERATION (CPPKG)

Given each of contestants' identity $ID_{\mathcal{C}}$ and a set of public parameters $(\mathcal{D}, H_a, H_b, |F_q| \approx 2^{80}, F, \bar{F}, C, g)$, the KGC proceeds as follows:

a. It selects a random number $\Upsilon_{\mathcal{C}}$ from the following set: {1,2,3, ... , q−1}
b. It computes $\beta_{\mathcal{C}}$ and $\gamma_{\mathcal{C}}$ using following equations:

$$\beta_{\mathcal{C}} = \Upsilon_{\mathcal{C}}.\mathcal{D}$$
$$\gamma_{\mathcal{C}} = \Upsilon_{\mathcal{C}} + w.H_a(ID_{\mathcal{C}} \| \beta_{\mathcal{C}} \| \delta_{\mathcal{C}})$$

c. It finally transmits a pair $(\beta_{\mathcal{C}}, \gamma_{\mathcal{C}})$ to each contestant with identity $ID_{\mathcal{C}}$ through secure network.
d. Each contestant with identity $ID_{\mathcal{C}}$ accepts the pair $(\beta_{\mathcal{C}}, \gamma_{\mathcal{C}})$ on the condition that $\gamma_{\mathcal{C}}.\mathcal{D} = \beta_{\mathcal{C}} + H_a(ID_{\mathcal{C}} \| \beta_{\mathcal{C}} \| \delta_{\mathcal{C}}).u$.

4) CONTESTANT FULL PRIVATE KEY GENERATION (CFPKG)

The CFPKG algorithm is also applied by each of the contestants. It involves taking a contestant's partial private key pair $(\beta_{\mathcal{C}}, \gamma_{\mathcal{C}})$, secret value $\alpha_{\mathcal{C}}$ and the identity $ID_{\mathcal{C}}$. As an outcome, a pair of private key is produced as follows: $PR_c = (\gamma_{\mathcal{C}}, \alpha_{\mathcal{C}})$.

5) CONTESTANT FULL PUBLIC KEY GENERATION (CFPBKG)

The CFPBKG algorithm is also applied by each of the contestants. It involves taking a contestant's partial private key pair $(\beta_{\mathcal{C}}, \gamma_{\mathcal{C}})$, public value $\delta_{\mathcal{C}}$, and the identity $ID_{\mathcal{C}}$. As an outcome, the following pair is produced: full public key $PK_c = (\beta_{\mathcal{C}}, \delta_{\mathcal{C}})$.

6) SYMMETRIC KEY GENERATION (SKG)

Given de-encapsulation identity ID_{dn}, public key δ_{dn}, a set of public parameters ($\mathcal{D}$, H_a, H_b, $|F_q| \approx 2^{80}$, F, $\bar{F}$, C, g) and master public key u, the sender proceeds as follows:

a. It picks a random number μ, where $\mu \in \{1,2,3 \ldots\ldots., q-1\}$
b. It calculates the value of Ω using the relation: $\Omega = \mu.\mathcal{D}$
c. It computes $\mathcal{R}$ as follows: $\mathcal{R} = (\mu.H_a(\beta_{dn}\|ID_{dn}\|\delta_{dn}).u + \beta_{dn} + \delta_{dn})$
d. It computes $\mathcal{F}$ using the relation: $\mathcal{F} = \mu.\delta_{dn}$

7) CERTIFICATELESS ENCAPSULATION (CLEN)

The algorithm considers the following information:

- Encapsulation and de-encapsulation identity (ID_{en}, ID_{dn})
- A set of public parameters ($\mathcal{D}$, H_a, H_b, H_c, $|F_q| \approx 2^{80}$, F, $\bar{F}$, C, g)
- De-encapsulation public key δ_{dn}
- A fresh nonce Non
- Arbitrary tag t
- Encapsulation private key pair (γ_{en}, α_{en})
- The variables Ω, $\mathcal{R}$, and $\mathcal{F}$

Then, as a next step, the algorithm proceeds as follows:

- It selects a shared secret key $\mathcal{SK}$ from the Advanced Encryption Standard (AES)
- It sets m as $m = (\mathcal{SK}\|m\|t\|$ Non $\|$ $ID_{en})$
- It computes $\mathcal{C}$ using the equation: $\mathcal{C} = H_b(\Omega\|\mathcal{R}\|\mathcal{F}\|$ $ID_{dn}) \oplus m$
- It calculates the signature $\mathcal{S}$ using the equation: $\mathcal{S} = \gamma_{en} + \alpha_{en}.H_c(\Omega\|\mathcal{R}\|m\|ID_{en}\|\delta_{en}) + \mu.H_c(\Omega\|\mathcal{R}\|m\|ID_{en}\|\beta_{en})$
- Finally, it ends up producing the encapsulated tuple $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$ for recipient.

8) CERTIFICATELESS DE-ENCAPSULATION (CLDEN)

The CLDEN algorithm considers the encapsulated tuple $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$. Here, it is worth mentioning that prior to accepting the tuple $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$, it takes the following parameters as input in order to verify the signature and proceed with decrypting the cipher text.

- Set of public parameters ($\mathcal{D}$, H_a, H_b, $|F_q| \approx 2^{80}$, F, $\bar{F}$, C, g)
- Encapsulation and de-encapsulation identities (ID_{en}, ID_{dn})
- Encapsulation of public key δ_{en}
- De-encapsulation of private key pair (Υ_{dn}, α_{dn})
- Public key δ_{dn}

Then, the algorithm performs the following tasks:

- It computes $\lambda = \alpha_{dn}.\Omega$
- It recovers the secret key $m' = \mathcal{C} \oplus H_b(\Omega\|\mathcal{R}'\|\lambda\|$ $ID_{dn})$, where, $\mathcal{R}' = (\Omega.(\alpha_{dn} + \gamma_{dn}))$
- It checks for the equation:

$$\mathcal{S}.\mathcal{D} = (\beta_{en} + H_a(ID_{en}\|\beta_{en}\|\delta_{en}).u + \delta_{en}.H_c(\Omega\|\mathcal{R}'\|m'\|ID_{en}\|\delta_{en}) + \Omega.H_c(\Omega\|\mathcal{R}'\|m'\|ID_{en}\|\beta_{en})$$

D. CORRECTNESS

The de-encapsulation can recover the cipher text as:

$$\begin{aligned}
m' &= \mathcal{C} \oplus H_b(\Omega||\mathcal{R}'||\lambda||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||\mathcal{R}'||\lambda||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\Omega.(\alpha_{dn} + \gamma_{dn}))||\lambda||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\Omega.(\alpha_{dn} + \gamma_{dn}))||\alpha_{dn}.\Omega||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\mu.\mathcal{D}.(\alpha_{dn} + \gamma_{dn}))||\alpha_{dn}.\mu.\mathcal{D}||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\mu.(\alpha_{dn}.\mathcal{D} + \gamma_{dn}.\mathcal{D}))||\alpha_{dn}.\mu.\mathcal{D}||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\mu.(\delta_{dn} \\
&\quad + \beta_{dn} + H_a(ID_{dn}||\beta_{dn}||\delta_{dn}).u))||\mu.\delta_{dn}||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\mathcal{R}||\mu.\delta_{dn}||ID_{dn}) \\
&= \mathcal{C} \oplus H_b(\Omega||(\mathcal{R}||\mathcal{F}||ID_{dn}) \\
&= H_b(\Omega||\mathcal{R}||\mathcal{F}||ID_{dn}) \quad \oplus m \oplus_b (\Omega||(\mathcal{R}||\mathcal{F}||ID_{dn}) \\
&= m
\end{aligned}$$

Also, it can verify the signature as follows:

$$\begin{aligned}
\mathcal{S}.\mathcal{D} &= (\beta_{en} + H_a(ID_{en}||\beta_{en}||\delta_{en}).u \\
&\quad + \delta_{en}.H_c(\Omega||\mathcal{R}'||m'||ID_{en}||\delta_{en}) \\
&\quad + \Omega.H_c(\Omega||\mathcal{R}'||m'||ID_{en}||\beta_{en}) \\
&= \mathcal{S}.\mathcal{D} \\
&= (\gamma_{en} + \alpha_{en}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\delta_{en}) \\
&\quad + \mu.H_c(\Omega||\mathcal{R}||m||ID_{en}||\beta_{en})).\mathcal{D} \\
&= (\gamma_{en}.\mathcal{D} + \alpha_{en}.\mathcal{D}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\delta_{en}) \\
&\quad + \mu.\mathcal{D}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\beta_{en})) \\
&= ((\Upsilon_{en} + w.H_a(ID_{en}||\beta_{en}||\delta_{en})).\mathcal{D} \\
&\quad + \alpha_{en}.\mathcal{D}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\delta_{en}) \\
&\quad + \mu.\mathcal{D}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\beta_{en})) \\
&= (\Upsilon_{en}.\mathcal{D} + w.\mathcal{D}.H_a(ID_{en}||\beta_{en}||\delta_{en}) \\
&\quad + \alpha_{en}.\mathcal{D}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\delta_{en}) \\
&\quad + \mu.\mathcal{D}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\beta_{en})) \\
&= (\beta_{en} + H_a(ID_{en}||\beta_{en}||\delta_{en}).u \\
&\quad + \delta_{en}.H_c(\Omega||\mathcal{R}||m||ID_{en}||\delta_{en}) \\
&\quad + \Omega.H_c(\Omega||\mathcal{R}||m||ID_{en}||\beta_{en}))
\end{aligned}$$

VI. SECURITY ANALYSIS

This section aims to justify the effectiveness of the proposed scheme in resisting well-known attacks.

A. FORMAL SECURITY ANALYSIS USING AVISPA

In this subsection, results produced from the simulation work using the AVISPA tool are presented [43]. This is primarily done to ascertain the potency of the proposed scheme against replay and man-in-the-middle attacks. AVISPA is a push-button tool for providing an expressive and modular

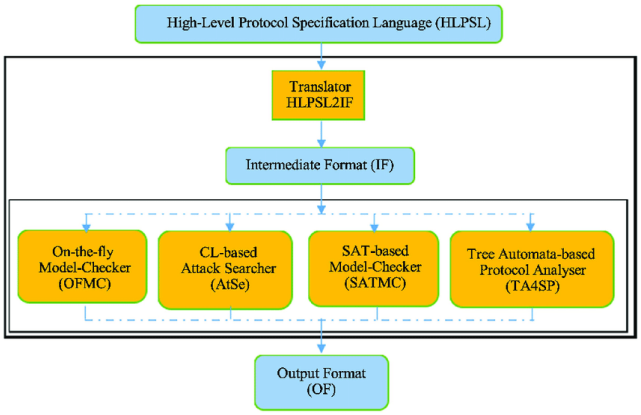


FIGURE 4. Architecture of the AVISPA tool v.1.1 [55].

formal language to simulate protocols and their security properties. SPAN [44], the security animator protocol for AVISPA, is designed to assist protocol developers write High Level Protocol Specification Language (HLPSL) specifications [45]. The HLPSL specifications of the security protocol are translated into an Intermediate Format (IF) by the HLPSLIF translator. Then, it is transformed to the output format (OF) with either On-the-fly Model-Checker (OFMC) [46], CL-based Attack Searcher (AtSe) [47], SAT-based Model-Checker (SATMC) or Tree Automata-based Protocol Analyzer (TA4SP). These embedded tools examine the security claims of the said IF code of an algorithm for two types of attack i.e. replay and man-in-the-middle attacks. The IF code works under two validation states: Safe, if the cryptographic scheme can resist the man-in-the-middle attack and; unsafe, in case the IF code does not provide resistance against man-in-the-middle attack. Formal security verification with this tool has been used in numerous studies to demonstrate the security of various authentication protocols against replay and man-in-the-middle attacks [48]–[53]. The basic architecture of the AVISPA tool is shown in Fig. 4.

The proposed scheme has been implemented for CLEN and CLDEN in HLPSL, as illustrated in Tables 2 and 3. The experiment is performed on a computer workstation having following specifications: Haier Win8.1 PC; Intel (R) Core (TM) i3-4010U CPU @ 1.70 GHz; 64-bit Operating System; x64-based processor. The software platforms consulted were Oracle VM virtual Box (version: 5.2.0.118431) and SPAN (version: SPAN-Ubuntu-10.10-light_1).

As with any security protocol to be analyzed in AVISPA, the roles for session, goal and environment have been implemented as shown in Tables 4 and 5. In order to gauge the probability of attacks on the proposed scheme, the widely-used OFMC and CL-AtSe backends are selected for the execution test. Since other backends such as SATMC and TA4SP are not compatible with bitwise XOR operations, the simulation results of SATMC and TA4SP are not included in our research work. It is essential to know whether the legitimate agents can execute the specified protocol or not.

TABLE 2. HLPSL role for CLEN.

```
role
role_Clen(Clen:agent,Clden:agent,Ben:public_key,Bdn:public_key,SND,RCV:channel(dy)) played_by Clen
def=
        local

State:

nat,Non:text,U:text,Hb:hash_func,T:text,M:text,Sk:text,Iden:text,
Encryption:hash_func,K:symmetric_key
        init
                State: = 0
        transition
 1. State=0

/\ RCV(start) =|> State':=1 /\ SND(Clen.Clden)

2. State=1

/\ RCV(Clden.{Non'}_Bdn) =|> State':=2 /\
 Iden':=new() /\ T':=new() /\ Sk':=new() /\ M':=new() /\
secret(M',sec_2,{Clen}) /\

witness (Clen,Clden,auth_1,M') /\ U':=new() /\ K':=new() /\

SND(Clen.{Encryption(M'.Sk'.T'.Iden')}_K'.{U'.Hb(M'.Sk'.T'.Iden')}_inv(Ben))
end role
```

TABLE 3. HLPSL role for CLDEN.

```
role
role_Clden(Clen:agent,Clden:agent,Ben:public_
key,Bdn:public_key,SND,RCV:channel(dy)) played_by Clden
def=
        local
State:

nat,Non:text,U:text,Hb:hash_func,
T:text, M:text, Sk:text,Iden:text,
Encryption:hash_func,K:symmetric_key

        init
                State: = 0
        transition

1. State=0

/\ RCV(Clen.Clden) =|> State':=1 /\ Non':=new() /\
SND(Clden.{Non'}_Bdn)

2. State=1

/\RCV(Clen.{Encryption(M'.Sk'.T'.Iden')}_K'.{U'.Hb(M'.Sk'.T'.Iden')}_inv(Ben))
=|> State':=2 /\ request(Clden,Clen,auth_1,M') /\
secret(M',sec_2,{Clen})
end role
```

The back-ends perform check operations to ascertain that. Then, the information about a few normal sessions between legitimate agents is provided to the intruder. Secondly, the

TABLE 4. HLPSL role for session.

```
role
session1(Clen:agent,Clden:agent,Ben:public_key,Bdn:public_key)

def=
        local
                SND2,RCV2,SND1,RCV1:channel(dy)
        composition

        role_Clden(Clen,Clden,Ben,Bdn,SND2,RCV2)
/\ role_Clen(Clen,Clden,Ben,Bdn,SND1,RCV1)

end role

role
session2(Clen:agent,Clden:agent,Ben:public_key,Bdn:public_key)

def=
        local
                SND1,RCV1:channel(dy)
        composition

role_Clen(Clen,Clden,Ben,Bdn,SND1,RCV1)
end role
```

TABLE 5. HLPSL role for environment.

```
role
 environment()

def=
        const

        hash_0:hash_func,ben:public_key,alice:agent,bob:agent,
bdn:public_key,const_1:agent,const_a:public_key,const_b:
public_key,auth_1:protocol_id,sec_2:protocol_id

intruder_knowledge = {alice,bob}

composition

session2(i,const_1,const_a,const_b) /\

session1(alice,bob,ben,bdn)

end role



goal
        authentication_on auth_1
        secrecy_of sec_2
end goal
environment()
```

susceptibility of the system to man-in-the-middle attack is also estimated by the back-ends. This is done to verify the Dolev-Yao (DY) model. The scheme is, also, simulated under SPAN (Specific Protocol Animator for AVISPA) web-tool and the results are shown in Fig. 5 and Fig.6 for OFMC and ATSE respectively. It is evident that the proposed scheme is far more secure against replay and man-in-the-middle attack.

B. PROVABLE SECURITY ANALYSIS

This section is dedicated to highlight the contributions of the proposed scheme [55] in upholding security, that includes resistance to replay attack, confidentiality, forward secrecy, integrity and unforgeability. Each of the characteristics are briefly analyzed in the following subsections.

1) CONFIDENTIALITY

Theorem 1: The proposed CL-KESC scheme for FANET is secure against the adaptively chosen ciphertext attacks if the lemmas A and B are proven true.

Lemma A: If the type 1 adversary TA_1 has the advantage ξ against the IND- CL-KESC-CCA2-I security of the proposed CL-KESC scheme for a Flying Ad-hoc Network FANET, and accomplishing $\mathcal{Q}_{Hj}$ queries to oracles H_j (j = a,b,c), $\mathcal{Q}_{CPPK}$ extract contestant partial private key query, and $\mathcal{Q}_{CFPKG}$ contestant full private key generation query.

Also, there exists a probabilistic time algorithm which solves the hyper elliptic curve discrete logarithm problem (HECDLP) with the winning probability depicted as follows:

$$\xi\cdot(1-\frac{\mathcal{Q}_{CPPK}}{\mathcal{Q}_{Ha}})\cdot(1-\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}})\cdot(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{CPPK}-\mathcal{Q}_{CFPKG})})\cdot\frac{1}{\mathcal{Q}_{Hb}}.$$

Proof: Suppose $\mathcal{D}$ is the divisor of a hyper elliptic curve and fixed $\mathcal{I} = \acute{d}.\ p.\ \mathcal{D}$ where $\acute{d}$ and p are the two randomly selected numbers from {1,2, 3 . . . . . ., q−1}. Then by using the HECCDH oracle, the challenger $\mathcal{B}$ can calculate the point $\mathcal{O} = \Phi.\ \mathcal{D}$ where $\Phi = \acute{d}.\ p(\text{mod q})$. Assume that IND- CL-KESC-CCA2-I security of proposed CL-KESC scheme can be broken by the TA_1. To do so, the challenger $\mathcal{B}$ can use TA_1 to calculate $\acute{d}.\ p.\ \mathcal{D}$ as the solution for resolving the HECDLP in the following game.

The challenger $\mathcal{B}$ selects the master secret key as: $w\epsilon$ {1,2,3, ..., q−1}. Then, it calculates a master public key u using the relation $u = w.\ \mathcal{D}$ and selects a set of public parameters. $\mathcal{B}$ keeps a master secret key w and it sends the set of public parameters with master public key u to TA_1. $\mathcal{B}$ also maintains a list $\mathcal{L}_j$ (a ≤ j ≤ c) to preserve the consistency among the responses to the hash queries asked by the TA_1 and $\mathcal{L}_\kappa$ of issue keys that are primarily unoccupied. $\mathcal{B}$ choose σ such that 1 ≤ σ ≤ $\mathcal{Q}_{Ha}$ and takes the target identity as ID_σ. $\mathcal{B}$ randomly picks υ_σ, α_σ {1,2,3, ..., q−1}, sets the variables as follows:

$$-\upsilon_\sigma = (H_a(ID_\sigma||\beta_\sigma||\delta_\sigma)); \quad \beta_\sigma = \upsilon_\sigma.u + \acute{d}.\mathcal{D} + \boldsymbol{\alpha_\sigma}.\mathcal{D};$$
$$\text{and } \delta_\sigma = \boldsymbol{\alpha_\sigma}.\mathcal{D}.$$

Also, $\mathcal{B}$ adds the pairs $(ID_\sigma\|\beta_\sigma\|\delta_\sigma\| - \upsilon_\sigma)$ and $(ID_\sigma\|\beta_\sigma\|\delta_\sigma\|\perp\|\boldsymbol{\alpha_\sigma})$ into $\mathcal{L}_a$ and $\mathcal{L}_\kappa$ respectively.

Further, $\mathcal{B}$ also answers the queries asked by H_j (a ≤ j ≤ c) of a TA_1.

H_a queries: When TA_1 asks H_a for $(ID_j\|\beta_j\|\delta_j)$ for some j ϵ[a, $\mathcal{Q}_a$], $\mathcal{B}$ checks for its availability in $\mathcal{L}_a$. In case it

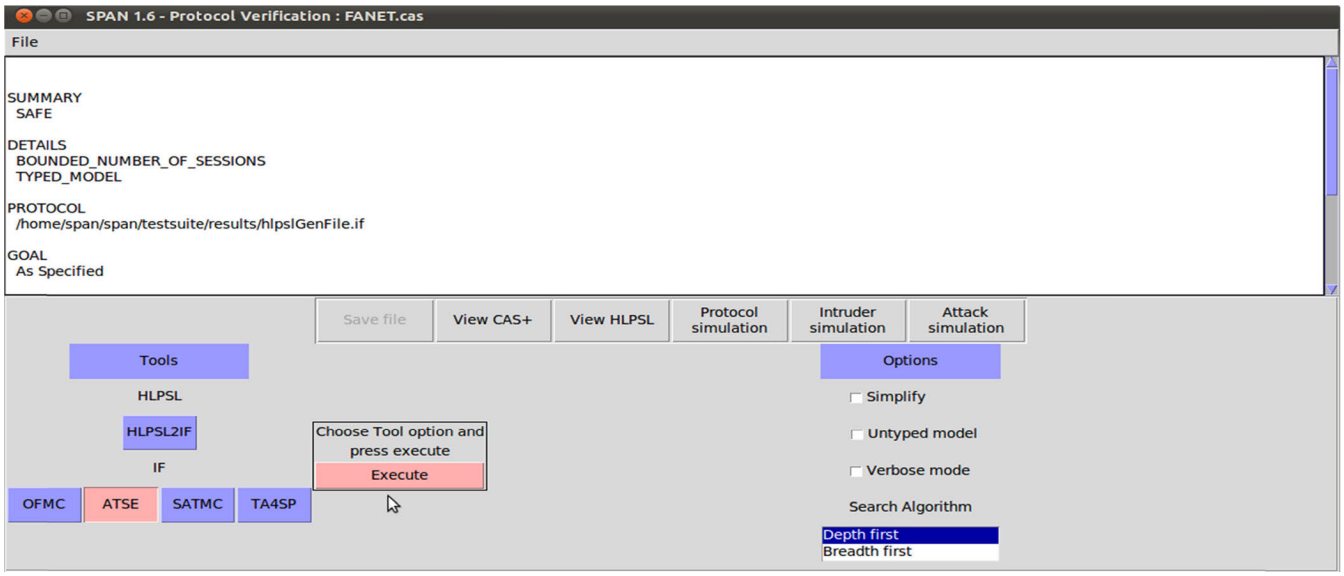


FIGURE 5. Simulation results for ASTE.

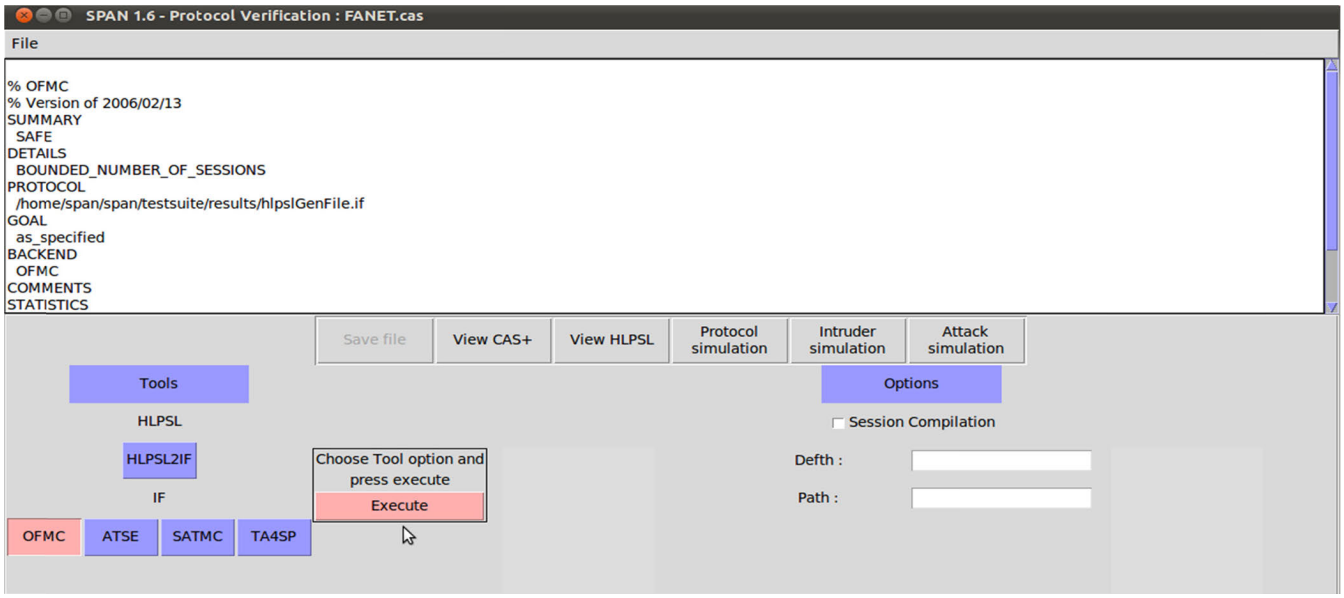


FIGURE 6. Simulation results for OFMC.

is readily available in the pair $(\mathrm{ID_j}\|\beta_j\|\delta_j\| - \upsilon_j)$, $\mathcal{B}$ returns $-\upsilon_j$ to TA_1, otherwise it randomly picks $\upsilon_j \epsilon\{1,2,3, \ldots\ldots., q-1\}$ and returns $-\upsilon_j$ to TA_1. After this step, $\mathcal{B}$ includes $(\mathrm{ID_j}\|\beta_j\|\delta_j\| - \upsilon_j)$ into $\mathcal{L}_a$.

H_b queries: Once TA_1 asks H_b for $(\Omega_j\|\mathcal{R}_j\|\chi_j\|\ \mathrm{ID_j})$ for j ϵ[b, $\mathcal{Q}_b$], for the input of HECCDH, oracle $\mathcal{B}$ sets a tuple$(\Omega_j\|\mathcal{R}_j\|\mathcal{d}.\ \mathcal{D})$. If the resulting answer of HECCDH oracle is true, then $\mathcal{B}$ suggests $\mathcal{R}_i$ as the solution of $\mathcal{d}.\ p.\ \mathcal{D}$ and discontinues. Otherwise $\mathcal{B}$ combs in $\mathcal{L}_b$, if $(\Omega_j\|^*\|\chi_j\|\ \mathrm{ID}_j\|\eta_j)$ is already available, it replaces $\mathcal{R}_j$ with * symbol and returns η_j. Otherwise, $\mathcal{B}$ randomly picks $\eta_i\{0, 1\}^q$, and sends it back to the TA_1, and $\mathcal{B}$ also $\mathcal{B}$ ncludes $(\Omega_j\|\mathcal{R}_j\|\chi_j\|\ \mathrm{ID_j}\|\eta_j)$ into $\mathcal{L}_b$.

H_c queries: Upon receiving H_c for $(\Omega_j\|\mathcal{R}_j\|m_j\|\ \mathrm{ID_j}\|\delta_j/\beta_j)$ from TA_1 for some j ϵ[c, $\mathcal{Q}_c$], $\mathcal{B}$ checks for the availability of $(\Omega_j\|\mathcal{R}_j\|m_j\|\ \mathrm{ID_j}\|\delta_j/\beta_j\|\mathcal{H}_j)$ in $\mathcal{L}_c$. If it is $\mathcal{RH}$ lready available in $\mathcal{L}_c$, $\mathcal{B}$ returns $\mathcal{H}_j$. Otherwise, $\mathcal{B}$ randomly picks $\mathcal{H}_j$ from the set $\{1,2,3, \ldots\ldots., q-1\}$ and sends it to TA_1. Further, $\mathcal{B}$ includes $(\Omega_j\|\mathcal{R}_j\|m_j\|\ \mathrm{ID_j}\|\delta_j/\beta_j\ \|\ \mathcal{H}_j)$ into $\mathcal{L}_c$.

Additionally, $\mathcal{B}$ can answer the following queries requested by TA_1.

Phase I:

A. Create-Contestant-Oracle: Upon the request of TA_1, the secret value of the contestant with identity $\mathrm{ID_j}$ is received. $\mathcal{B}$ returns α_j from $\mathcal{L}_\kappa$. Here, the contestant $\mathrm{ID_j}$ public key is not replaced.

B. Reveal-Contestant-Partial-Private-Key-Oracle: When TA_1 asks for a partial private key of a contestant with identity $\mathrm{ID_j}$, $\mathcal{B}$ checks for the answer. If $\mathrm{ID_j} = \mathrm{ID}_\sigma$ then the processing is aborted. Otherwise, $\mathcal{B}$ combs in $\mathcal{L}_\kappa$.

If ($ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j$) exists, $\mathcal{B}$ outputs the partial private key (γ_j) for TA_1. Else, it computes γ_j by calling contestant Partial Private Key Generation algorithm, sends γ_j to TA_1 and includes ($ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j$) into $\mathcal{L}_\kappa$.

C. Reveal-Contestant-Private-Key-Oracle: When TA_1 asks for the private key of a contestant with identity ID_j, $\mathcal{B}$ checks for $ID_j = ID_\sigma$. If so, the processing is aborted. Otherwise, $\mathcal{B}$ combs in $\mathcal{L}_\kappa$. If ($ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j$) exists, $\mathcal{B}$ outputs the private key pair as ($\alpha_{j,j}$) for TA_1. Otherwise, $\mathcal{B}$ picks $\upsilon_j, \alpha_i, p_j \epsilon \{1,2,3, \ldots\ldots, q-1\}$ and sets the variables as follows:

$$-\upsilon_j = (H_a(ID_j || \beta_j || \delta_j))$$
$$\beta_j = \upsilon_j . u + p_j . \mathcal{D}$$
$$\delta_j = \alpha_j . \mathcal{D} . \gamma_{j=p_j}$$

It fulfills the equation $\gamma_j . \mathcal{D} = \beta_j + H_a(ID_j \| \beta_j \| \delta_j) . u$. In the end of this process, $\mathcal{B}$ sends the pair (α_j, γ_j) to TA_1. Moreover, it includes ($ID_j \| \beta_j \| \delta_j \| -\upsilon_j$) and ($ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j$) in $\mathcal{L}_b$, and $\mathcal{L}_\kappa$ respectively.

D. Set-Contestant-Public-Key–Oracle: When TA_1 asks for the public key of a contestant with identity ID_j, $\mathcal{B}$ checks for $ID_j = ID_\sigma$. If so, it aborts further processing. Otherwise, $\mathcal{B}$ combs in $\mathcal{L}_\kappa$. If ($ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j$) exists, $\mathcal{B}$ outputs the public key pair as (β_j, δ_j) for TA_1. Otherwise, $\mathcal{B}$ picks υ_j, α_i, $p_j \epsilon \{1,2,3, \ldots\ldots, q-1\}$ and sets the variables as follows:

$$-\upsilon_j = (H_a(ID_j || \beta_j || \delta_j))$$
$$\beta_j = \upsilon_j . u + p_j . \mathcal{D}$$
$$\delta_j = \alpha_j . \mathcal{D} . \gamma_{j=p_j}$$

It fulfills the equation $\gamma_j . \mathcal{D} = \beta_{\mathcal{C}} + H_a(ID_j \| \beta_j \| \delta_j) . u$. At the end of this process, $\mathcal{B}$ sends (β_j, δ_j) to TA_1. It includes ($ID_j \| \beta_j \| \delta_j \| -\upsilon_j$) and ($ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j$) as part of $\mathcal{L}_b$ and $\mathcal{L}_\kappa$. respctively.

E. Public-Key-Replacement –Oracle: When TA_1 asks for the replacement of public key (β_j, δ_j) with ($ID_j, \beta_j^{/}, \delta_j^{/}$), then $\mathcal{B}$ prepares a tuple of the following forms accordingly: (ID_j, $-, -, {}_j^{/}, \delta_j^{/}$).

F. Symmetric-Key-Generation And Encapsulation-Oracle: When TA_1 asks for ψ with the tuple ($ID_{en} \| \beta_{en} \| \delta_{en}$, $ID_{dn} \| \beta_{dn} \| \delta_{dn} \| m$), then $\mathcal{B}$ checks for IDen $\neq ID_\sigma$. Then it calls Contestant Full PRIVATE Key Generation algorithm to compute PR_{en} of ID_{en}. In the next stage, it computes ψ by calling Symmetric Key Generation algorithm and Certificateless Encapsulation algorithm and then sends ψ to TA_1. If $ID_{en} = ID_\sigma$ and IDdn $\neq ID_\sigma$, then $\mathcal{B}$ obtains PR_{dn} by calling Contestant Full PRIVATE Key Generation, and perform following computations:

$$\Omega = \mathcal{L} . D - 1/\mathcal{H}\sigma . d . D$$
$$\mathcal{R} = (\Omega . (\alpha_{dn} + \gamma_{dn}))$$

It sets $H_c(ID_{en}, \delta en, m, \mathcal{R}, \Omega) = \mathcal{H}\sigma^{/}$, $H_c(ID_{en}, \beta en\, m, \mathcal{R}, \Omega) = \mathcal{H}\sigma$, and includes $H_c(ID_{en}, \delta en\, m, \mathcal{R}, \Omega, Hj^{/})$, $H_c(ID_{en}, \beta en\, m, \mathcal{R}, \Omega, Hj_{,})$ into $\mathcal{L}_c$, while $\mathcal{L}$, $\mathcal{H}\sigma, \mathcal{H}\sigma^{/} \epsilon \{1,2,3, \ldots\ldots, q-1\}$.

$\mathcal{B}$ calculates $\mathcal{C} = H_b(\Omega \| \mathcal{R} \| \mathcal{F} \| ID_{dn}) \oplus m$ and $\mathcal{S} = \mathcal{L} . \mathcal{H}\sigma + \alpha_{en} . \mathcal{H}\sigma^{/} - \alpha_{en}$. The output of $\mathcal{B}$ as (ID_{en},$ID_{dn,}$ $\psi = (\Omega, \mathcal{C}, \mathcal{S})$).

G. De-Encapsulation-Oracle: When TA_1 submits (ID_{en}, $ID_{dn,} \psi$) to $\mathcal{B}$, and if $ID_{dn} \neq ID_\sigma$, then $\mathcal{B}$ gets the private key of ID_{dn} and sends the result of the *De-encapsulation algorithm* to TA_1. It is obvious that, if the public key of ID_{dn} is replaced, then it cannot be possible for $\mathcal{B}$ to get the secret value of ID_{dn}. In this type of situation, $\mathcal{B}$ can obtain the secret value of ID_{dn} from TA1. Otherwise, $\mathcal{B}$ looking for ($\Omega \| R \| m \| ID_j \| \delta_j / \beta_j \| \mathcal{H}_j$) and ($\Omega \| R \| m \| ID_j \| \delta_j / \beta_j \| \mathcal{H}_j^{/}$) in $\mathcal{L}_c$. If these two entries are already available and the equality of the following equation $\mathcal{S}$. $D = \beta_{en} + \mathcal{H}_a(\beta_{en} \| ID_{en} \| \delta_{en})$. $u + \delta_{en} . Hj_+ \Omega . \mathcal{H}_j^{/}$ holds, then it retrieves $\mathcal{R}$. Further, if $\mathcal{B}$ seen a tuple ($\Omega \| \mathcal{R} \| \chi \| ID_{dn} \| \eta$) from $\mathcal{L}_b$ and producing HECCDH oracle gives positive result; means 1 on the query ($d.D, \Omega, \mathcal{R}$) then the plaintext (message) is $\mathcal{C} \oplus \eta$.

Challenge: In this section, the TA_1 submits sender and recipient identity (ID_{en}, ID_{dn}), along with two different but the same length messages $\mathcal{M}_0$ and $\mathcal{M}_1$. However, in *Phase I*, TA_1 is restricted to calling *Reveal-Contestent-Secrete-Value-Oracle on* ID_{dn}^*. Also, it cannot extract the partial private of ID_{dn}^* and the public key has not been changed yet. Here, if $ID_{dn} \neq ID_\sigma$, then $\mathcal{B}$ stop the execution of this game. Otherwise, $\mathcal{B}$ can make the challenge signcrypted text from the following steps.

1. It sets $\Omega* = p . \mathcal{D}$, and choose $\mathcal{R}^* \epsilon \{1,2,3, \ldots\ldots, q-1\}$
2. Pick a random bit $\partial \epsilon \{0,1\}$ and hash value $\mathcal{H}$, then compute $C^* = \mathcal{M}\partial \oplus \mathcal{H}$
3. It sets $\mathcal{S}^*$ and it satisfies $\mathcal{S}^*$. $\mathcal{D} = (\beta_{en} + H_a(ID_{en} \| \beta_{en} \| \delta_{en}) . u + \delta_{en} . H_c(\Omega * \| \mathcal{R}^* \| m \| ID_{en} \| \delta_{en}) + \Omega^* . H_c(\Omega^* \| \mathcal{R}^* \| m \| ID_{en} \| \beta_{en})$
4. Sends $\psi^* = (\mathcal{S}^*, C^* \Omega^*)$ to TA_1.

Phase II: Just like in Phase 1, TA_1 probe for the same oracle query adaptively, ignoring the Reveal-Contestant-Partial-Private-Key-Oracle and Reveal-Contestant-Secrete-Value-Oracle with recipient identity, further, excepting Certificateless De-encapsulation-Oracle with (ψ^*, ID_{en}, ID_{dn}), unless the public key of ID_{en}, ID_{dn} has been altered.

***Guess*:** So, TA_1 is capable to break IND- CL-KESC-CCA2-I security of proposed CL-KESC scheme for flying ad-hoc network, a H_b query should have been requested with ($\Omega^* \| \mathcal{R}^* \| \chi^* \| ID_{dn}^*$). Here, $\mathcal{R}^* = (p . H_a(\beta_{dn} \| ID_{dn} \| \delta_{dn}) . u + \beta_{dn} + \delta_{dn}) =$ d. $p . \mathcal{D}$. One of the $\mathcal{R}$ in $\mathcal{L}_b$ is the solution of hyper elliptic curve discrete logarithm problem. $\mathcal{B}$ randomly pick one $\mathcal{R} \epsilon \{1,2,3, \ldots\ldots, q-1\}$ and results as the clarification of HECDLP.

Analyses: Suppose $\Xi_{a,} \Xi_{b,}$ and Ξ_c be the three events, in which $\mathcal{B}$ stop the execution of a game.

a) Ξ_a is that event, when TA_1 asked for a partial private key of ID_σ and the probability of Ξ_a is $\frac{Q_{CPPK}}{Q_{Ha}}$.
b) Ξ_b is that event, when TA_1 asked for a private key of ID_σ and the probability of Ξ_a is $\frac{Q_{CFPKG}}{Q_{Ha}}$.

c) Ξ_c is that event, in which ID_σ has not been selected as a de-encapsulation by TA_1 and the probability of Ξ_c is $1 - \frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{CPPK}-\mathcal{Q}_{CFPKG})}$.
Hence, $\mathcal{B}$ may not stop this game's probability of PROB $[\neg\Xi_a \wedge \neg\Xi_b \wedge \neg\Xi_c] = (1 - \frac{\mathcal{Q}_{CPPK}}{\mathcal{Q}_{Ha}}).(1 - \frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{CPPK}-\mathcal{Q}_{CFPKG})})$.
$\mathcal{B}$ may pick the problem's solution of HECDLP from $\mathcal{L}_b$ with probability $\frac{1}{\mathcal{Q}_{Hb}}$. Thus, the successful advantage ξ of $\mathcal{B}$ is $ADV_{\mathcal{B}}^{HECCDH} = \xi.(1 - \frac{\mathcal{Q}_{CPPK}}{\mathcal{Q}_{Ha}}).(1 - \frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{CPPK}-\mathcal{Q}_{CFPKG})}).\frac{1}{\mathcal{Q}_{Hb}}$.

Lemma B: If the type 2 adversary TA_2 has the advantage ξ against the IND- CL-KESC-CCA2-II for breaking the security of proposed CL-KESC scheme for flying ad-hoc network and accomplishing $\mathcal{Q}_{Hj}$ queries to oracles H_j (j = a,b,c), $\mathcal{Q}_{Csv}$ extract Contestant Secret Value query, and $\mathcal{Q}_{CFPKG}$ contestant full private key generation query. Also, there exists a probabilistic time algorithm which solves the hyper elliptic curve discrete logarithm problem (HECDLP) with the winning probability ξ. $(1 - \frac{\mathcal{Q}_{Csv}}{\mathcal{Q}_{Ha}}).(1 - \frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{Csv}-\mathcal{Q}_{CFPKG})})$.

Proof: Let $\mathcal{D}$ be the divisor of a genus 2 hyper elliptic curve and set $\mathcal{J} = \acute{d}.\ p.\ \mathcal{D}$, here $\acute{d}$ and p are the two uniformly selected numbers from {1,2,3 . . . . . ., q−1}. Then by using HECCDH oracle, the challenger $\mathcal{B}$ can calculate the point = $\Phi.\ \mathcal{D}$ where $\Phi = d.\ p$(mod q). Let the IND- CL-KESC-CCA2-II security of proposed CL-KESC scheme can break by TA_2, for this purpose the challenger $\mathcal{B}$ can use TA_2 to calculate $\acute{d}.\ p.\ \mathcal{D}$ as the solution of solving *HECDLP* in the following game.

$\mathcal{B}$ picks a master secret key $\mathcal{W}$ as: {1,2,3, ..., q−1}. Then, it calculates a master public key u using the relation $u = \mathcal{W}.\ \mathcal{D}$ and selects the set of public parameters. $\mathcal{B}$ sends the set of public parameters with master public key u and secrete key $\mathcal{W}$ to TA_2. $\mathcal{B}$ also maintains a list $\mathcal{L}_j$ (a ≤ j ≤ c) to save the consistency among the responses to the asked hash queries by TA_2 and $\mathcal{L}_\kappa$ of issue keys that are primarily unoccupied. $\mathcal{B}$ choose σ such that $(1 \le \sigma \le \mathcal{Q}_{Ha})$ and take the target identity as ID_σ. $\mathcal{B}$ randomly pick $e_\sigma, x_\sigma \in$ {1,2,3, ..., q−1}, set $e_\sigma = (H_a(ID_\sigma \| \beta_\sigma \| \delta_\sigma))$, $\beta_\sigma = x_\sigma.\mathcal{D}$, $\gamma_\sigma = x_\sigma + \mathcal{W}.\ e_\sigma$, and $\delta_\sigma = d\ D$. Also, $\mathcal{B}$ includes $(ID_\sigma \| \beta_\sigma \| \delta_\sigma \| e_\sigma)$ into the $\mathcal{L}_a$ and $(ID_\sigma \| \beta_\sigma \| \delta_\sigma \| \perp \| \gamma_\sigma)$ into the $\mathcal{L}_\kappa$.

Further, $\mathcal{B}$ answers the following asked queries H_j (a ≤ j ≤ c) of a TA_2.

H_a queries: Once TA_2 asked H_a query for $(ID_j \| \beta_j \| \delta_j)$ for some j ∈[a, $\mathcal{Q}_a$], $\mathcal{B}$ checks in $\mathcal{L}_a$, if it is already available $(ID_j \| \beta_j \| \delta_j \| e_j)$ then $\mathcal{B}$ return e_j to TA_2, otherwise it randomly picks $e_j \in$ {1,2,3, ..., q−1} and return e_j to TA_2. After this, $\mathcal{B}$ includes $(ID_j \| \beta_j \| \delta_j \| e_j)$ into $\mathcal{L}_a$.

H_b queries: If TA_2 submit H_b query for $(\Omega_j \| \mathcal{R}_j \| \chi_j \| ID_j)$ for some j ∈[b, $\mathcal{Q}_b$], here, for the input of HECCDH oracle $\mathcal{B}$ sets a tuple$(\Omega_j \| \mathcal{R}_j \| \acute{d}.\ \mathcal{D})$. If the resulting answer of HECCDH oracle is true, then $\mathcal{B}$ suggest χ_j as the solution of $\acute{d}.\ p.\ \mathcal{D}$ and discontinues, otherwise $\mathcal{B}$ combs in $\mathcal{L}_b$, if $(\Omega_j \| ^* \| \mathcal{R}_j \| ID_j \| e_j)$ is already available, it replaces χ_j with * symbol and returns e_j. Otherwise, $\mathcal{B}$ randomly picks $e_j \in \{0,1\}^q$, sends e_j back to the TA_2, and also $\mathcal{B}$ includes $(\Omega_j \| \mathcal{R}_j \| \chi_j \| ID_j \| e_j)$ into $\mathcal{L}_b$.

H_c queries: When TA_2 submit H_c query for $(\Omega_j \| \mathcal{R}_j \| m_j \| ID_j \| \delta_j / \beta_j)$ for some j ∈[c, $\mathcal{Q}_c$]. $\mathcal{B}$ checks in $\mathcal{L}_c$, if $(\Omega_j \| \mathcal{R}_j \| m_j \| ID_j \| \delta_j / \beta_j \| \mathcal{H}_j)$ already presented in $\mathcal{L}_c$, $\mathcal{B}$ return $\mathcal{H}_j$. Then, $\mathcal{B}$ erratically pick $\mathcal{H}_j \in$ {1,2,3, ..., q−1} and send it to TA_2. Further, $\mathcal{B}$ comprise $(\Omega_j \| \mathcal{R}_j \| m_j \| ID_j \| \delta_j / \beta_j \| \mathcal{H}_j)$ into $\mathcal{L}_c$.

Additionally, $\mathcal{B}$ can answer the below requested queries of TA_1.

A. Create-Contestant-Key-Oracle: When TA_2 asked for the secret value of the contestant with identity ID_j, then $\mathcal{B}$ first check if $ID_j = ID_\sigma$ then it abort further processing. If IDj ≠ ID_σ, $\mathcal{B}$ combs in $\mathcal{L}_\kappa$, if $(ID_j \| \beta_j \| \delta_j \| \boldsymbol{\alpha_j} \| \gamma_j)$ exists, $\mathcal{B}$ outputs is $\boldsymbol{\alpha_j}$ for TA_2. Otherwise, $\mathcal{B}$ uniformly selects e_j, $\mathbf{x_j}$, and $\boldsymbol{\alpha_j} \in$ {1,2,3, ..., q−1} and then set $e_j = (H_a(ID_j \| \beta_j \| \delta_j))$, $\beta_j = \mathbf{x_j}.\mathcal{D}$, $\gamma_j = \mathbf{x_j} + \mathcal{W}.\ e_j$, and $\delta_j = \boldsymbol{\alpha_j}\ \mathcal{D}$. $\mathcal{B}$ send $\boldsymbol{\alpha_j}$ to TA_2 and includes $(ID_j \| \beta_j \| \delta_j \| \boldsymbol{\alpha_j} \| \gamma_j)$ into $\mathcal{L}_\kappa$.

B. Reveal-Contestant-Private-Key-Oracle: When TA_2 asked for the secret value of the contestant with identity ID_j, then $\mathcal{B}$ first check if $ID_j = ID_\sigma$ then it abort further processing. If IDj ≠ ID_σ, $\mathcal{B}$ combs in $\mathcal{L}_\kappa$, if $(ID_j \| \beta_j \| \delta_j \| \boldsymbol{\alpha_j} \| \gamma_j)$ exists, $\mathcal{B}$ outputs is $\boldsymbol{\alpha_j}$ for TA_2. Otherwise, $\mathcal{B}$ uniformly selects e_j, $\mathbf{x_j}$, and $\boldsymbol{\alpha_j} \in$ {1,2,3, ..., q−1} and then set $e_j = (H_a(ID_j \| \beta_j \| \delta_j))$, $\beta_j = \mathbf{x_j}.\mathcal{D}$, $\gamma_j = \mathbf{x_j} + \mathcal{W}.\ e_j$, and $\delta_j = \boldsymbol{\alpha_j}\ \mathcal{D}$. $\mathcal{B}$ send $(\boldsymbol{\alpha_j}, \gamma_j)$ to TA_2 and includes $(ID_j \| \beta_j \| \delta_j \| \boldsymbol{\alpha_j} \| \gamma_j)$ into $\mathcal{L}_\kappa$.

C. Set-Contestant-Public-Key –Oracle: When TA_2 asked for the public key of a contestant with identity ID_j, after this $\mathcal{B}$ check, if $ID_j = ID_\sigma$ then it abort further processing. Otherwise, $\mathcal{B}$ combs in $\mathcal{L}_\kappa$, if $(ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j)$ exists, $\mathcal{B}$ outputs the public key pair as (β_j, δ_j) for TA_2 Otherwise, $\mathcal{B}$ uniformly selects e_j, $\mathbf{x_j}$, and $\boldsymbol{\alpha_j} \in$ {1,2,3, ..., q−1} and then set $e_j = (H_a(ID_j \| \beta_j \| \delta_j))$, $\beta_j = \mathbf{x_j}.\mathcal{D}$, $\gamma_j = \mathbf{x_j} + \mathcal{W}.\ e_j$, and $\delta_j = \boldsymbol{\alpha_j}\ D$. At the end of this process, $\mathcal{B}$ sends (β_j, δ_j) to TA_2 and includes $(ID_j \| \beta_j \| \delta_j \| e_j)$ into $\mathcal{L}_a$, also includes $(ID_j \| \beta_j \| \delta_j \| \alpha_j \| \gamma_j)$ into $\mathcal{L}_\kappa$.

D. Symmetric-Key-Generation And Encapsulation-Oracle: When TA1 asked for ψ with the tuple $(ID_{en} \| \beta_{en} \| \delta_{en}, ID_{dn} \| \beta_{dn} \| \delta_{dn} \| m)$, then $\mathcal{B}$ first check, if IDen ≠ ID_σ then it calls Contestant Full PRIVATE Key Generation algorithm to compute PR_{en} of ID_{en}. After, it computes ψ by calling Symmetric Key Generation algorithm and Certificateless Encapsulation algorithm then send ψ to TA1. If $ID_{en} = ID_\sigma$ and IDdn ≠ ID_σ, then $\mathcal{B}$ obtain PR_{dn} by calling Contestant Full PRIVATE Key Generation, compute $\Omega = \mathcal{L}.\ D - \mathcal{H}\sigma / \mathcal{H}\sigma'.d.D$, $\mathcal{R} = (\Omega.(\alpha_{dn} + \gamma_{dn}))$, set $H_c(ID_{en}, \delta en, m, \mathcal{R}, \Omega) = \mathcal{H}\sigma'$, $H_c(ID_{en}, \beta en\, m, \mathcal{R}, \Omega) = \mathcal{H}\sigma$, and includes $H_c(ID_{en}, \delta en\, m, \mathcal{R}, \Omega\mathcal{H}_j')$, $H_c(ID_{en}, \beta en\, m, \mathcal{R}, \Omega, \mathcal{H}_j)$ into $\mathcal{L}_c$, while $\mathcal{L}, \mathcal{H}\sigma, \mathcal{H}\sigma' \in$ {1,2,3, ..., q−1}. $\mathcal{B}$ calculate $\mathcal{C} = H_b(\Omega \| \mathcal{R} \| \mathcal{F} \| ID_{dn}) \oplus m$ and $\mathcal{S} = \mathcal{L}.\ \mathcal{H}\sigma + \gamma\sigma$. The output of $\mathcal{B}$ as $(ID_{en}, ID_{dn}, \psi = (\Omega, \mathcal{C}, \mathcal{S}))$.

E. De-Encapsulation-Oracle: When TA1 submits (ID_{en}, ID_{dn}, ψ) to $\mathcal{B}$, and if $ID_{dn} \neq ID_\sigma$, then $\mathcal{B}$ get the private key of ID_{dn} and send the result of a *De-encapsulation algorithm*

to TA_2. Otherwise, $\mathcal{B}$ looking for $(\Omega\|\mathcal{R}\|m\|\ ID_j\|\delta_j/\beta_j\|\mathcal{H}j)$ and $(\Omega\|\mathcal{R}\|m\|\ ID_j\|\delta_j/\beta_j\|\mathcal{H}j')$ in $\mathcal{L}_c$. If these two entries are already available and the equality of the following equation $\mathcal{S}.\ D = \beta_{en} + H_a\ (\beta_{en}\|\ ID_{en}\|\delta_{en}).\ u + \delta_{en}.\ \mathcal{H}j_+\Omega.\ \mathcal{H}j'$ holds, then it retrieves $\mathcal{R}$. Further, if $\mathcal{B}$ seen a tuple $(\Omega\|\mathcal{R}\|\chi\|\ ID_{dn}\|\eta)$ from $\mathcal{L}_b$ and producing HECCDH oracle gives positive result; means 1 on the query $(d.\ D\Omega, \mathcal{R})$ then the plaintext (message) is $\mathcal{C} \oplus \eta$.

Challenge: In this section, the TA_2 submits sender and recipient identity (ID^*_{dn},ID^*_{dn}), along with two different but the same length messages $\mathcal{M}_0$ and $\mathcal{M}_1$. However, in Phase I, TA_2 is restricted to calling Reveal-Contestant-Private-Key-Oracle on ID^*_{dn}. Here, if $ID_{dn} \neq ID_\sigma$, then $\mathcal{B}$ stop the execution of this game. Otherwise, $\mathcal{B}$ can make the challenge signcrypted text from the following steps.

1. It sets $\Omega* = p.\ \mathcal{D}$, and choose $\mathcal{R}^*\{1,2,3, \ldots\ldots., q-1\}$
2. Pick a random bit $\partial\epsilon$ {0,1} and hash value H, then compute $\mathcal{C}^* = \mathcal{M}\partial \oplus \mathcal{H}$
3. It sets $\mathcal{S}$ * and it satisfies $\mathcal{S}$ *. $\mathcal{D} = (\beta_{en} + H_a\ (ID_{en}\|\beta_{en}\|\delta_{en}).\ u + \delta_{en}.\ H_c(\Omega^*\|\mathcal{R}^*\|m\|ID_{en}\|\delta_{en}) + \Omega^*.\ H_c\ (\Omega * \|\mathcal{R}^*\|m\|ID_{en}\|\beta_{en})$
4. Sends $\psi^* = (\mathcal{S}$ *, $\mathcal{C}^*\Omega^*)$ to TA_2.

Phase II: Just like in Phase 1, TA_2 probe for the same oracle query adaptively, excepting Certificateless De-encapsulation-Oracle on ψ^*.

Guess: TA_2 is capable to break IND- CL-KESC-CCA2-II security of proposed CL-KESC scheme for FANET, a H_b query should have been requested with $(\Omega^*\|\mathcal{R}^*\|\chi^*\|\ ID^*_{dn})$. Here, $\mathcal{R}^* = \lambda = \alpha_{dn}.\Omega = d.\ p.\ \mathcal{D}$. One of the χ^* in $\mathcal{L}_b$ is the solution of hyper elliptic curve discrete logarithm problem. $\mathcal{B}$ randomly pick one χ^* {1,2,3, ……., q−1} and results as the clarification of HECDLP.

Analyses: *Suppose* $\Xi_a, \Xi_b,$ and Ξ_c be the three events, in which $\mathcal{B}$ stop the execution of a game.

a) Ξ_a is that event, when TA_2 asked for a secret value of ID_σ and the probability of Ξ_a is $\frac{Q_{Csv}}{Q_{Ha}}$.
b) Ξ_b is that event, when TA_2 asked for a full private key of ID_σ and the probability of Ξ_a is $\frac{Q_{CFPKG}}{Q_{Ha}}$.
c) Ξ_c is that event, in which ID_σ has not been selected as a de-encapsulation by TA_2 and the probability of Ξ_c is $1 - \frac{1}{(Q_{Ha}-Q_{Csv}-Q_{CFPKG})}$.

Hence, $\mathcal{B}$ may not stop this game's probability of PROB $[\neg\Xi_a \wedge \neg\Xi_b \wedge \neg\Xi_c] = \xi.(1 - \frac{Q_{Csv}}{Q_{Ha}}).(1 - \frac{Q_{CFPKG}}{Q_{Ha}}).(\frac{1}{(Q_{Ha}-Q_{Csv}-Q_{CFPKG})})$.

$\mathcal{B}$ may pick the problem's solution of HECDLP from $\mathcal{L}_b$ with probability $\frac{1}{Q_{Hb}}$. Thus, the successful advantage ξ of $\mathcal{B}$ is $ADV_{\mathcal{B}}$ HECCDH $= \xi.(1 - \frac{Q_{Csv}}{Q_{Ha}}).(1 - \frac{Q_{CFPKG}}{Q_{Ha}}).(\frac{1}{(Q_{Ha}-Q_{Csv}-Q_{CFPKG})})$.

Theorem 2: The proposed CL-KESC scheme for FANET is secure against EUF- CL-KESC-CMA-I if the following Lemma A and B is proved.

Lemma C: If the type 1 forger TF_1 has the advantage ξ against the EUF- CL-KESC-CMA-I security of proposed CL-KESC scheme for flying ad-hoc network and accomplishing Q_{Hj} queries to oracles H_j (j = a,b,c), Q_{CPPK} extract contestant partial private key query, and Q_{CFPKG} contestant full private key generation query. Also, there exists a probabilistic time algorithm which solves the HECDLP with the winning probability $\xi.(1 - \frac{Q_{CPPK}}{Q_{Ha}}).(1 - \frac{Q_{CFPKG}}{Q_{Ha}}).(\frac{1}{(Q_{Ha}-Q_{CPPK}-Q_{CFPKG})})$.

Proof: Suppose $\mathcal{D}$ is the divisor of a hyper elliptic curve and fixed $\mathcal{I} = d'.\ p.\ \mathcal{D}$ where d', p are the two randomly selected numbers from {1,2, 3 ……., q−1}. Then by using HECCDH oracle, the challenger $\mathcal{B}$ can calculate the point $\mho = \Phi.\ \mathcal{D}$ where $\Phi = d.\ p(\text{mod q})$. Let the *EUF- CL-KESC-CMA-I* security of proposed CL-KESC scheme can break by TF_1, for this purpose the challenger $\mathcal{B}$ can use TF_1 to calculate $d'.\ p.\ \mathcal{D}$ as the solution of solving *HECDLP* in the following game.

Training Phase: TF_1 could create a same series of queries oracles as like in the IND- *CL-KESC*-CCA2-I game in *Lemma A*.

Forgery: TF_1 outputs is a valid signcrypted text from encapsulation ID_{en} to the de-encapsulation ID_{dn}. If $ID_{en} \neq ID_\sigma$ then $\mathcal{B}$ stop the further processing of the game. So, it is observed from forking lemma [56], if there exist an efficient and powerful TF_1,in the aforementioned interaction, then there exists a polynomial time Turing machine $TF_1^{\overline{1}}$ that make two signcrypted text triples i.e. $(\Omega^*, \mathcal{C}^*, \mathcal{S})$ and $(\Omega^*, \mathcal{C}^*, \mathcal{S}^*)$ on the similar plaintext (m). Then we may two types of output such as. $\mathcal{D} = \beta_{en} - \upsilon_\sigma.\ u + \delta_{en}.\ \mathcal{H}\sigma' + \Omega.\ \mathcal{H}\sigma$ and $\mathcal{S}^*.\ \mathcal{D} = \beta_{en} - \upsilon_\sigma.\ u + \delta_{en}.\ \mathcal{H}\sigma' + \Omega.\ \mathcal{H}\sigma^*$. Suppose $\Omega = p.\ \mathcal{D}$ and then we get the following output:

$$
\begin{aligned}
(\mathcal{S}.\mathcal{D}) - (\mathcal{S}^*.\mathcal{D}) &= (\beta_{en} - \upsilon_\sigma.u + \delta_{en}.\mathcal{H}\sigma' + \Omega.\mathcal{H}\sigma) \\
&\quad - (\beta_{en} - \upsilon_\sigma.u + \delta_{en}.\mathcal{H}\sigma' + \Omega.\mathcal{H}\sigma^*) \\
&= ((\beta_{en} - \upsilon_\sigma.u) + \delta_{en}.\mathcal{H}\sigma' + \Omega.\mathcal{H}\sigma) \\
&\quad - ((\beta_{en} - \upsilon_\sigma.u) + \delta_{en}.\mathcal{H}\sigma' + \Omega.\mathcal{H}\sigma^*) \\
(\mathcal{S}.\mathcal{D}) - (\mathcal{S}^*.\mathcal{D}) &= \Omega.\mathcal{H}\sigma - \Omega.\mathcal{H}\sigma^* \\
\mathcal{D}.(\mathcal{S} - \mathcal{S}^*) &= \Omega.(\mathcal{H}\sigma - \mathcal{H}\sigma^*) \\
\mathcal{D}.(\mathcal{S} - \mathcal{S}^*) &= p.\mathcal{D}.(\mathcal{H}\sigma - \mathcal{H}\sigma^*) \\
(\mathcal{S} - \mathcal{S}^*) &= p.(\mathcal{H}\sigma - \mathcal{H}\sigma^*) \\
\frac{(\mathcal{S}\ \mathcal{S}*)}{(\mathcal{H}\sigma - \mathcal{H}\sigma*)} &= p
\end{aligned}
$$

Hence, $\mathcal{B}$ can solve the hard problem of HECDLP as $p = \frac{(\mathcal{S}\ \mathcal{S}*)}{(\mathcal{H}\sigma - \mathcal{H}\sigma*)}$.

Analyses: *Suppose* $\Xi_a, \Xi_b,$ and Ξ_c be the three events, in which $\mathcal{B}$ stop the execution of a game.

a) Ξ_a is that event, when *forger* TF_1 asked for a partial private key of ID_σ and the probability of Ξ_a is $\frac{Q_{CPPK}}{Q_{Ha}}$.
b) Ξ_b is that event, when *forger* TF_1 asked for a private key of ID_σ and the probability of Ξ_a is $\frac{Q_{CFPKG}}{Q_{Ha}}$.
c) Ξ_c is that event, in which ID_σ has not been selected as a de-encapsulation by *forger* TF_1 and the probability of Ξ_c is $1 - \frac{1}{(Q_{Ha}-Q_{CPPK}-Q_{CFPKG})}$.

Hence, $\mathcal{B}$ may not stop this game's probability of PROB $[\neg\Xi_{\mathrm{a}}\lambda\neg\Xi_{\mathrm{b}}\lambda\neg\Xi_{\mathrm{c}}]= (1-\frac{\mathcal{Q}_{CPPK}}{\mathcal{Q}_{Ha}}).(1-\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{CPPK}-\mathcal{Q}_{CFPKG})})$.
$\mathcal{B}$ may pick the problem's solution of HECDLP from $\mathcal{L}_{\mathrm{b}}$ with probability $\frac{1}{\mathcal{Q}_{Hb}}$. Thus, the successful advantage ξ of $\mathcal{B}$ is $\mathrm{ADV}_{\mathcal{B}}^{HECCDH} = \xi.(1-\frac{\mathcal{Q}_{CPPK}}{\mathcal{Q}_{Ha}}).(1-\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{CPPK}-\mathcal{Q}_{CFPKG})})$.

Lemma D: If the type 2 adversary TF_2 has the advantage ξ against the EUF- CL-KESC-CMA-II for breaking the security of proposed CL-KESC scheme for flying ad-hoc network and accomplishing $\mathcal{Q}_{\mathrm{Hj}}$ queries to oracles H_{j} (j = a,b,c), $\mathcal{Q}_{\mathrm{Csv}}$ extract Contestant Secret Value query, and $\mathcal{Q}_{\mathrm{CFPKG}}$ contestant full private key generation query. Also, there exists a probabilistic time algorithm which solves the *HECDLP with the winning probability* ξ. *(1-*$\frac{\mathcal{Q}_{Csv}}{\mathcal{Q}_{Ha}}$*)*.$(1-\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{Csv}-\mathcal{Q}_{CFPKG})})$.

Proof: Let $\mathcal{D}$ be the divisor of a genus 2 hyper elliptic curve and set $\mathcal{I} = \acute{d}.\ p.\ \mathcal{D}$, here $\acute{d}$ and p are the two uniformly selected numbers from $\{1,2,3\ldots\ldots,\mathrm{q}-1\}$. Then by using HECCDH oracle, the challenger $\mathcal{B}$ can calculate the point $= \Phi.\ \mathcal{D}$ where $\Phi = d.\ p(\mathrm{mod\ q})$. Let the *EUF- CL-KESC-CMA-II* security of proposed CL-KESC scheme can break by TF_2, for this purpose the challenger $\mathcal{B}$ can use TF_2 to calculate $\acute{d}.\ p.\ \mathcal{D}$ as the solution of solving *HECDLP* in the following game.

$\mathcal{B}$ picks a master secret key $\mathcal{W}$ as: {1,2,3, ……., q−1}. Then, it calculates a master public key u using the relation $u = \mathcal{W}.\ \mathcal{D}$ and selects the set of public parameters. $\mathcal{B}$ sends the set of public parameters with master public key u and secrete key $\mathcal{W}$ to TF_2. $\mathcal{B}$ also maintains a list $\mathcal{L}_{\mathrm{j}}$ (a ≤ j ≤ c) to save the consistency among the responses to the asked hash queries by TF_2 and $\mathcal{L}_{\kappa}$ of issue keys that are primarily unoccupied.

***Training Phase*:** TF_2 could create a same series of queries oracles as like in the IND- *CL-KESC*-CCA2-II game in *Lemma B*.

Forgery: Finally, TF_1 produced is an effective signcrypted text from encapsulation $\mathrm{ID}_{\mathrm{en}}$ to the de-encapsulation $\mathrm{ID}_{\mathrm{dn}}$. If $\mathrm{ID}_{\mathrm{en}} \neq \mathrm{ID}_{\sigma}$ then $\mathcal{B}$ discontinue the running of this game. Consequently, it is perceived from forking lemma [56], uncertainty there available an effective and influential TF_2, in the aforesaid interaction, then there exists a polynomial time another Turing machine $\mathrm{TF}_2^{\overline{1}}$ that make two signcrypted text triples i.e. (Ω^*, $\mathcal{C}^*$, $\mathcal{S}$) and (Ω^*, $\mathcal{C}^*$, $\mathcal{S}^*$) on the similar plaintext (m). Then we may two types of output such as $\mathcal{S}.\mathcal{D} = \beta_{\mathrm{en}}.\mathrm{e}_{\sigma}.\ u + \delta_{\mathrm{en}}.\mathcal{H}\sigma^{/} + \Omega.\mathcal{H}\sigma$ and $\mathcal{S}^*.\ \mathcal{D} = \beta_{\mathrm{en}}.\mathrm{e}_{\sigma}.\ u + \delta_{\mathrm{en}}.\mathcal{H}\sigma^{/} + \Omega.\mathcal{H}\sigma^*$. Suppose $\Omega = p.\ \mathcal{D}$ and then we get the following output:

$$\begin{aligned}(\mathcal{S}.\mathcal{D}) - (\mathcal{S}^*.\mathcal{D}) &= (\beta_{\mathrm{en}}.\mathrm{e}_{\sigma}.u + \delta_{\mathrm{en}}.\mathcal{H}\sigma^{/} + \Omega.\mathcal{H}\sigma) \\ &\quad - (\beta_{\mathrm{en}}.\mathrm{e}_{\sigma}.u + \delta_{\mathrm{en}}.\mathcal{H}\sigma^{/} + \Omega.\mathcal{H}\sigma^*) \\ &= ((\beta_{\mathrm{en}}.\mathrm{e}_{\sigma}.u) + \delta_{\mathrm{en}}.\mathcal{H}\sigma^{/} + \Omega.\mathcal{H}\sigma) \\ &\quad - ((\beta_{\mathrm{en}}.\mathrm{e}_{\sigma}.u) + \delta_{\mathrm{en}}.\mathcal{H}\sigma^{/} + \Omega.\mathcal{H}\sigma^*)\end{aligned}$$

$$(\mathcal{S}.\mathcal{D}) - (\mathcal{S}^*.\mathcal{D}) = \Omega.\mathcal{H}\sigma - \Omega.\mathcal{H}\sigma^*$$

$$\mathcal{D}.(\mathcal{S}-\mathcal{S}^*) = p.\mathcal{D}.(\mathcal{H}\sigma - \mathcal{H}\sigma^*)$$

$$(\mathcal{S}-\mathcal{S}^*) = p.(\mathcal{H}\sigma - \mathcal{H}\sigma^*)$$

$$\frac{(\mathcal{S}\,\mathcal{S}*)}{(\mathcal{H}\sigma - \mathcal{H}\sigma*)} = p$$

Hence, $\mathcal{B}$ can solve the hard problem of HECDLP as $p = \frac{(\mathcal{S}\,\mathcal{S}*)}{(\mathcal{H}\sigma-\mathcal{H}\sigma*)}$.

Analyses: *Suppose* $\Xi_{\mathrm{a}}, \Xi_{\mathrm{b}}$, and Ξ_{c} be the three events, in which $\mathcal{B}$ stop the execution of a game.

a) Ξ_{a} is that event, when TF_2 asked for a secret value of ID_{σ} and the probability of Ξ_{a} is $\frac{\mathcal{Q}_{Csv}}{\mathcal{Q}_{Ha}}$.
b) Ξ_{b} is that event, when TF_2 asked for a full private key of ID_{σ} and the probability of Ξ_{a} is $\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}$.
c) Ξ_{c} is that event, in which ID_{σ} has not been selected as a de-encapsulation by TF_2 and the probability of Ξ_{c} is $1-\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{Csv}-\mathcal{Q}_{CFPKG})}$.
Hence, $\mathcal{B}$ may not stop this game's probability of PROB $[\neg\Xi_{\mathrm{a}}\wedge\neg\Xi_{\mathrm{b}}\wedge\neg\Xi_{\mathrm{c}}]= \xi.(1-\frac{\mathcal{Q}_{Csv}}{\mathcal{Q}_{Ha}}).(1-\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{Csv}-\mathcal{Q}_{CFPKG})})$.
$\mathcal{B}$ may pick the problem's solution of HECDLP from $\mathcal{L}_{\mathrm{b}}$ with probability $\frac{1}{\mathcal{Q}_{Hb}}$. Thus, the successful advantage ξ of $\mathcal{B}$ is $\mathrm{ADV}_{\mathcal{B}}^{HECCDH} = \xi.(1-\frac{\mathcal{Q}_{Csv}}{\mathcal{Q}_{Ha}}).(1-\frac{\mathcal{Q}_{CFPKG}}{\mathcal{Q}_{Ha}}).(\frac{1}{(\mathcal{Q}_{Ha}-\mathcal{Q}_{Csv}-\mathcal{Q}_{CFPKG})})$.

C. INFORMAL SECURITY ANALYSIS

1) REPLAY ATTACK

In the proposed scheme, the attacker may not give response to old messages. The scheme privileges replay attack resistance by offering renewal of key and nonce (Non) in each session i.e. $\mathcal{C} = H_{\mathrm{b}}(\Omega\|\mathcal{R}\|\mathcal{F}\|\ \mathrm{ID}_{\mathrm{dn}}) \oplus m$ where $m = (\mathcal{S}\mathcal{K}\|m\|t\|\ \mathrm{Non}\ \|\ \mathrm{ID}_{\mathrm{en}})$.

In case an attacker intrudes the message of one session, then he/she cannot infiltrate the messages of other sessions with the same key, because the session key and nonce is renewed. Nonce (Non), as a property, refreshes itself at each instance. The receiver is required to perform an up-to-dateness check with every message and in case an outdatedness is detected the message is trashed to the black box.

2) INTEGRITY

The sender takes "hash value" of the key before sending the message i.e. $H_{\mathrm{c}}(\Omega\|\mathcal{R}\|m\|\mathrm{ID}_{\mathrm{en}}\|\delta_{\mathrm{en}})$ or $H_{\mathrm{c}}(\Omega\|\mathcal{R}\|m\|\ \mathrm{ID}_{n}\mathrm{en}\ \|\beta_{\mathrm{en}})$. When an attacker wants to do a change from $\mathcal{C}$ to $\mathcal{C}'$ then it is indispensable for him to convert m to m' and H_{c} to H'_{c}. The 'hash' exhibits a property of being an irreversible function. Firstly, the sender counts 'hash' of the key and the arbitrary tag 't', which decides the validity of the encapsulation. At receiver to check the integrity. Attacker cannot generate hash.

3) FORWARD SECRECY

The proposed scheme offers forward secrecy. Every session completion process follows the renewal of the sender's secret key. Here, the adversary is not able to read signcrypted

TABLE 6. Computational cost.

Schemes	Secret key generation	Encapsulation	De-Encapsulation	Total
Seo et al [33]	4 em	2 em	7 em	13 em
Liu et al [34]	3 em	1 em	3 em	7 em
Zhou et al [20]	11 em	7 em	8 em	26 em
Reddy et al [35]	2 em	1 em	4 em	7 em
Xiong et al [36]	6 em	1 em	6 em	13 em
Won et al [28]	3 em	3em	3em	9 em
Proposed	3 $\hbar m$	1 $\hbar m$	1 $\hbar m$	5 $\hbar m$

TABLE 7. Computational cost in milliseconds.

Schemes	Secret key generation	Encapsulation	De-Encapsulation	Total
Seo et al [33]	3.88 ms	1.94 ms	6.79 ms	12.61 ms
Liu et al [34]	2.91 ms	0.97 ms	2.91 ms	6.79 ms
Zhou et al [20]	10.67 ms	6.79 ms	7.76 ms	25.22 ms
Reddy et al [35]	1.94 ms	0.97 ms	3.88 ms	6.79 ms
Xiong et al [36]	5.28 ms	0.97 ms	5.28 ms	12.67 ms
Won et al [28]	2.91 ms	2.91 ms	2.91 ms	8.73 ms
Proposed	1.44 ms	0.48 ms	0.48 ms	2.40 ms

messages and therefore, session messages cannot be recovered. Moreover, above all, a secret key is regenerated in each session.

VII. PERFORMANCE COMPARISON

This section is dedicated to compare the performance of our proposed scheme with the schemes proposed by Seo *et al.* [33], Liu *et al.* [34], Zhou *et al.* [20], Reddy *et al.* [35], Xiong *et al.* [36] and Won *et al.* [28].

A. COMPUTATIONAL COST

In Table 6, the proposed scheme is compared, in terms of computational cost, with the existing ones, i.e. as Seo *et al.* [33], Liu *et al.* [34], Zhou et al. [20], Reddy *et al.* [35], Xiong *et al.* [36] and Won *et al.* [28]. We consider hyperelliptic divisor multiplication as the elliptic curve scalar multiplication has been found to be the most expensive operation in the related existing schemes. From the computational costs observed in Table 7, it is clearly evident that our scheme outperforms as compared to the schemes presented by Seo *et al.* [33], Liu *et al.* [34], Zhou *et al.* [20], Reddy et al. [35], Xiong *et al.* [36], and Won *et al.* [28]. In Table 6, the variables $\hbar m$ and em denote hyperelliptic curve divisor multiplication and elliptic curve scalar multiplication respectively. It has been observed that a single scalar multiplication takes 0.97 milliseconds for Elliptic Curve Point Multiplication (ECPM). In order to estimate the performance of the proposed approach, the Multi-precision Integer and Rational Arithmetic C Library (MIRACL) [57] is used to test the runtime of the basic cryptographic operations up to 1000 times. The observance is made on a workstation having following specifications: Intel Core i7- 4510U CPU @ 2.0 GHz, 8 GB RAM and Windows 7 Home Basic 64-bit Operating System [33]. Similarly, the Hyperelliptic Curve Divisor Multiplication (HCDM) is assumed to be 0.48 milliseconds due to the smaller key size i.e. 80-bits key size, as opposed to elliptic curve that is sized 160 bits [58].

Our proposed scheme proves to be quicker than the schemes presented by Seo *et al.* [33], Liu *et al.* [34], Zhou *et al.* [20], Reddy *et al.* [35], Xiong et al. [36] and Won *et al.* [28] by 80.96%, 64.65%, 90.48 %, 64.65%, 81.05% and 72.50 % respectively.

B. COMMUNICATION COST

Table 8 summarizes the comparison between the proposed scheme and the major existing schemes, presented by Seo *et al.* [33], Liu *et al.* [34], Zhou *et al.* [20],

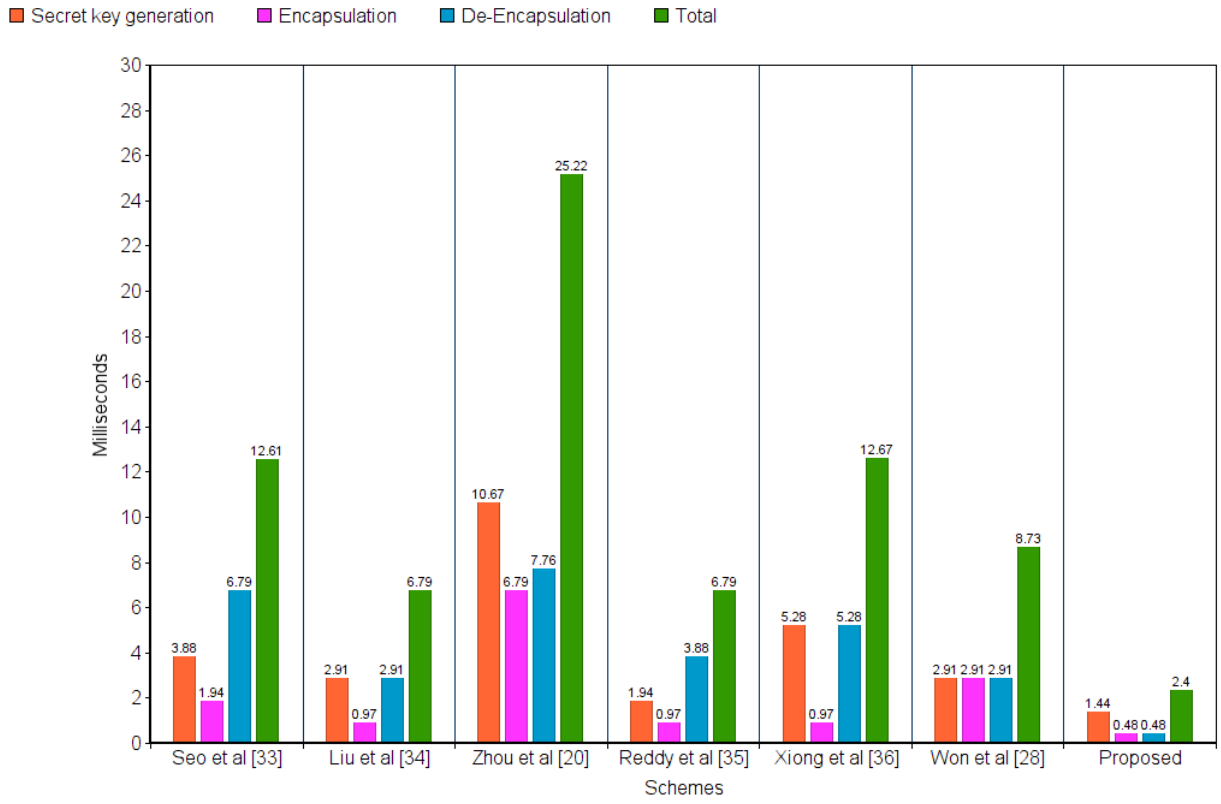


FIGURE 7. Computational cost.

TABLE 8. Communication cost.

Schemes	Signcrypted text	Communication cost in bits
Seo et al [33]	3|q|+ |m|	1504
Liu et al [34]	3|q|+ |m|	1504
Zhou et al [20]	3|q|+ |m|	1504
Reddy et al [35]	3|q|+ |m|	1504
Xiong et al [36]	3|q|+ |m|	1504
Won et al [28]	3|q|+ |m|	1504
Proposed	$\|m\|$+3|n|	1264

Reddy *et al.* [35], Xiong *et al.* [36] and Won *et al.* [28]. The variables involved have following assumed values: $|q| \cong 160$ bits for elliptic curve; $|n| \cong 80$ bits for hyperelliptic curve; and $|m| = 1024$ bits for message. The communication cost incurred for the existing schemes, i.e. Seo *et al.* [33], Liu *et al.* [34], Zhou *et al.* [20], Reddy *et al.* [35], Xiong et al. [36] and Won *et al.* [28], is 1504 bits ($3|q| + |m| = 1504$) as all the six schemes use elliptic curve. On the other hand, the communication cost for the proposed scheme, that uses hyperelliptic curve, is 1264 bits ($|m| + 3|n| = 1264$). Moreover, it has been concluded that, as compared to the existing schemes, in the proposed scheme the speed response experiences a boost of 15.95%.

C. SECURITY FUNCTIONALITIES

Table 9 presents a brief comparison between the proposed scheme and major existing schemes in term of security functionality. It is worth noting, from Table 9, that

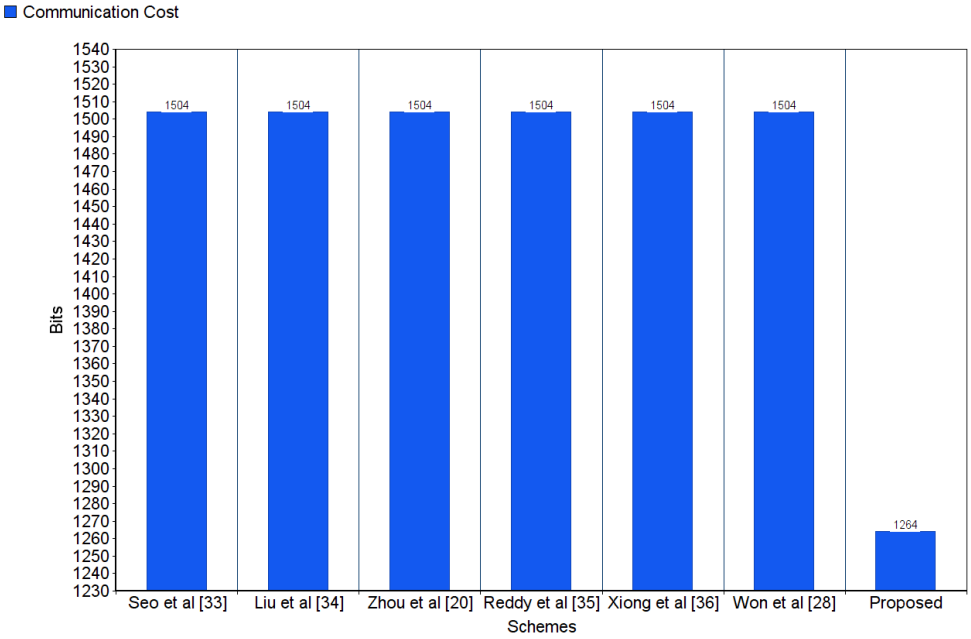


FIGURE 8. Communication cost.

TABLE 9. Comparison with relevant existing schemes. Legend: U: unforgeability, I: integrity, C: confidentiality, FS: forward secrecy, RA: replay attack, FA: Formal analysis.

Schemes	Security functionalities					
	Informal					Formal
	U	I	C	FS	RA	FA
Seo et al [33]	✓	×	✓	×	✓	×
Liu et al [34]	×	×	✓	×	×	×
Zhou et al [20]	✓	×	✓	×	✓	×
Reddy et al [35]	✓	×	×	×	✓	×
Xiong et al [36]	✓	×	×	×	✓	×
Won et al [28]	✓	✓	✓	×	✓	×
Proposed	✓	✓	✓	✓	✓	✓

the related schemes are not validated through formal security validation tools, such as AVISPA, and none of them guarantee Forward Secrecy (FS) and Replay Attack (RA).

VIII. APPLICATION SCENARIO

A. PRECISION AGRICULTURE

The proposed scheme is evaluated for application scenario i.e precision agriculture that involves monitoring of crop health in a cultivated field as illustrated in Fig 2. The high-resolution images of crops are obtained with the help of air-borne platforms (i.e. UAVs). The images are, then, processed to extract information that can be used to provide future decisions. Therefore, in the proposed system, crop health is monitored using the data collected from the Normalized Difference Vegetation Index (NDVI) mapping of spectral images. The images are captured by a multi-spectral camera mounted on M-UAVs. The NDVIs are computed to differentiate healthy plants from the unhealthy ones. This is done by measuring the chlorophyll content in the crops. The information is further used to localize the area under stress precisely. The M-UAVs capture and transmit the images to the linked B-UAV. Upon receiving the images, the on-board microcontroller on the B-UAV generates the tasks. The Decision Support Engine (DSE), or the local microcomputer, then, processes the tasks.

Here, it is pertinent to mention that the M-UAVs can be accoutred with relevant accessories, such as cameras, IMU, sensors and GPS unit etc, to cater to a wide range of customized tasks.

IX. CONCLUSION

Flying Ad-hoc Network (FANET) is an emerging technology for uniting small UAVs. It involves analyzing the continuously evolving data from heterogeneous sources for creating a new era of real-life applications. However, the participating UAVs in FANET are usually resource-constrained, which makes them luring targets for cyber-attacks. To address this challenge, in this paper, we propose a Certificateless Key-Encapsulated Signcryption (CL-KESC) scheme. Unfortunately, the existing construction models of CL-KESC rely on the use of elliptic curve-based operations, which are computationally expensive for small UAVs. Therefore, in this paper, we presented a new construction scheme of CL-KESC based on hyperelliptic curve, an advanced version of elliptic curve characterized

by a small parameter and key size (80 bits) as compared to the elliptic curve, where key size is 160 bits. A security analysis, including the formal security verification, is performed using the widely-recognized AVISPA tool and, in the findings, our proposed scheme proves to offer significant immunity against adversely attacks. To further complement the pros, the presented scheme, in addition, is far more computationally efficient.

REFERENCES

[1] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, May 2013.

[2] W. Zafar and B. M. Khan, "Flying ad-hoc networks: Technological and social implications," *IEEE Technol. Soc. Mag.*, vol. 35, no. 2, pp. 67–74, Jun. 2016.

[3] M. A. Khan, I. M. Qureshi, and F. Khanzada, "A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (FANET)," *Drones*, vol. 3, no. 1, p. 16, Feb. 2019.

[4] A. Abdallah, M. Ali, J. Mišić, and V. Mišić, "Efficient security scheme for disaster surveillance UAV communication networks," *Information*, vol. 10, no. 2, p. 43, Jan. 2019.

[5] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption)," in *Proc. Annu. Int. Cryptol. Conf.* Springer, 1997, pp. 165–179.

[6] A. W. Dent, "Hybrid signcryption schemes with outsider security," in *Proc. 8th Int. Conf. Inf. Secur.* (Lecture Notes in Computer Science). Springer-Verlag, 2005, pp. 203–217.

[7] Z. Chen, S. Chen, H. Xu, and B. Hu, "A security scheme of 5G ultradense network based on the implicit certificate," *Wireless Commun. Mobile Comput.*, vol. 2018, May 2018, Art. no. 8562904.

[8] C. K. N, A. Basit, P. Singh, and V. V. Ch, "Lightweight cryptography for distributed PKI based MANETS," *Int. J. Comput. Netw. Commun.*, vol. 10, no. 2, pp. 69–83, Mar. 2018.

[9] S. Ullah, L. Marcenaro, and B. Rinner, "Secure smart cameras by aggregate-signcryption with decryption fairness for multi-receiver IoT applications," *Sensors*, vol. 19, no. 2, p. 327, 2019.

[10] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 1984, pp. 19–23.

[11] Y. Zhan and B. Wang, "Cryptanalysis of a certificateless aggregate signature scheme for healthcare wireless sensor network," *Secur. Commun. Netw.*, vol. 2019, pp. 1–5, Jun. 2019.

[12] P. Kumar, S. Kumari, V. Sharma, X. Li, A. K. Sangaiah, and S. H. Islam, "Secure CLS and CL-AS schemes designed for VANETs," *J. Supercomput.*, vol. 75, no. 6, pp. 3076–3098, Jun. 2019.

[13] F.-T. Zhang, Y.-X. Sun, L. Zhang, M.-M. Geng, and S.-J. Li, "Research on certificateless public key cryptography," *J. Softw.*, vol. 22, no. 6, pp. 1316–1332, Jun. 2011.

[14] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proc. ASICC*, New York, NY, USA, 2008, pp. 18–20.

[15] G. Lippold, C. Boyd, and J. M. González Nieto, "Efficient certificateless KEM in the standard model," in *Proc. 12th Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer-Verlag, 2010, pp. 34–46.

[16] F. Li, M. Shirase, and T. Takagi, "Certificateless hybrid signcryption," in *Information Security Practice and Experience* (Lecture Notes in Computer Science), vol 5451. Berlin, Germany: Springer, 2009, pp. 112–123.

[17] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, "A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices," *Sensors*, vol. 18, no. 11, p. 3868, Nov. 2018.

[18] M. Yu, J. Zhang, J. Wang, J. Gao, T. Xu, R. Deng, Y. Zhang, and R. Yu, "Internet of Things security and privacy-preserving method through nodes differentiation, concrete cluster centers, multi-signature, and blockchain," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 12, Dec. 2018, Art. no. 155014771881584.

[19] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, Aug. 2018.

[20] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Secur. Commun. Netw.*, vol. 2017, Aug. 2017, Art. no. 8405879.

[21] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *J. Supercomput.*, vol. 74, no. 12, pp. 6428–6453, Dec. 2018.

[22] A. A. Omala, A. S. Mbandu, K. D. Mutiria, C. Jin, and F. Li, "Provably secure heterogeneous access control scheme for wireless body area network," *J. Med. Syst.*, vol. 42, no. 6, p. 108, Jun. 2018.

[23] C. Tamizhselvan and V. Vijayalakshmi, "An energy efficient secure distributed naming service for IoT," *Int. J. Adv. Stud. Sci. Res.*, vol. 3, no. 8, pp. 6–10, 2019.

[24] V. S. Naresh, R. Sivaranjani, and N. V. E. S. Murthy, "Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 31, no. 15, p. e3763, Oct. 2018.

[25] A. U. Rahman, I. Ullah, M. Naeem, R. Anwar, N.-U.-A., H. Khattak, and S. Ullah, "A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve," *ijacsa*, vol. 9, no. 5, pp. 160–167, Jun. 2018.

[26] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive security for multilevel ad hoc networks," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 533–547, 2002.

[27] S. He, Q. Wu, J. Liu, W. Hu, B. Qin, and Y.-N. Li, "Secure communications in unmanned aerial vehicle network," in *Proc. Int. Conf. Inf. Security Pract. Exper.* Springer, vol. 2017, pp. 601–620.

[28] J. Won, S.-H. Seo, and E. Bertino, "Certificateless cryptographic protocols for efficient drone-based smart city applications," *IEEE Access*, vol. 5, pp. 3721–3749, 2017.

[29] J. Won, S.-H. Seo, and E. Bertino, "A secure communication protocol for drones and smart objects," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur. (ASIA CCS)*, 2015, pp. 249–260.

[30] B. Semal, K. Markantonakis, and R. N. Akram, "A certificateless group authenticated key agreement protocol for secure communication in untrusted UAV networks," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, London. U.K., Sep. 2018, pp. 1–8.

[31] E. Barka, C. Kerrache, R. Hussain, N. Lagraa, A. Lakas, and S. Bouk, "A trusted lightweight communication strategy for flying named data networking," *Sensors*, vol. 18, no. 8, p. 2683, Aug. 2018.

[32] M. Bae and H. Kim, "Authentication and delegation for operating a multi-drone system," *Sensors*, vol. 19, no. 9, p. 2066, May 2019.

[33] S.-H. Seo, J. Won, and E. Bertino, "pCLSC-TKEM: A pairing-free certificateless signcryption-tag key encapsulation mechanism for a privacy-preserving IoT," *Trans. Data Privacy*, vol. 9, no. 2, pp. 101–130, Aug. 2016.

[34] W. Liu, M. A. Strangio, and S. Wang, "Efficient certificateless signcryption tag-KEMs for resource constrained devices," Ph.D. dissertation, Cornell Univ., Ithaca, NY, USA, 2015.

[35] P. V. Reddy, A. R. Babu, and N. B. Gayathri, "Efficient and secure Identity-based Strong Key-insulated Signature scheme without pairings," *J. King Saud Univ.-Comput. Inf. Sci.*, to be published.

[36] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Syst. J.*, to be published.

[37] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, p. 177, pp. 203–209, 1987.

[38] (2019). *Hyperelliptic Curve*. [Online]. Available: https://en.wikipedia.org/wiki/Hyperelliptic_curve

[39] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar, "Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Springer, 2003, pp. 351–365.

[40] D. G. Cantor, "Computing in the Jacobian of a hyperelliptic curve," *Math. Comp.*, vol. 48, no. 177, p. 95, Jan. 1987.

[41] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, "Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols," in *Proc. 1st Int. Conf. Latest trends Electr. Eng. Comput. Technol. (INTELLECT)*, Nov. 2017, pp. 1–9.

[42] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[43] (2019). *AVISPA. Automated Validation of Internet Security Protocols and Applications*. [Online]. Available: http://www.avispa-project.org/

[44] (2019). *SPAN: A Security Protocol Animator for AVISPA*. [Online]. Available: http://www.avispa-project.org/

[45] D. V. Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. APPSEM Workshop*, Tallinn, Finland, Sep. 2005, pp. 1–2.

[46] D. Basin, S. Mödersheim, and L. Viganò, "OFMC: A symbolic model checker for security protocols," *Int. J. Inf. Secur.*, vol. 4, pp. 181–208, Jun. 2005.

[47] M. Turuani, "The CL-Atse protocol analyser," in *Proc. Int. Conf. Rewriting Techn. Appl.*, Seattle, WA, USA, Aug. 2006, pp. 227–286.

[48] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, "Secure authentication protocol for wireless sensor networks in vehicular communications," *Sensors*, vol. 18, no. 10, p. 3191, Sep. 2018.

[49] K. Park, Y. Park, Y. Park, A. Goutham Reddy, and A. K. Das, "Provably secure and efficient authentication protocol for roaming service in global mobility networks," *IEEE Access*, vol. 5, pp. 25110–25125, 2017.

[50] V. Odelu, A. K. Das, K.-K.-R. Choo, N. Kumar, and Y. Park, "Efficient and secure time-key based single sign-on authentication for mobile devices," *IEEE Access*, vol. 5, pp. 27707–27721, 2017.

[51] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Gener. Comput. Syst.*, vol. 68, pp. 74–88, Mar. 2017.

[52] K. Park, Y. Park, Y. Park, and A. K. Das, "2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment," *IEEE Access*, vol. 6, pp. 30225–30241, 2018.

[53] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, N. Kumar, Y. Park, and S. Tanwar, "Design of an anonymity-preserving group formation based authentication protocol in global mobility networks," *IEEE Access*, vol. 6, pp. 20673–20693, 2018.

[54] (2019). *AVISPA v1.1 User Manual*. [Online]. Available: http://www.avispa-project.org/package/user-manual.pdf

[55] B. Zhang, Z. Jia, and C. Zhao, "An efficient Certificateless generalized Signcryption scheme," *Secur. Commun. Netw.*, vol. 2018, May 2018, Art. no. 3578942.

[56] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, Jun. 2000.

[57] *Shamus Sofware Ltd. Miracl Library*. Accessed: Aug. 21, 2019. [Online]. Available: http://github.com/miracl/MIRACL

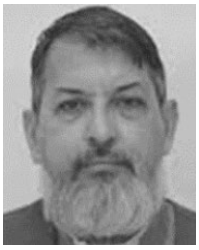
[58] Ullah, Amin, Naeem, Khattak, and Ali, "A Novel provable secured signcryption scheme PSSS: A hyper-elliptic curve-based approach," *Mathematics*, vol. 7, no. 8, p. 686, Jul. 2019.

MUHAMMAD ASGHAR KHAN received the bachelor's degree in electronic engineering from Iqra University, Karachi, Pakistan, and the master's degree in electrical engineering from the Center of Advanced Studies in Engineering (CASE), Islamabad, Pakistan. He is currently pursuing the Ph.D. degree in electronic engineering with the School of Engineering and Applied Sciences (SEAS), ISRA University, Islamabad. He is also serving as a Lecturer with the Department of Electrical Engineering, Hamdard University, Islamabad. His research interests include UAVs/Drones with a focus on networks, platforms, security, as well as applications and services.

INSAF ULLAH received the master's degree in computer sciences from the Department of Information Technology, Hazara University Manshera, Pakistan, where he is currently pursuing the Ph.D. degree in computer sciences. He is also serving as a Lecturer with the Department of Computer Sciences, Hamdard University, Islamabad. Until now, he has published 23 articles in different journals and conferences. His research interest includes network security.

SHIBLI NISAR is currently an Assistant Professor with the Department of Electrical Engineering, National University of Sciences and Technology (NUST), Pakistan. He has attended and presented his research papers in various national and international conferences. His research interests include signal processing, speech processing, mathematical modeling, designing, and analysis of wireless ad-hoc and sensor networks and machine learning. He is the author/coauthor of 18 research papers, including impact factor journals and peer-reviewed international and local conferences.

FAZAL NOOR received the M.Eng. and B.Eng. degrees from Concordia University, in 1986 and 1984, respectively, and the Ph.D. degree in engineering from McGill University, Canada, in 1993. He is currently working as an Associate Professor with the Faculty of Computer Science and Information Systems, Islamic University of Madinah. He has numerous publications in international conferences and journals. His current research interests are in network security, parallel and distributed computing, embedded systems, the IoT, robotics, and computer vision.



IJAZ MANSOOR QURESHI received the bachelor's degree in avionic engineering from the NED University of Engineering and Technology, Karachi, Pakistan, the master's degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, and the Ph.D. degree in high energy physics from the University of Toronto, ON, Canada. He has to his credit a post-PhD experience stretching 27 years in various Pakistani higher education institutes of repute. He is currently with the Electrical Engineering Department, Air University, as a Professor. He has supervised about 37 Ph.D. thesis so far. His research interests include digital/wireless communications, digital signal processing, information security, soft computing, and evolutionary computing.

FAHIM ULLAH KHANZADA received the bachelor's degree in electronic engineering from the Balochistan University of Information Technology, Engineering and Management Sciences (BUITEMS), Quetta, and the master's degree in electrical engineering from the University of Nottingham, Nottingham, U.K. He is currently associated with Descon Engineering Limited, Lahore, Pakistan. His experience encompasses academia, industry, and standardization.

NOOR UL AMIN received the master's degree in computer science from the University of Peshawar, Pakistan, in 1996, and the Ph.D. degree in computer science from the Department of Information Technology, Hazara University, Pakistan, where he has been the Head of the Department of Information Technology and the Director IT for 11 years. He is currently the Chair of the Department of Telecommunication, Hazara University. He has completed recently an Research and Development project sponsored by the Ministry of Science and Technology, Pakistan, and established 07 hi-tech research and development labs. His research interests are in the areas of information security, mobile adhoc networks (MANETs), wireless sensor networks (WSNs), and information-centric networking (ICN).