information

MDPI

Article

Efficient Security Scheme for Disaster Surveillance UAV Communication Networks

Asmaa Abdallah, M. Zulfiker Ali, Jelena Mišić * and Vojislav B. Mišić

Department of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada; arabdal@ryerson.ca (A.A.); mzulfiker.ali@ryerson.ca (M.Z.A.); vmisic@ryerson.ca (V.B.M.)
* Correspondence: jmisic@ryerson.ca; Tel.: +1-416-979-5000

Received: 4 October 2018; Accepted: 24 January 2019; Published: 29 January 2019

check for updates

Abstract: The Unmanned Aerial Vehicles (UAVs) play a significant role to alleviate the negative impacts of disasters by providing essential assistance to the rescue and evacuation operations in the affected areas. Then, the reliability of UAV connections and the accuracy of exchanged information are critical parameters. In this paper, we propose networking and security architecture for disaster surveillance UAV system. The networking scheme involves a two-tier cluster network based on IEEE 802.11ah, which can provide traffic isolation between the tiers. The security scheme guarantees the accuracy and availability of the collected information from the disaster area applying fingerprint features and data redundancy techniques; the proposed scheme also utilizes the lightweight Ring-Learning with Errors (Ring-LWE) crypto-system to assure the confidentiality of the transmitted data with low overhead.

Keywords: UAVs; ring-LWE crypto-system; IEEE 802.11ah; drones

1. Introduction

Unmanned Aerial Vehicles (UAVs) [1,2] have many military and civilian applications, such as border surveillance, public safety and transportation management. Mainly, UAVs are crucial for rescue and recovery operations during disasters, such as volcanoes and earthquakes, especially when the regular communication networks in the area are partially or completely destroyed. In addition to collecting information about the disaster, UAVs can carry equipment, i.e., medical aids, to the disaster area without involving humans. Other applications for UAV networks could be monitoring the effect of rocket launch operation on the surrounding area [3], preserving public safety during terrorist attacks or natural disasters using 4G [4] or satellite communications [5], prompting the smart farming [6], or utilized in IoT aerial sensing [7].

The UAV system could consist of one large drone, as is often the case in military missions, or a group of small drones that are usually cooperating to complete one task, such as traffic monitoring or forest surveillance. Using a swarm of small drones instead of single large drone brings higher system reliability and scalability, but it also introduces additional issues related to management of system components and functions; in particular, it requires the following:

1. Central ground substation (CS) which collects data from relaying drones.
2. Clusters of drones participating in the same application that perform some kind of surveillance/sensing application.
3. Cluster head drone(s) in each cluster have two main functions. Firstly, they keep administrative membership data for the cluster. Secondly, they collect sensed data perform data fusion, integrity and confidentiality. Due to a large distance between cluster head(s) and central substation (CS), cluster head(s) do not send data directly to it. Instead, cluster head(s) transmits the data to relay

drones. Please note that it is also possible that each node in the cluster can communicate with the relay node and that a single or a few nodes keep administrative functions.
4. Relay drones which need to leave the cluster and return to the ground station in order to re-charge the batteries. Relay drones collect the data from the cluster head and hand them over to the central substation.

Usually, drones have small batteries that could last for 20 to 30 min only, so that it is crucial to have communications over short distances, small contention in access protocol and a lightweight security scheme. The UAV network should be self-forming and reorganizing, in addition, fault and delay tolerant [1,2]. In this work, we propose a secure disaster surveillance UAV system built over drones that communicate using IEEE 802.11ah standard. This technology features multiple Restricted Access Windows (RAW) which can be used to isolate intra-cluster from cluster to relay communications [8,9]. In combination with multiple physical channels and higher communication range, this technology is a serious candidate for reliable multi-cluster drone networks.

Specific characteristics for UAV networks should be taken into consideration when designing a secure UAV system. First, the drone's limited energy (because of its bounded battery size) restricts the flying time to a specified period. Second, the drone's computation capability does not permit the performance of complex cryptographic operations. Our security scheme uses the lightweight Ring-Learning with Errors (Ring-LWE) crypto-system to protect the exchanged messages' confidentiality with low computation overhead. It also utilizes data redundancy and fingerprint features techniques.

The remainder of the paper is organized as follows: Section 2 discusses related work related to drone networks from the networking, security and application aspects. Section 3 introduces our system model, security parameters, and design goals. Section 4 reviews the Ring-LWE Encryption Scheme. In Section 5, we present our proposed security scheme. Section 6 gives the security analysis, while Section 7 evaluates the performance of our scheme. Finally, Section 8 concludes the paper.

2. Related Work

The UAV systems face many communication challenges due to the drones' intermittent connections so that it is hard to maintain an end-to-end path to destination. Consequently, the UAV network should be a disruption tolerant; information should reach the destination even after some delay and even if the created path to the destination has some fluctuations [10]. The UAVs can be utilized as micro-scale mobile relays to enhance the cellular network performance [11] or as unmanned aerial base stations in mission-critical public safety communications [12]. A simple short horizon algorithm [13] plans dedicated paths for UAVs so that all locations are scanned frequently, at the same time, the optimal UAV speed to achieve the best network throughput and energy constraints is preserved. A fly-hover-and-communicate protocol [14] partitions the ground terminals into disjoint clusters with directional antenna UAV that hovers above the cluster center. The quality of the communication link between UAV and ground nodes is impacted by frequent occlusions in the urban environment so that Ref. [15] combines a Gaussian Process learning approach with relay trajectory planner to predict the strength of the UAV communication relay missions. In Ref. [16], Huo et al. propose a distributed multi-layer UAV (DAMU) 5G wireless network considering different types of UAV designs and the corresponding 5G application scenarios. In Ref. [17], Lin et al. study the LTE connectivity for low altitude small UAVs; it offers wide-area high speed, and secure wireless connectivity, which can enhance control and safety of UAV operations and enable beyond visual line-of-sight (LOS) use cases. In Ref. [18], UAV communications utilize proactive caching of the ground nodes to store the collected readings which can be retrieved anytime from the node's local cache or its nearest neighbor via device-to-device (D2D) communications. An efficient cell-based allocations approach [19] provides the optimal UAV positioning for full coverage in 5G networks while optimizing the throughput coverage.

However, cellular network connectivity can not be guaranteed in disaster areas and it also comes with high cost. For that reason, Wi-Fi technology deserves to be considered as well. Wi-Fi can provide low cost and high performance user interface [20]. On the other hand, classical Wi-Fi suffers from high contention and short-range communications which result in high multi-hop latency [21] and opens avenue for synchronization attacks [22]. Fortunately, recent standard IEEE 802.11ah extends the transmission range of Wi-Fi and offers support for relaying [8,9].

Only few research works discuss the security aspects of UAV networks. In Ref. [23], Altawy and Youssef introduce various security and privacy concerns for UAV systems. According to Ref. [24], the authors propose a physical layer security solution to protect the UAV communication systems against eavesdropping and malicious jamming. The solution is based on combining artificial noise with the transmitted information to protect the transmitted data and to confuse the attacker. An inferred context-free grammar method [25] validates the UAV commands by comparing them with the assigned command's format, and consequently alleviates the impact of jamming and hijacking attacks. The coagulation attack that attempts to fully control the UAV by alternating the UAV physical configurations, modifying its waypoint to cause collision, or UAV hijacking is introduced in [26]. While Sharma et al. [27] present the securing context information scheme that supports the 3D localization of drones in urban scenarios.

3. System Model

In this section, we present networking and application models which are depicted in Figure 1.

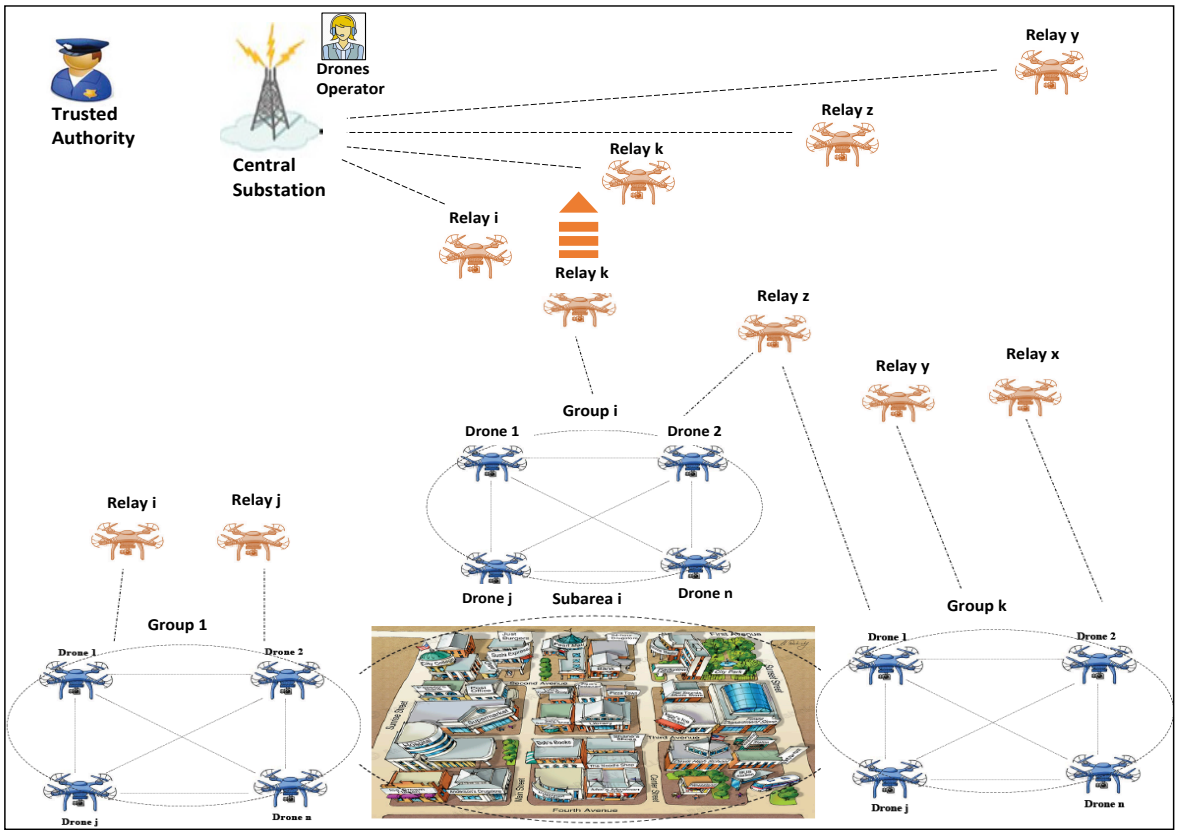


Figure 1. System model.

3.1. Network Model Using IEEE 802.11ah

All drone UAV nodes including CS and drones communicate using IEEE 802.11ah as medium access control protocol (MAC). Sensing nodes in the cluster comprise the second networking tier. Relay

nodes can directly communicate with CS and they comprise first communication tier. Cluster head nodes then only keep administrative data while each sensing drone can communicate with the relay drone. Each sensing (non-relaying) drone is associated with single relay node. All packets received from non-relay nodes are forwarded to CS. Similarly, the response message transmitted by CS are received by the relay node and forwarded to the non-relay nodes.

In our framework, the UAV network uses Restricted Access Window (RAW) scheme where beacon interval is divided into separate RAW slots to implement the uplink channel access to a relay and non-relay nodes. Relay nodes use a dedicated RAW slot to communicate with CS. Additional isolation among different clusters can be also achieved using RAW slots. During network initialization, the initial relay nodes in tier one are configured first. In addition, in initialization, each non-relay drone is first associated with CS. As the non-relay drones fly away from the transmission range of CS, the non-relay nodes get re-associated with a relay node in order to achieve two hop communication with CS. In steady state operation, tier two drones form a cluster where each node works as an administrative cluster head and thus increase the scalability and lifetime of the network. When tier two node's energy level drops below a certain threshold, this node moves towards CS and takes on the relay function. We assume that the re-charging process on the ground ensures a steady supply of the nodes in tier two and tier one. The selection of new cluster head node from the cluster of non-relay nodes may be based on available energy level of the non-relay nodes.

CS will allocate a RAW slot to a relay node and broadcast the allocation using an S1G beacon (DTIM) frame. The DTIM beacon frame contains the allocation of which slot is allocated to which relay node and the start time of each RAW slot. At the beginning of an allocated RAW slot, a relay node will broadcast a TIM beacon frame which contains the bitmap for the nodes that have packets for downlink direction. The TIM beacon also carries the information of the duration for which the non-relay nodes are allowed to contend for medium access. At the end of transmission or the expiry of RAW duration, non-relay nodes go to sleep mode and wake up at the start of the next TIM beacon transmission time. However, the relay node goes to sleep mode only at the expiry of RAW duration and wakes up at the next beacon transmit time.

3.2. UAV Application Model

Our application model divides the disaster area into n subareas $\mathbb{A} = \{\mathbb{A}_1, \mathbb{A}_2, \ldots, \mathbb{A}_n\}$. Each subarea is monitored by the number of air drones $D = \{D_1, D_2, \ldots, D_m\}$, where m is the total number of drones in a specific subarea. The monitoring drones in each subarea are divided by the central substation CS into separate groups $G = \{G_1, G_2, \ldots, G_k\}$, where k is the number of the groups in a certain subarea; k is varied according to the size and importance of the subarea. The number of drones per group G_i is varied as the drones can enter/leave the group anytime; $|G_1| + |G_2| + \cdots + |G_k| \leq |D|$; where $|D| = m$. The drones collect information about the assigned subarea and forward their readings to CS via β cluster heads; selecting the cluster heads depends on various parameters, such as the current number of the drones in the group, the quality of the transmission channels, and the surrounding environment. The cluster heads are regular drones chosen to forward the readings of the whole group to CS. The readings are transmitted from cluster heads to CS via the drones that are flying back to the station to recharge their batteries; these drones work as relay nodes. $R = \{R_1, R_2, \ldots, R_j\}$, where j is the current number of relays in the subarea. Finally, the model has a trusted authority (TA) that is providing the keying parameters for different parties in the connection. Figure 1 shows the network model.

3.3. Adversary Model

In the disaster area, malicious adversaries can threaten the integrity and availability of the monitoring information collected by drones and then negatively impact the efficiency of the rescue and evacuation operations in the area. The CS is a trusted party; it is located in a protected place. It will not attempt to falsify the received information. Drones are non-trusted parties because of their location

in the air; they are prone to be compromised by adversaries or malfunctioned because of the hostile surrounding environment. The attacker $\mathscr{A}$ can compromise and impersonate D_s, and also intercept and block their messages. In addition, $\mathscr{A}$ can begin a replay attack, or attempt to forge the transmitted messages. However, $\mathscr{A}$ has limited resources; he/she cannot compromise all drones in the subarea; only a limited number of them.

3.4. Security Requirements and Design Goals

The proposed scheme aims to prevent the negative influence of malicious $\mathscr{A}$ and hostile environment on the rescue and evacuation operation by guaranteeing the collected information's integrity and availability; it should prevent delaying or blocking the messages. Furthermore, the proposed scheme has to be lightweight in terms of communication and computation complexity because of the limited-battery drones.

4. Preliminaries of the Security Protocol

4.1. The Ring-LWE Encryption Scheme

The Ring-LWE based encryption scheme is a lattice-based crypto-system that exploits the learning with errors (LWE) problem, which is to distinguish random linear equations that have been perturbed by a small amount of noise from truly uniform ones. The LWE problem has been proven to be as hard as worst-case lattice problems and are considered to be secure against post-quantum attacks [28,29].

4.1.1. The Ring-LWE Problem

Two polynomial a and s are chosen uniformly from the polynomial ring $R_q = Z_q[x]/\langle f \rangle$, where f is an irreducible polynomial of degree $n-1$. An error polynomial e of degree n is sampled from am error distribution $\mathcal{X}$, which is a discrete Gaussian distribution $\mathcal{X}_\sigma$ with standard deviation σ. The Ring-LWE distribution $A_{s,\mathcal{X}}$ over $R_q \times R_q$ consists of tuples (a,t), where $t = a.s + e \in R_q$. Given polynomial pairs (a,t) from $A_{s,\mathcal{X}}$, it is very difficult to find s. This problem is known as the search ring-LWE problem [30]. This paper utilizes the efficient version of the ring-LWE based crypto-system [31] that minimizes the computation overhead of the encryption scheme.

4.1.2. Key Generation

Two polynomials r_1 and r_2 are sampled from $\mathcal{X}_\sigma$ using a discrete Gaussian sampler. Then, compute

$$\tilde{r}_1 \leftarrow NTT(r_1), \tilde{r}_2 \leftarrow NTT(r_2), \tilde{p} \leftarrow \tilde{r}_1 - \tilde{a} * \tilde{r}_2,$$

where the Number Theoretic Transform (NTT) corresponds to the Fast Fourier Transform (FFT) when the primitive roots of unity are from a finite ring of integers instead of complex numbers.

The private key is $\tilde{r}_2$ and the public key is $(\tilde{a}, \tilde{p})$.

4.1.3. Encryption

The message m is encoded to a polynomial $\bar{m} \in R_q$. Error polynomials $e_1, e_2, e_3 \in R_q$ are generated from $\mathcal{X}_\sigma$ using a discrete Gaussian sampler. Then, compute the ciphertext $(\tilde{c}_1, \tilde{c}_2)$:

$$\tilde{e}_1 \leftarrow NTT(e_1); \tilde{e}_2 \leftarrow NTT(e_2),$$

$$(\tilde{c}_1, \tilde{c}_2) \leftarrow (\tilde{a} * \tilde{e}_1 + \tilde{e}_2, \tilde{p} * \tilde{e}_1 + NTT(e_3 + \bar{m}).$$

4.1.4. Decryption

Compute

$$\bar{m} = INTT(\tilde{c}_1 * \tilde{r}_2 + \tilde{c}_2) \in R_q$$

using the inverse NTT. Then, the original message m is recovered from $\bar{m}$ using a decoder.

5. The Proposed Security Scheme

Our proposed scheme secures the formation and monitoring operations of the Surveillance UAV Networks in the disaster areas. Consequently, the efficiency of the rescue and evacuation operation in the areas is improved, i.e., evacuating more people in less time as their exact locations can be detected, protecting the rescuers' lives as they know exactly what to expect in addition to more ability to use robots in the most danger sites, and finally gathering precise data to study the disaster and predict/resist it in the future. The proposed scheme is divided into two phases.

5.1. Setup Phase

The setup phase is responsible for preparing the air drones for their missions and assigning the required security parameters for different parties. Figure 2 shows the setup phase.

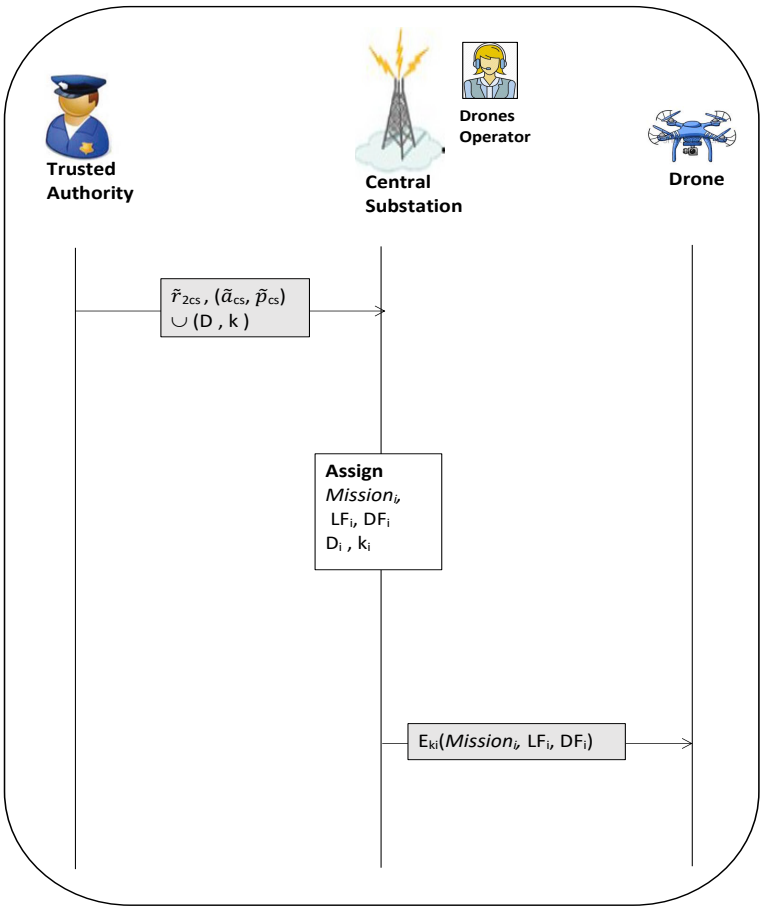


Figure 2. Setup phase.

- ***Issuing Key Parameters***

  TA provides the keying parameters for each party in the connection:

  TA issues a public/private key pair for CS; $\tilde{r}_{2cs}$ as a private key and $(\tilde{a}_{cs}, \tilde{p}_{cs})$ as public key, and then sends the key pair to CS via secure channel.

$$TA \xrightarrow{\tilde{r}_{2cs},(\tilde{a}_{cs},\tilde{p}_{cs})} CS.$$

TA assigns a set of unique secret IDs and secret session keys Ks and sends it securely to the CS. CS then provides a unique secret ID = D_w and a secret session key K = k_w for each drone, which uses D_w to prove its identity to CS. While the k_w is utilized during the round that the drone plays the cluster head role to organize the joining/leaving groups process:

$$TA \xrightarrow{\cup(D,k)} CS.$$

CS assigns these security parameters for the drones before releasing them to begin their missions ($CS\ assign < D_w, k_w > to\ D$).

- ***Forming the Groups***

  The whole disaster area is divided into subareas $\mathbb{A}$s; each subarea $\mathbb{A}_j$ is monitored by certain number of air drones m that are arranged in k groups; the number of the groups in the subarea is varied according to the size and importance of the subarea. For instance, the city downtown usually is more crucial than uptown, as downtown is crowded by people and has most of the business and important buildings.

  Drones can join or leave the group at any time as they can fly to the group or fly back to the CS to recharge their batteries. When the drones are flying over the subarea to collect information about the subarea, forward messages from the earth, or take photos/videos; they are called *Monitoring Drones*. While the drones are flying back to the substation, they are working as *Relaying Drones* to forward the messages from the cluster heads to the base station CS. In other words, the monitoring drone is converted to a relaying drone once it leaves its group to recharge the battery.

  For each group G_l, the CS assigns a different number of cluster heads $D_1, \ldots, D_\beta$ for each reading round, where β is varied for each group per reading round. For example, substation CS connects to four groups in the subarea A_1; each group has 20 drones. For group G_1, substation chooses drones D_2 and D_{15} to be two cluster heads for round 1; drones D_{20}, D_{18}, and D_{13} to be three cluster heads for round 2; and so on. The chain of cluster heads for each group is assigned by the corresponding CS, i.e., the CS operator.

  The groups in the same subarea are using different frequency channels to reduce the interference with other groups' connections.

- ***Programming the Drones***

  At CS, the drones' operator prepares each drone for its assigned task. The object *Mission* is assigned for each drone; *Mission* includes which subarea the drone has to scan, at what level it will fly, which group to enroll, and what mission to accomplish, such as taking photos/videos for the subarea, connecting with and forwarding the messages from the earth, or transferring first aid medical equipment. In addition, the operator determines when that specific drone works as a cluster head of its group. Consequently, the drone's path is pre-planned, i.e., the drones are tracked by GPS, so that, if a drone is redirected from the specified path, a suspicious action alarm is declared. For instance, the operator assigns $Mission_r$ for the drone D_r

  The operator sets two fingerprints' features [32] for each drone; these features extract characteristics from transmitted signals from the wireless devices and their environments to generate non-forgeable signatures. The first feature is a Location Feature LF to guarantee that the drones stick to their paths. For instance, the drone D_r mission is to join group G_y at the subarea $\mathbb{A}_s$ to take photos. Thus, the location feature LF_r for D_r is to be within the $\mathbb{A}_s$ subarea. If it moves outside $\mathbb{A}_s$, this means that it is controlled by a malicious adversary or a malfunction. Then, the operator does not depend on the D_r's information. The second feature, which is the Data Feature DF, is utilized to detect the compromised drones. The operator assigns a certain protocol

that the drone should follow before encrypting the collected data; if the drone does not apply that protocol, it is declared as a compromised node. The data feature could be a specific padding data that were added in certain bits, repeating the data in the packet in a pre-determined sequence, or adding the drone's ID in different location. For example, the operator assigns a certain padding data DF_r for D_r to add in specific bits in the message before encryption:

$$CS \xrightarrow{\mathbb{E}_{k_r}(Mission_r, LF_r, DF_r)} D_r.$$

In addition, the features are changed every time that the drone returns to the station for recharging its battery, i.e., the drone has a new mission in a different subarea and assigned other new features.

5.2. Surveillance UAV Network Operation Phase

This phase organizes the connection between the drones and CS to guarantee the efficiency of rescue operations in the disaster area.

5.2.1. Collecting the Area Information Procedure

- ***Collecting the Area Information***

  At the beginning of the reading round, each drone D_q in the group G_p scans its subarea $\mathbb{A}_{\rtimes}$, collects the required information, concatenates its secret ID D_q to the collected data v_j and inserting the data feature DF_q, and then encrypts the result by the CS public key $(\tilde{a}_{cs}, \tilde{p}_{cs})$:

  The message $v_w = (v_j|D_q|DF_q)$ is encoded to a polynomial $\bar{v}_w \in R_q$. Error polynomials $o_1, o_2, o_3 \in R_q$ are generated from $\mathcal{X}_\sigma$. Next, compute

$$\tilde{o}_1 \leftarrow NTT(o_1); \quad \tilde{o}_2 \leftarrow NTT(o_2).$$

  The encrypted message $(\tilde{h}, \tilde{z})$ equals

$$(\tilde{h}, \tilde{z}) \leftarrow (\tilde{a}_{cs} * \tilde{o}_1 + \tilde{o}_2, \tilde{p}_{cs} * \tilde{o}_1 + NTT(o_3 + \bar{v}_w).$$

  Then, D_q sends its message $(\tilde{h}, \tilde{z})$ to the current cluster heads of the group; assume that there are two cluster heads D_x and D_y for the current readings round:

$$D_q \xrightarrow{(\tilde{h}, \tilde{z})} D_x,$$

$$D_q \xrightarrow{(\tilde{h}, \tilde{z})} D_y.$$

  Each cluster head D_x and D_y concatenates all the received messages of the group:

$$B = concat_i \left(\tilde{h}_i, \tilde{z}_i\right),$$

  where i is the number of drones enrolled in the group during the current reading round.

  Then, D_x and D_y encrypt B by their session keys k_x and k_z:

$$\acute{B} = \mathbb{E}_{k_x}(B),$$

$$\bar{B} = \mathbb{E}_{k_y}(B)$$

  before forwarding the result messages $\acute{B}$, $\bar{B}$ to the CS via different relay drones R:

$$D_x \overset{\acute{B}}{\Rightarrow} R_j \overset{\acute{B}}{\Rightarrow} CS,$$

$$D_y \overset{\bar{B}}{\Rightarrow} R_k \overset{\bar{B}}{\Rightarrow} CS,$$

where the relays R_j and R_k are drones flying towards the CS.

- ***Verifying the Collected Information***

  When CS receives all the messages $\mathbb{B} = (\acute{B_1}, \bar{B_1}, \acute{B_2}, \bar{B_2}, \ldots \acute{B_n}, \bar{B_n})$, where n is the number of the groups in the subarea $\mathbb{A}_{\bowtie}$ (notice that each group sends two messages from two different cluster heads), it first decrypts and de-concatenates each message in $\mathbb{B}$. For example, the messages $\acute{B}, \bar{B}$ that were sent by the cluster heads D_x and D_y in the group G_p are decrypted and then the resulted B is concatenated as:

$$B = \mathbb{D}_{k_x}(\acute{B}),$$

$$B = \mathbb{D}_{k_y}(\bar{B}),$$

$$(\tilde{h}_i, \tilde{z}_i) = deconcat(B).$$

  For each message $(\tilde{h}, \tilde{z})$, CS decrypts:

$$\bar{v}_w = INTT(\tilde{h} * \tilde{r}_{2cs} + \tilde{z}) \in R_q$$

  using the inverse NTT. Then, the original message $v_w = (v_j | D_q)$ is recovered using a decoder.

  CS checks the drone's features: if the drone is flying within its assigned location LF_q; in addition, CS checks the presence of the inserted DF_q.

  Then, it verifies the validity of D_q before accepting the message v_j as an accurate reading.

  CS then compares between the received data from different drones (that assigned for the same task) and accepts the information from the majority; all drones of the groups $\mathbb{G}$ in the subarea $\mathbb{A}_{\bowtie}$ are expected to send similar readings to CS. Figure 3 shows the main procedure for collecting the subarea information. Figure 3 demonstrates the Collecting the Area Information procedure .

5.2.2. Join/Leave Procedures

- ***Join Process:***

  When a new drone D_n is sent from CS to join a specific group, the operator provides D_n by the session keys k_x and k_y of the current cluster heads D_x and D_y. Then, D_n encrypts two hello messages using k_x and k_y; notice that the hello messages include timestamps and random nonce to prevent the replay attacks:

$$c_{hello-x} = \mathbb{E}_{k_x}(m_{hello-x}),$$

$$c_{hello-y} = \mathbb{E}_{k_y}(m_{hello-y}).$$

  Then, the drone forwards them to D_x and D_y:

$$D_n \overset{c_{hello-x}}{\Longrightarrow} D_x,$$

$$D_n \overset{c_{hello-y}}{\Longrightarrow} D_y.$$

  The cluster heads reply to D_n by acknowledgment messages ACK encrypted by their session keys. Figure 4a shows the join procedures.

Leave Procedure:

The drone D_l, which needs to leave for battery recharging, embeds a recharge request in its previous reading message $v_l = (v_o | D_l | DF_l | Recharge)$, encrypts it to $(\tilde{h_l}, \tilde{z_l})$ and sends the result

to the current cluster heads of the group D_z and D_s; they include the message $(\tilde{h}_l, \tilde{z}_l)$ in the current total aggregated message F, encrypt F to $\check{F} = \mathbb{E}_{k_z}(F), \ddot{F} = \mathbb{E}_{k_s}(F)$ before forwarding $\check{F}, \ddot{F}$ to CS.

The CS then sends a forward request to the group heads D_z and D_s; this request asks them to forward the future aggregated messages of the group via the returning drone D_l, which becomes the relay R_l. The forward requests are encrypted by the cluster heads session keys k_z and k_s:

$$c_{forward-z} = \mathbb{E}_{k_z}(m_{forward-z}),$$

$$c_{forward-s} = \mathbb{E}_{k_s}(m_{forward-s}).$$

The forward request messages reach the cluster heads via the previously returning relays, e.g., R_j and R_k.

$$CS \xrightarrow{c_{forward-z}} R_j \xrightarrow{c_{forward-z}} D_z,$$

$$CS \xrightarrow{c_{forward-s}} R_k \xrightarrow{c_{forward-s}} D_s.$$

Then, D_z and D_s aggregate the group readings in one message P and send their encrypted group messages $\hat{P}, \dot{P}$ to the relay drone R_l:

$$D_z \xRightarrow{\hat{P}} R_l,$$

$$D_s \xRightarrow{\dot{P}} R_l.$$

R_l stores $\hat{P}, \dot{P}$ until it enters the CS's range and then forwards them to CS. Figure 4b shows the leave procedures:

$$R_l \xRightarrow{\hat{P},\dot{P}} CS.$$

In join/leave procedures, the symmetric session keys guarantee the confidentiality of the exchanged messages with tiny computation delay; in addition, the keys are only used for the current reading round, i.e., the probability of breaking the key is very low.

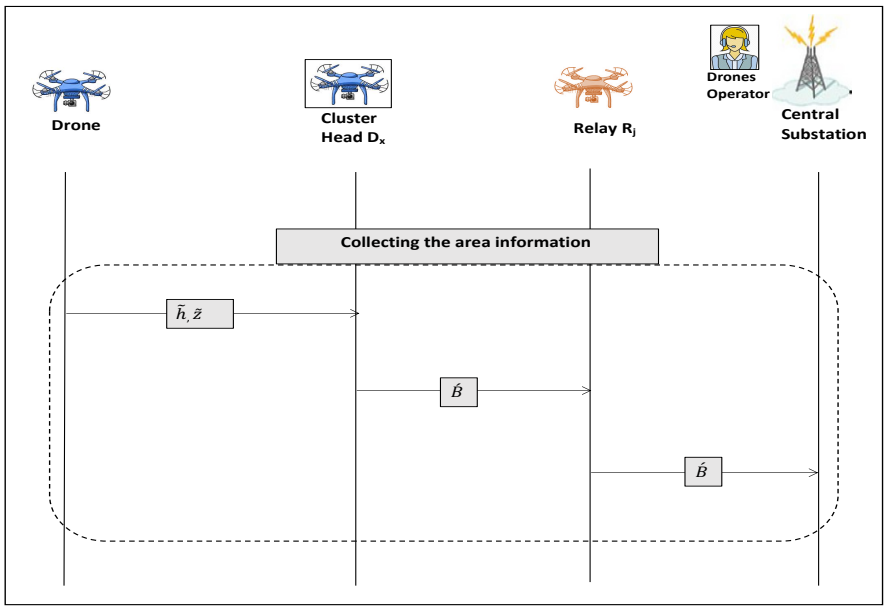
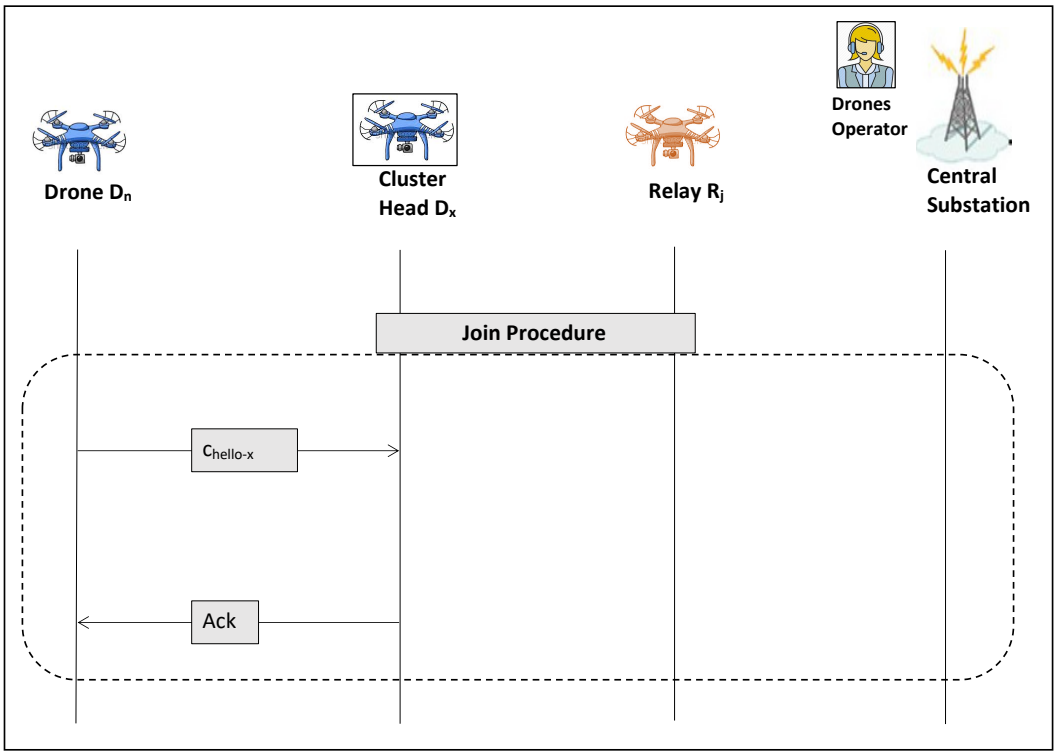
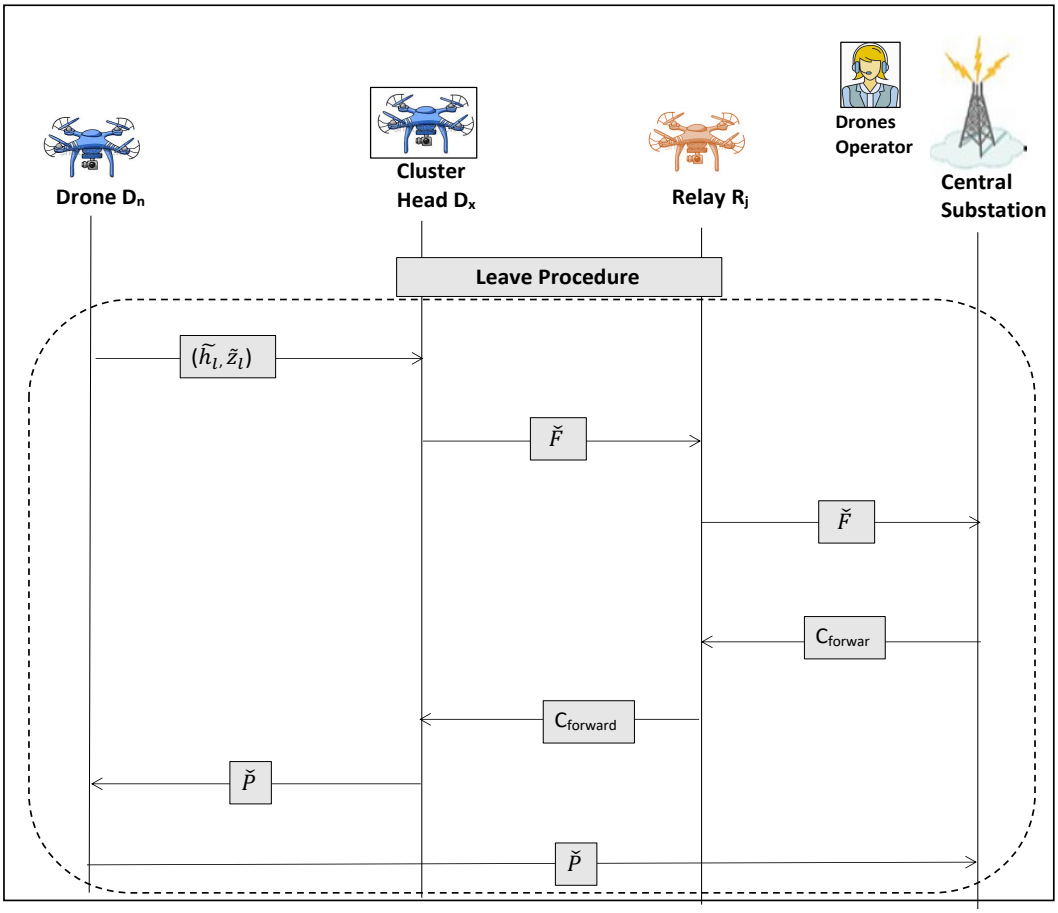


Figure 3. Surveillance UAV network operation phase collecting the area information.



(**a**) Join procedure.



(**b**) Leave procedure.

Figure 4. Surveillance UAV network operation phase—join/leave procedures.

6. Security Analysis

The main security concerns for the disaster surveillance UAV communication networks are the integrity and availability of drones' readings.

6.1. Information Integrity

The integrity of information collected for CS is a critical concern. In the disaster area, any false data could lead to aggravate the situation in the area or put the rescuers' lives in danger. The malicious adversaries can violate the integrity by compromising drones or intercepting the exchanged messages and falsifying them.

6.1.1. Compromised Drones

In our UAV communication model, there are two different types of drones. The first type is the monitoring drones D; these drones are enrolled in the monitoring groups to scan specific subareas and mainly collect useful information for the rescue operations. There are k number of groups that are responsible for each task in every subarea, where k is varied according to the size and importance of the subarea; CS receives several copies from the same piece of data from different groups. If an adversary $\mathscr{A}$ compromises a drone D_a from group G_u, there are several other drones in the same group that are sending the same data to the cluster heads. Thus, CS still can distinguish the false data from the received chain of messages.

If that compromised drone by the chance is one of the cluster heads for the reading round, the two messages received from the two heads are not identical. CS in that case rejects the two sets and reports that group as a malicious one. However, CS still obtains the required information from the other honest groups in the subarea. The CS follows the same procedure if $\mathscr{A}$ compromises both cluster heads or even compromises the whole group. If $\mathscr{A}$ compromises several groups, i.e., CS receives different sets of values for the same reading, then CS accepts the majority. We assumed that $\mathscr{A}$ can only compromise a limited number of groups and the probability that the attacker compromises a large number of groups is low.

The second group of drones is the relays R, which are drones flying back to the station, i.e., to recharge their batteries. Relays are forwarding the messages from the cluster heads to CS. If $\mathscr{A}$ compromises a relay drone R_x, he/she should know all IDs for the monitoring drones and cluster heads for all transmitted messages from different groups to be able to fabricate the forwarded messages, i.e., $\mathscr{A}$ has to decrypt all the messages and extract the concatenated IDs, which is an NP-hard problem. If $\mathscr{A}$ attempts to destroy the relay R_x and prevent it from reaching CS, or block the forwarded messages from it, there are plenty of other relays to perform the same job. For example, on average, the drone scans the area for 15 min before flying back to recharge.

In addition, all drones have to follow their assigned location and data features; if $\mathscr{A}$ attempts to redirect the drone to another location, its LF is different than that assigned by the operator. Thus, a malicious action is detected, while, if $\mathscr{A}$ compromises the drone and inserts his/her fake information, he/she does not follow the determined routine by the operator, i.e., does not insert a data feature DF, and just encrypts the fake data by CS's public key. Consequently, the drone is declared as compromised device and the operator does not relay on its information.

6.1.2. Intercepted Messages

Adversaries cannot falsify the readings during their transmission to CS, as the messages are encrypted by the powerful ring-LWE crypto-system. If $\mathscr{A}$ manages to intercept the message $(\tilde{h}, \tilde{z})$ that was sent from drone D_q to the cluster heads of the group D_x and D_y, he/she cannot modify its value because $\mathscr{A}$ does not have the decryption key $\tilde{r}_{2cs}$ and consequently cannot extract the plaintext measurement m_j from $(\tilde{h}, \tilde{z})$. Moreover, the messages contain timestamps and random nonces to

prevent the replay attack. Thus, $\mathscr{A}$ cannot interpret and modify the message's content or begin a replay attack.

According to join/leave procedure messages, they are encrypted by the cluster heads' session keys; only the head and the connecting drone share the key to manage the drone join/leave process. The cluster heads and consequently their session keys are changing per minute, i.e., every round, so that the probability of compromising these keys is diminished. Even if certain cluster heads, and their keys, are compromised, the *CS* still obtains the accurate monitoring information for the area via alternative ways.

6.2. Information Availability

To guarantee the efficiency and reliability of the surveillance UAV Network operation, the information aggregated from the disaster area should be available to *CS* whenever *CS* asked for it. Our proposed scheme allows redundancy in the readings, as several drones are monitoring the same subarea and sending their versions of the scanning data to *CS* via cluster heads and relays. If some drones in the same group or same subarea are not available, the remaining units still send their data to *CS*. If $\mathscr{A}$ compromises a certain number of drones, *CS* still can guarantee the correctness of each reading value by receiving redundant values for the same information from other drones.

Moreover, *CS* reduces the probability of attacks by eliminating the suspicious drones. If *CS* does not receive the expected messages from certain groups, or if units that forward the group's messages to *CS* are not the current chosen cluster heads, *CS* blocks these malicious heads and checks the whole group, i.e., *CS* realizes that these nodes are compromised by $\mathscr{A}$.

In case of a regular drone malfunction, other drones in the group are still sending their information. If a cluster head fails or malfunctions, *CS* receives the data from the other head. Even if the whole group failed, other groups are supplying *CS* by the required information. Thus, the data availability is guaranteed and drone malfunction does not have an effect on the rescue operation's efficiency.

6.3. Confidentiality

Although messages' confidentiality is not a primacy security concern for UAV networks, it is still a crucial requirement. If the messages are exchanged in plaintext, then the malicious adversaries can intercept/eavesdrop and falsify its content to do damage or make the situation worst in the disaster area. Thus, our proposed scheme guarantees the confidentiality of the exchanged messages between different parties in the network. Outside adversaries cannot extract the contents of the transmitted messages because it is encrypted by the powerful LWE encryption scheme. If $\mathscr{A}$ manages to intercept a message $(\tilde{h}, \tilde{z})$, he/she cannot obtain the plaintext reading ms_i because $\mathscr{A}$ does not have access to the private key of *CS* $\tilde{r}_{2cs}$ and consequently cannot decrypt m_j. Even with compromising certain drones, capturing/falsifying their messages, or preventing them from reaching to *CS*, *CS* still receives the readings from other honest drones in the group; in addition, certain other groups scan the subarea and send similar data so that compromising drones or blocking messages does not have a significant impact on the final results.

Based on the hardness of Ring-LWE problem, it is an NP-hard problem to extract the plaintext messages from the encrypted versions. If $R = \mathbb{Z}[X]/(X^n + 1)$ for n a power of two, and $R_q = R/qR$, where elements of R_q are polynomials of degree $< n$ with mod-q coefficients. It is an NP-hard problem to find secret ring element $s(X) \in R_q$, given:

$$
\begin{aligned}
a_1 &\rightarrow R_q, \quad b_1 = a_1.s + e_1 \in R_q, \\
a_2 &\rightarrow R_q, \quad b_2 = a_2.s + e_2 \in R_q, \\
a_3 &\rightarrow R_q, \quad b_3 = a_3.s + e_3 \in R_q, \\
&\vdots
\end{aligned}
$$

where $(a_i, b_i) \in R_q \times R_q$ are uniformly random subject to $b_i - a_i.s \approx 0$ and the error $e_i \in R$ are small values. $\mathscr{A}$ cannot compromise the CS's secret key $\tilde{r}_{2cs}$ even via a quantum computer [30].

In summary, the integrity and availability of the received information by CS are guaranteed because each subarea is monitored by several groups of drones and encrypted versions of their readings are sent to CS via different relays. Then, data redundancy and network's reliability assures the UAV communication security.

Table 1 summarizes the security aspects of our proposed scheme.

Table 1. The proposed scheme security.

Information Integrity	• Guarantee readings messages integrity while some compromised monitoring drones/cluster heads exist. • Guarantee forwarding packages integrity even if certain compromised relay drones exist • Guarantee the accuracy of drones' locations/flying paths. • Guarantee the integrity of transmitted readings messages during monitoring operation. • Guarantee the integrity of exchanged messages during join/leave Procedure.
Information Availability	• Guarantee information availability even if some monitoring drones are compromised/malfunctioned. • Guarantee readings availability even if one or both cluster heads of a group are compromised/malfunctioned. • Guarantee information availability in case of a whole group failure/malfunctioned.
Confidentiality	• Guarantee that only CS can access the plaintext readings messages. • Guarantee that CS still receives an accurate version of the readings even if a certain number of drones are compromised. • Guarantee that adversaries cannot extract the plaintext information nor falsify it.

7. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme in terms of communication overhead and computation complexity.

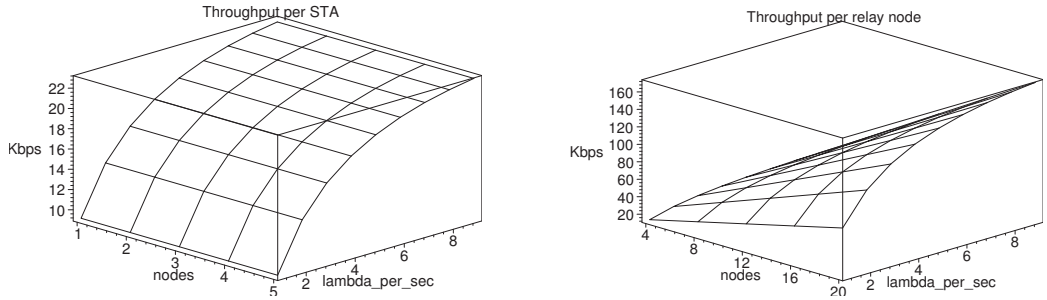
7.1. Network Performance

We have a modeled network at MAC level using Maple 13 from Maplesoft, Inc. of Waterloo, ON, Canada. The UAV network per cluster has a bandwidth of 4 MHz. We assume that all non-relay nodes belong to the traffic class 0 and that each non-relay has the same Poisson packet arrival rate $\lambda_k = 1.9$ per second. Cluster membership was varied between 4 and 20 drones. The parameters for the model are shown in Table 2.

Table 2. Parameters for model of the UAV network at MAC level.

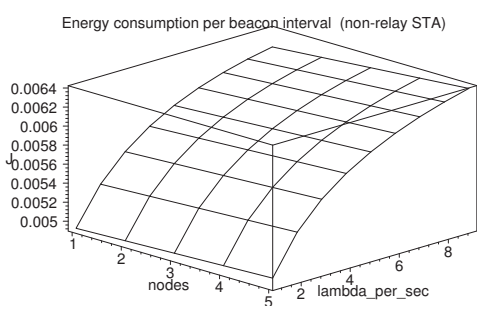
Parameters	Numerical Values
Duration of Time slot, ω	52 μs
Bit error rate (BER)	$2 * 10^{-6}$ bits/s
Minimum physical (PHY) layer header	6ω μs
Arbitration inter frame space (AIFSN)	2
Data rate	650 Kbps
MAC service data unit (MSDU) length	256 octets
Short Inter-frame space (SIFS) duration	160 μs
MAC header length	14 bytes
PS-Poll	6ω μs
PS-Poll-ACK	6ω μs
Block acknowledgment (BA)	6ω μs
Maximum retry limit , R	7
Max. number of antennas in AP, A_{ap}	1
Number of antennas in STA	1
Bandwidth	4 MHz
OFDM symbol duration	40 μs
Number of bits in an OFDM symbol	54
Modulation and Coding scheme, MCS	0
Beacon interval, BI	1 s
RAW slot	200 ms
Minimum contention window size W_{min}	31
Transaction opportunity (TXOP) limit	2 packets
Idle mode energy consumption ω_i	$18.2 * 10^{(-9)}$ J
Receive mode energy consumption ω_r	$17.9 * 10^{(-6)}$ J
Transmit mode energy consumption ω_t	$15.8 * 10^{(-6)}$ J

Figure 5a,b show the throughput of a non-relay and relay drones, respectively. Throughput of non-relay nodes is not sensitive to the number of nodes due to the feature that each node can have only a single transmission during RAW slot and is forced to a doze mode after the transmission. However, node throughput shows asymptotic behavior when packet arrival rate per node increases since each node can get at most $\frac{1}{N}$ of available bandwidth. Consequently, the throughput of relay nodes shows linearity with respect to the number of nodes in the cluster and asymptotic behavior with respect to the packet arrival rate. Energy consumption of non-relay drones and relay drones respectively during one beacon interval (which is set to 1 s) is shown in Figure 6a,b. We observe that the energy consumption of a non-relay and relay drones increase strongly with packet arrival rate while there is a mild increase with a number of nodes in the cluster. If we look into total airborne time of 30 min where node is non-relaying for 20 min and relaying for 10 min, this results in maximum energy consumption of ≈150 J. Those costs largely overshadow energy consumption for encryption.

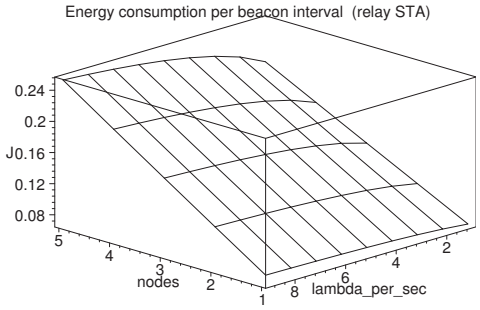


(**a**) Throughput of a non relay node. (**b**) Throughput of a relay node.

Figure 5. Per node throughput.



(**a**) Energy consumption per beacon interval of a non-relay node.



(**b**) Energy consumption per beacon interval of a relay node.

Figure 6. Energy consumption.

7.2. Communication Complexity

To guarantee the accuracy of rescue operations, information about current status of the disaster area should be sent periodically to *CS*. Thus, the communication duty for each monitoring drone every reading round is sending two messages to the two cluster heads while each head sends the concatenated message to *CS* via relays; this communication burden is affordable for the drones. According to the relays, they store the received messages and then forward them all as one packet to the *CS* when the relay enters the coverage range of *CS*. In other works, relays' communication load is considered one message too. Because the proposed scheme guarantees the integrity and availability of the information by several ways, the probability for *CS* to ask for specific information retransmission due to malfunction or malicious attacks is reduced.

The communication overhead, the number of transmitted messages, for each monitoring drone equals two messages and for the whole mission time, i.e., 15 min, equals 1800 messages (if the drone sends its reading every second). While the relays receive and store certain number of messages during the journey but forward them as a one message when reaching to *CS*, the relay is sending one message. However, in the join/leave process, the connecting drone only sends one message to the head to join the group. In conclusion, the total communication load for drones is limited and trivial overhead. Table 3 demonstrates the communication overhead for each monitoring drone per round and for the whole flying time (15 min) in the proposed scheme.

Table 3. Monitoring drones communication overhead.

Drone Type	Communication Overhead	
	Per Round	Total
Monitoring	2	1800
Relay	0	1

Figure 7 shows the communication overhead for the operation of collecting the area information per reading round; it presents the communication load for readings transmission from the monitoring drones to the cluster heads in each group in the whole subarea. The number of transmitted messages by each monitoring drone in the reading round is fixed so that the communication delay of the group increases linearly as the number of the involved drones increases. Similarly, the communication overhead for the subarea increases as the number of the groups and number of participated drones increase. This leads to the fact that the more important subareas have higher communication burden, but the communication overhead is still affordable by the UAV network.

According to cluster heads, their communication loads include the overhead of forwarding the group aggregated readings to the chosen relays, i.e., in addition to the communication overhead of their monitoring mission. The number of relays is changing according to the drones' batteries conditions;

this number seems unpredictably and randomly changing for the cluster heads. Consequently, the total number of forwarded messages by clusters is fluctuated according to the current number of relays. Figure 8 shows the real-time variation in the cluster head's communication overhead due to forwarding the readings packages to relays. The figure presents three different cases: when the maximum number of drones in the group can reach 5, 10, and 20 drones. Notice that the number of forwarding packages fluctuates as the number of relays changes but within a range, i.e., less than the maximum number of drones in the group. For instance, when the group has 20 drones, the number of drones converted to relays is always changing in the range zero to less than 20. In addition, the cluster head selects number of relays (not all of the returning drones) to forward the reading package. Moreover, the drone plays the cluster head role for a limited time period; it may be not working as a cluster head at all for the whole flying time. Thus, the overhead to communicate relays is tolerable by the cluster head.

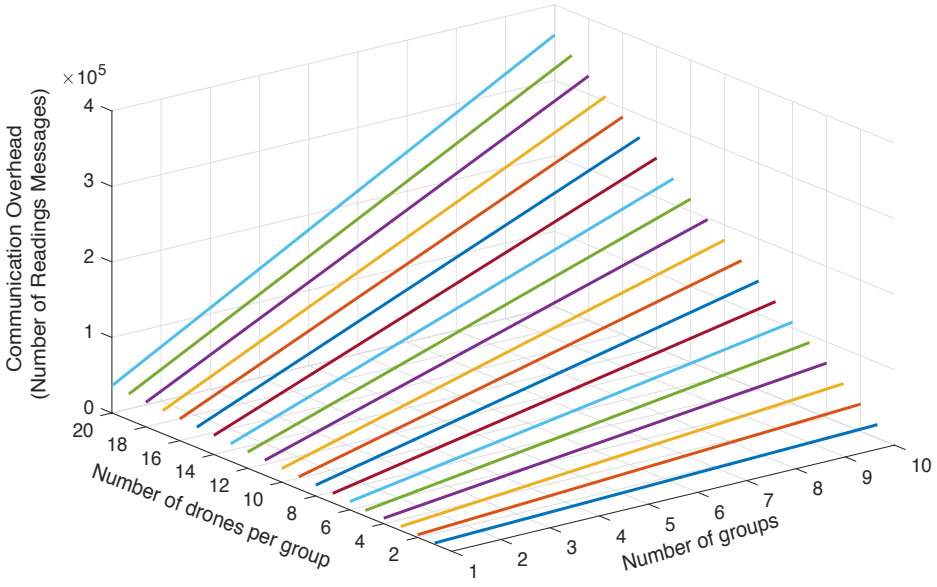


Figure 7. Communication overhead for monitoring operation.

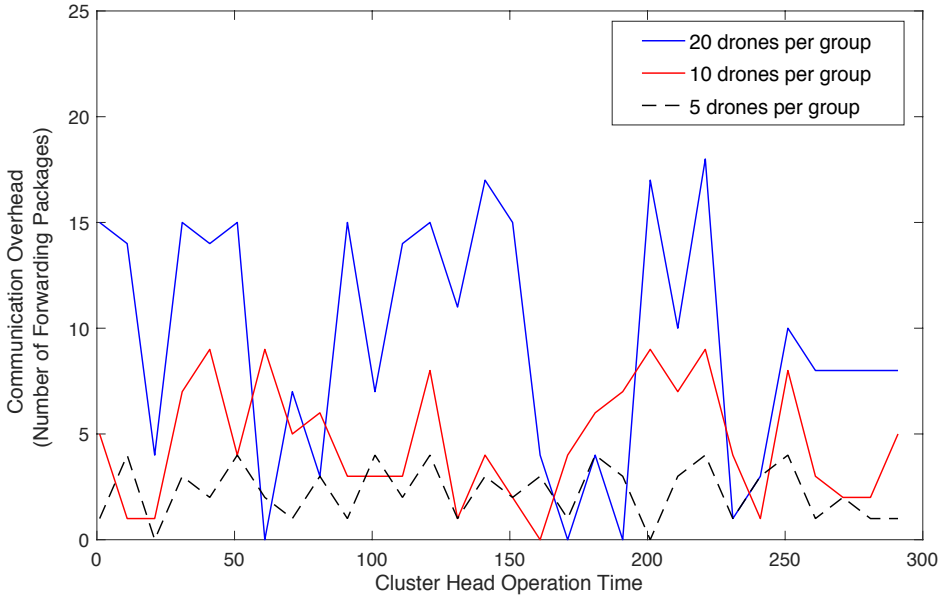


Figure 8. Cluster head communication overhead for the forwarding process.

7.3. Computation Complexity

The monitoring drones' main task is scanning the area and forwarding photos or videos about the situation in the disaster area to *CS*. The drones are not capable of performing complex cryptographic operations to preserve their energy. Our proposed scheme implements the efficient Ring-LWE crypto-system [33] on the drones to guarantee the security requirements without increasing the computation overhead on drones; the monitoring drones need to perform one encryption processes only per round. During the mission time, the monitoring drones encrypt and then send their periodic reports to the two cluster heads every second, which equals 1800 messages in total but only 900 encryption operations. $C_D = 900 * T_e$, where T_e is the encryption time; the computation overhead per drone is trivial and does not consider a load on the limited-computation capabilities units. For the whole subarea, the total computation delay equals $C_T = 900 * T_e * I * J$, where I is the number of groups in the subarea and J is the current number of participated drones in each group, i.e., J varies from one group to another and varies at the same group from time to another, while the computation load for the join/leave process is neglected as the process utilizes symmetric key encryption with tiny computation overhead. Figure 9 shows the total computation overhead for the subarea during the collecting the area information phase as the number of drones and groups vary. As shown, the total load for the subarea is linearly increased as the number of drones and groups increase, as the number of encrypted messages by the drone during the reading round is constant. However, the overhead is restricted by the maximum number of groups and drones in the area, which is a limited number. Then, the computation burden for the subarea is small and tolerable by the network.

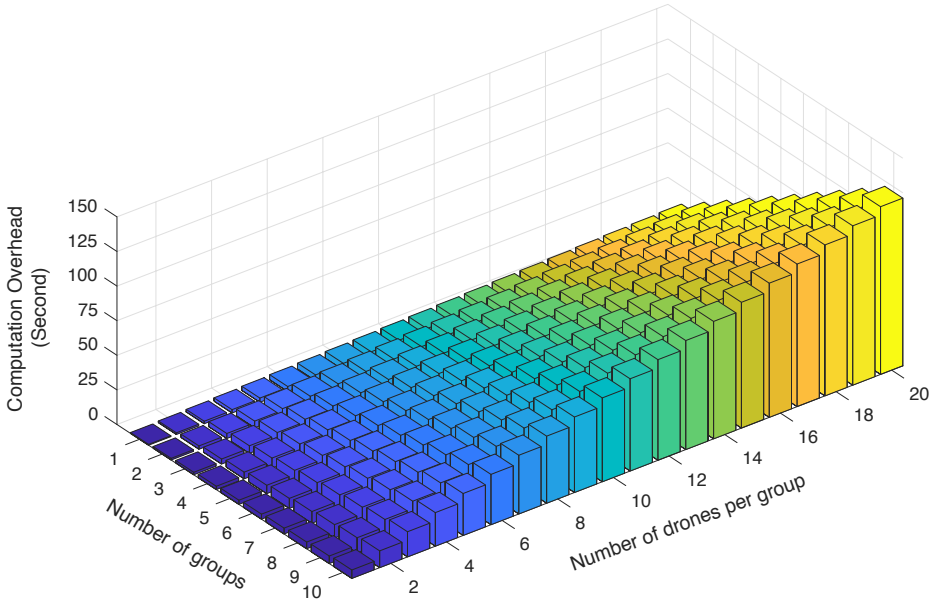


Figure 9. Computation overhead for monitoring operation.

In addition, we have studied the performance of our proposed scheme utilizing Ring-LWE versus using RSA 2048 crypto-system (which is widely used for securing data transmission in practice) implementing the ARM Cortex M4F platform (Cambridge, UK). Figure 10 shows the computation overhead comparison per group as the included drones' number changes. It can be seen that the overhead in the RSA-based scheme is rapidly increased while the number of computation delays for Ring-LWE based proposed scheme remains very few. The huge gap between the overhead in the two cases is clear, because of the simple ARM Cortex M4F processing abilities; the platform is suitable for the limited-computation capabilities drones in the system. As the lightweight Ring-LWE encryption/decryption operations do not require complex processing units, the proposed scheme performance is more efficient in the Ring-LWE-based case. According to the total computation overhead

of the subarea, our proposed scheme utilizing Ring-LWE crypto-system overhead increases as the number of groups increases from 900 msec in one group that includes a one-drone case to 18,000 msec in the case of 10 groups with 20 drones each. While the computation overhead for the RSA-based scenario is ranging between 79,200 msec until 1,584,000 msec as the number of groups and included drones increases. In summary, implementing our proposed scheme using a Ring-LWE crypto-system will not only protect it against the post-quantum computer attacks but also achieve the task with low computation overhead even on the simple Cortex M4F platform.

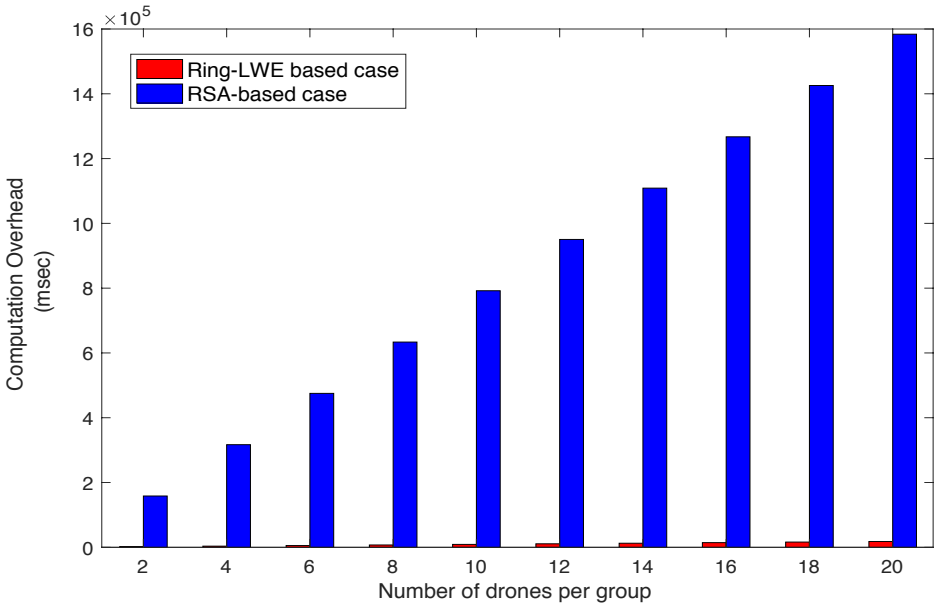


Figure 10. Computation overhead for a group Ring-LWE-based proposed scheme versus an RSA-based case.

7.4. Energy Consumption Estimation

To accomplish its mission in the affected area, the drone consumes energy to fly, scans the area, collects the targeted data, ciphers the message and forwards it to *CS* via other drones. All these operations are powered by the drone's battery, which has restricted capacity. Thus, one of the main aims of our proposed scheme is to be lightweight and efficient in terms of energy consumption. Although encrypting and transmitting the readings consumes power, other drone parameters, such as the drone's weight and hovering speed, can deplete the drone battery. Thus, planning the drone's mission, including defining the battery threshold level, should take these parameters into consideration. The energy threshold is the power level that the drone has to finish the mission and return to the *CS* by reaching it; this threshold should be higher than the battery recharging alarm level. In our proposed scheme, the drone is flying for 30 min; it hovers over the disaster area for around 15 min while the remaining time is evenly split between flying to the disaster area and back to the recharging station. This flying time is expected to be safe in terms of power consumption.

In this section, we investigate the energy consumption trend if the communication and computation burdens vary, assuming drones with equal weight and velocity. We use the data for Mavic series drones by DJI (Shenzhen, China, https://www.dji.com/products/mavic) which feature flying time of 30 min or more under a light load. Figure 11 shows the battery consumption pattern with Drone x working as a cluster head for the group for a period of time besides its main mission (monitoring the area and collecting information) while drone y only scans the area and aggregates data. As shown, the energy consumption curve is divided into three phases: the first one is the period of flying from the *CS* to the area. The two drones fly at the same speed so that the curves are identical (We assume that the drones fly at a high velocity from the station to the area; this is the reason for the

high energy consumption). When the drones reach their assigned locations, the second phase begins; the drone reduces its hovering speed to approximately 50–60%, i.e., notice that the consumption is lower than the first stage. Then, the drones start their missions. Clearly, Drone x produces higher loads and consumes more energy than Drone y because it performs more tasks than Drone y. However, the difference between the two power consumption curves is small due to the light communication and computation complexity of our proposed scheme. Finally, the drones fly back to the station, i.e., at a high velocity as the first phase. The drones have finished their missions and landed before reaching the threshold level.

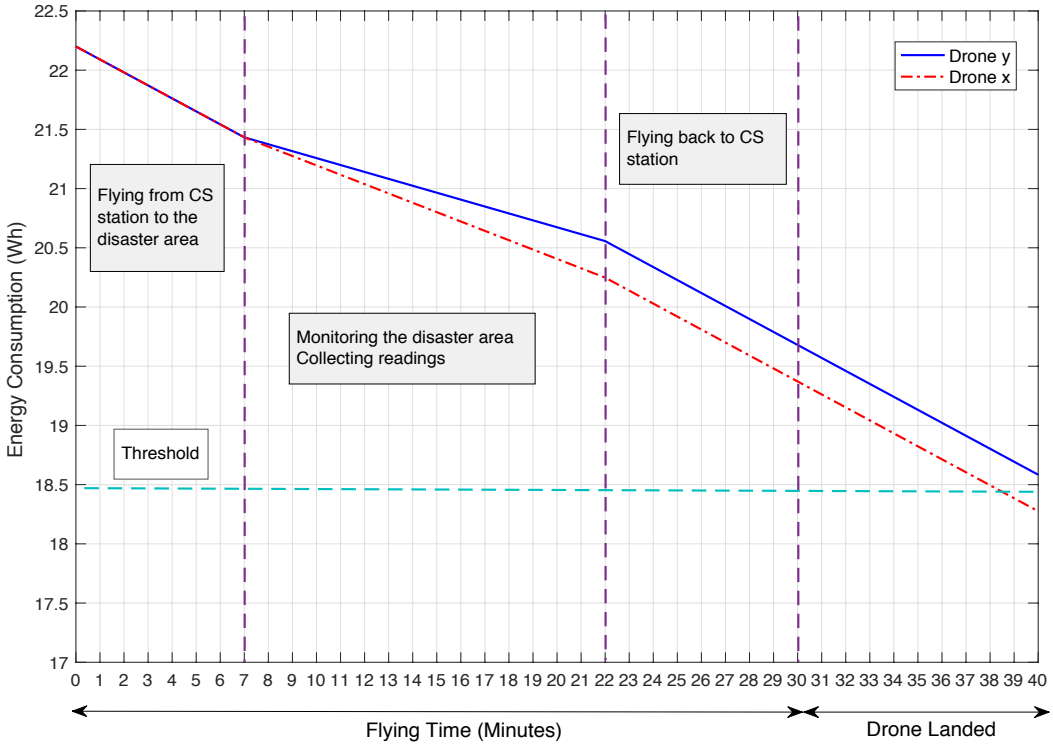


Figure 11. Energy consumption pattern for drone trip.

8. Conclusions

In this paper, we have proposed a network and security architecture for the disaster surveillance UAV system. The network architecture of the system is based on IEEE 802.11ah standard where non-relaying and relaying nodes are separated using different RAW slots which provide low access contention for sensed and relayed data. This system provides a high degree of accuracy and availability of the collected information from the disaster area and consequently assures the quality of the rescue and evacuation operations. The proposed scheme preserves the integrity and availability of the collected information by utilizing fingerprint features and data redundancy techniques; it also deploys the lightweight Ring-LWE crypto-system to further protect the confidentiality of the transmitted messages with a low computation burden.

Author Contributions: A.A., J.M., and V.B.M. designed the security scheme; A.A., M.Z.A., J.M., and V.B.M. designed the communication scheme and the performance evaluation experiments. J.M. and V.B.M. wrote the final version of the paper.

Funding: The work of J.M. and V.B.M. was supported through their respective NSERC Discovery Grants. The work of M.Z.A. was partially supported by the Faculty of Science, Ryerson University.

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

References

1. Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152. [CrossRef]
2. Frew, E.; Brown, T. Airborne Communication Networks for Small Unmanned Aircraft Systems. *Proc. IEEE* **2008**, *96*, 2008–2027. [CrossRef]
3. Silva, M.R.; Souza, E.S.; Alsina, P.J.; Francisco, H.C.; Medeiros, A.A.D.; Nogueira, M.B.; de Alburquerque, G.G.L.A.; Dantas, J.B.D. Communication Network Architecture Specification for Multi-UAV System Applied to Scanning Rocket Impact Area First Results. In Proceedings of the 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, Brazil, 8–11 November 2017.
4. Chen, T.; Xiao, Y.; Zhao, X.; Gao, T.; Xu, Z. 4G UAV Communication System and Hovering Height Optimization for Public Safety. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017.
5. Arnold, S.; Montgomery, G.; Gopal, R.; Losada, D. Validation of high-speed broadband satellite communications on airborne platforms. In Proceedings of the Military Communications Conference (MILCOM), San Jose, CA, USA, 31 October–3 November 2010.
6. Nintanavongsa, P.; Pitimon, I. Impact of Sensor Mobility on UAV-based Smart Farm Communications. In Proceedings of the 2017 International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 8–10 March 2017.
7. Chakareski, J. Aerial UAV-IoT Sensing for Ubiquitous Immersive Communication and Virtual Human Teleportation. In Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Atlanta, GA, USA, 1–4 May 2017.
8. Ali, M.Z.; Misic, J.; Misic, V.B. Efficiency of Restricted Access Window Scheme of IEEE 802.11ah under Non-ideal Channel Condition. In Proceedings of the 2018 IEEE International Conference on Green Computing and Communications (GreenCom-2018), Halifax, NS, Canada, 30 July–3 August 2018.
9. Ali, M.Z.; Misic, J.; Misic, V.B. Differentiated QoS to Heterogeneous IoT Nodes in IEEE 802.11ah RAW Mechanism. In Proceedings of the IEEE Globecom 2018, Abu Dhabi, UAE, 9–13 December 2018.
10. Zhu, Y.; Huang, Q.; Li, J.; Wu, D. Design and evaluation of airborne communication networks. In Proceedings of the 2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo, Japan, 7–10 July 2015.
11. Guo, W.; Devine, C.; Wang, S. Performance Analysis of Micro Unmanned Airborne Communication Relays for Cellular Networks. In Proceedings of the International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), Manchester, UK, 23–25 July 2014.
12. Kumbhar, A.; Güvenc, I.; Singh, S.; Tuncer, A. Exploiting LTE-Advanced HetNets and FeICIC for UAV-assisted Public Safety Communications. *IEEE Access* **2018**, *6*, 783–796. [CrossRef]
13. Scherer, J.; Rinner, B. Short and Full Horizon Motion Planning for Persistent multi-UAV Surveillance with Energy and Communication Constraints. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017), Vancouver, BC, Canada, 24–28 September 2017.
14. He, H.; Zhang, S.; Zeng, Y.; Zhang, R. Joint Altitude and Beam-width Optimization for UAV-Enabled Multiuser Communications. *IEEE Commun. Lett.* **2018**, *22*, 344–347. [CrossRef]
15. Ladosz, P.; Oh, H.; Chen, W. Prediction of Air-to-Ground Communication Strength for Relay UAV Trajectory Planner in Urban Environments. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.
16. Huo, Y.; Dong, X.; Lu, T.; Xu, W.; Yuen, M. Distributed and Multi-layer UAV Network for the Next-generation Wireless Communication. *arXiv Preprint* **2018**, arxiv:1805.01534.
17. Lin, X.; Yajnanarayana, V.; Muruganathan, S.; Gao, S.; Asplund, H.; Maattanen, H.; Bergström, M.; Euler, S.; Wang, Y. The sky is not the limit: LTE for unmanned aerial vehicles. *IEEE Commun. Mag.* **2018**, *56*, 204–210. [CrossRef]
18. Xu, X.; Zeng, Y.; Guan, Y.; Zhang, R. Overcoming endurance issue: UAV-enabled communications with proactive caching. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1231–1244. [CrossRef]
19. Sharma, V.; Jayakody, D.; Srinivasan, K. On the Positioning Likelihood of UAVs in 5G Networks. *Phys. Commun.* **2018**, *31*, 1–9. [CrossRef]

20. Yap, K.M.; Eu, K.S.; Low, J.M. Investigating Wireless Network Interferences of Autonomous Drones with Camera Based Positioning Control System. In Proceedings of the 2016 International Computer Symposium (ICS), Chiayi, Taiwan, 15–17 December 2016; pp. 369–373.
21. Kagawa, T.; Ono, F.; Shan, L.; Takizawa, K.; Miura, E.; Li, H.; Kojima, F.; Kato, S. A study on latency-guaranteed multi-hop wireless communication system for control of robots and drones. In Proceedings of the 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), Bali, Indonesia, 17–20 December 2017; pp. 417–421.
22. Biswas, S.; Misic, J.; Misic, V.B. DDoS Attack on WAVE-enabled VANET Through Synchronization. In Proceedings of the IEEE Globecom 2012, Anaheim, CA, USA, 3–7 December 2012.
23. Altawy, R.; Youssef, A. Security, privacy, and safety aspects of civilian drones: A survey. *ACM Trans. Cyber-Phys. Syst.* **2017**, *1*, 7. [CrossRef]
24. Liu, C.; Quek, T.; Lee, J. Secure UAV Communication in the Presence of Active Eavesdropper. In Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017.
25. Rubin, S.; Grefe, W.; Bouabana-Tebibel, T.; Chen, S.; Shyu, M.; Simonsen, K. Cyber-Secure UAV Communications using Heuristically Inferred Stochastic Grammars and Hard Real-Time Adaptive Waveform Synthesis and Evolution. In Proceedings of the 2017 IEEE International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 4–6 August 2017.
26. Sharma, V.; Kumar, R.; Srinivasan, K.; Jayakody, D. Coagulation Attacks over Networked UAVs: Concept, Challenges, and Research Aspects. *Int. J. Eng. Technol.* **2018**, *7*, 183–187.
27. Sharma, V.; Jayakody, D.; You, I.; Kumar, R.; Li, J. Secure and Efficient Context-Aware Localization of Drones in Urban Scenarios. *IEEE Commun. Mag.* **2018**, *56*, 120–128. [CrossRef]
28. Regev, O. The Learning with Errors Problem (Invited Survey). In Proceedings of the IEEE Conference on Computational Complexity, Cambridge, MA, USA, 9–11 June 2010.
29. Ducas, L.; Durmus, A. Ring-LWE in polynomial rings. In Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, 21–23 May 2012; pp. 34–51.
30. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. *J. ACM* **2013**, *60*, 43. [CrossRef]
31. Roy, S.; Vercauteren, F.; Mentens, N.; Chen, D.; Verbauwhede, I. Compact Ring-LWE Cryptoprocessor. In Proceedings of the 16th International Workshop on Cryptographic Hardware and Embedded Systems—CHES 2014, Busan, Korea, 23–26 September 2014; pp. 371–391.
32. Xu, Q.; Zheng, R.; Saad, W.; Han, Z. Device Fingerprinting in Wireless Networks: Challenges and Opportunities. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 94–104. [CrossRef]
33. De Clercq, R.; Roy, S.; Vercauteren, F.; Verbauwhede, I. Efficient software implementation of ring-LWE encryption. In Proceedings of the 2015 Design, Automation, and Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2015.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).