Hindawi
Security and Communication Networks
Volume 2018, Article ID 6475927, 17 pages
https://doi.org/10.1155/2018/6475927

Research Article

A UAV-Aided Cluster Head Election Framework and Applying Such to Security-Driven Cluster Head Election Schemes: A Survey

Gicheol Wang[1], **Byoung-Sun Lee**[1], **Jae Young Ahn**[1] **and Gihwan Cho**[2]

[1]*Autonomous Unmanned Vehicle Research Division, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea*
[2]*Division of Computer Science and Engineering, Chonbuk National University, Jeonju 54896, Republic of Korea*

Correspondence should be addressed to Gicheol Wang; gcwang@etri.re.kr

Received 16 March 2018; Revised 10 May 2018; Accepted 17 May 2018; Published 19 June 2018

Academic Editor: Angelos Antonopoulos

UAS (Unmanned Aerial Systems) are now drawing a lot of attention from academic and research fields as well as the general public. The UAS is expected to provide many promising applications such as intelligent transportation system, disaster management, search and rescue, public safety, smart delivery, wild species monitoring, and wireless service area extension. More specifically, as a part of the wireless service extension, we deal with the information dissemination and collection using a UAV in this paper. In this application, because the UAV communicates with each CH (Cluster Head) to collect data from sensor nodes or to disseminate information to the sensor nodes, well-behaved and qualified nodes should be elected as CHs and their integrity should be preserved. Even though a UAV makes the information dissemination and collection process efficient in a WSN, we can make the UAV help the election of new CHs to mitigate the threat of compromised CHs. To this aim, we first propose a UAV-aided CH election framework where a UAV delivers the critical information collected from sensors to the sink, and the sink reselects a set of well-behaved and qualified CHs considering the information. Then, we classify the existing security-driven CH election schemes into several categories and explain the principle of each category and its representative schemes. For each representative scheme, we also explain how to adapt it into the UAV-aided CH election framework. Next, we identify some desirable security properties that a CH election scheme should provide and reveal the security level that each representative scheme reaches for the desirable security properties. Next, we compare communication and computation overhead of the security-driven CH election schemes in terms of the big O notation. In conclusion, we reveal what we have learned from this survey work and provide a future work item.

1. Introduction

Recently, UAVs or drones are drawing a lot of attention from academic and industrial fields as well as the general public as they have been widely used in our daily lives. Generally, a UAV carries some mission devices to fulfil its duties during a flight, and it is controlled by a pilot with a controller or in a GCS (Ground Control Station). To control the UAV, there is a communication system between the GCS (controller) and the UAV to exchange data between them [1, 2]. The UAV, the GCS, and the communication system comprise a UAS (Unmanned Aerial System).

In the past, UAS (Unmanned Aerial Systems) were mainly employed for military missions such as surveillance and reconnaissance. With the rapid development of UAS-related technologies, UAS are expected to be utilized for many civilian applications as well as military missions [3]. The UAS civilian applications include intelligent transportation system [4], disaster management [5], search and rescue [6], public safety [7], smart delivery [8], wild species monitoring [9], and service extension of wireless communications [10–18]. Table 1 presents some promising UAS applications and describes the scenario of each application.

With significant advancements in wireless networking and UAS-related technologies, UAVs have been employed for extension of wireless communication service such as UAV-aided ubiquitous coverage [10–14], UAV-aided relaying [10], and UAV-aided information dissemination and collection

Table 1: Promising applications of UAS.

Category	Application	Scenario
Intelligent transportation system	Traffic accident report [4]	UAVs aware a traffic accident at a high altitude in advance and quickly alerts it to prevent a subsequent accident.
	Flying road side unit [4]	UAVs aware a construction or an emergency in advance and quickly alerts it to cause a pre-action of the following vehicles.
	Flying police eye [4]	UAVs obtain information of illegally running vehicles and support the proactive safety enforcement of police.
Disaster relief	Refuge support in disasters [5]	UAVs monitor the status of nuclear power plant under an earthquake or a tsunami and support quick refuge of resident and aid goods.
	Response to sea contamination accidents	UAVs monitor the status of sea contamination by oil or chemicals and propagate the status to lead to a quick and proper reaction.
	Mountain disaster relief [6]	UAVs recognize the injured part of the wounded in a mountain, and deliver the first aid kit before a helicopter arrives
Surveillance	Disaster surveillance	UAVs surveil infrastructure, wild fires and leak of oil and chemicals, and respond to them proactively.
	Terror surveillance [7]	UAVs recognize a terrorist in the crowd and help the police remove the terror risk proactively.
Public safety	Crime prevention in blind spots	UAVs film a crime scene in a blind spot from CCTV and alert a crime outbreak to the police.
	Helper for the old and the weak	UAVs quickly alert an emergency of the old or the weak to a related authority to lead a timely response.
Extension of wireless communication service	UAV-aided ubiquitous coverage [10–14]	UAVs provide seamless wireless coverage under an infrastructure damage or a natural disaster.
	UAV-aided relaying [10]	UAVs provide wireless connectivity between two distant users.
	UAV-aided information dissemination and collection [15–18]	UAVs disseminate (collect) delay-tolerant information to (from) a large number of distributed wireless devices.

[15–18] as shown in the bottom of Table 1. More specifically, the UAV-aided information dissemination and collection can be substantiated in the form of UAV-based WSN (Wireless Sensor Network) in real world. In the UAV-based WSN, UAVs fly over sensors deployed in the field to acquire data from them and often provide critical information to them. Papers [19, 20] proposed a reliable and energy-efficient data collection scheme for a UAV-based WSN. Paper [19] jointly optimizes wake-up schedules of nodes and a UAV's trajectory to achieve reliable and energy-efficient data collection under fading channels while paper [20] determines optimal locations of multiple UAVs and their mobility patterns for achieving the same aims. However, both schemes are based on a flat network where a UAV communicates with a sensor in a P2P manner. In the UAV-aided information dissemination and collection, the communication between a UAV and the sensors should be performed not in a P2P manner but in a multicast or broadcast manner. This is because the number of UAVs is much smaller than the number of sensors, and the UAVs hover over the sensors, collecting information from the sensors or disseminating information to the sensors. Therefore, combining the network of sensor nodes into some logical groups and making only the group leaders communicate with a UAV are preferred for saving energy consumption of the sensor nodes and extending the network lifetime. Combining neighbouring sensors into a logical group is referred to as clustering.

Clustering is divided into cluster formation and cluster head election. Making a logical group of neighbouring nodes is called the cluster formation and the group is called a cluster. Contrarily, electing a leader in a cluster which plays as an information collector or disseminator is referred to as the cluster head election. The cluster formation and the cluster head election are complementary to each other. In some clustering schemes [21–40], cluster heads are first elected and the cluster formation is completed after a CH broadcasts a declaration message and its members respond to it via a join message. Hereafter, we refer to these cluster formation schemes as CH-first schemes. In other clustering schemes [41–45], clusters are first formed through exchange of some messages and CHs of the clusters are later elected on the basis of a criterion or multiple criteria. These cluster formation schemes are referred to as cluster-first schemes, hereafter. In a clustered WSN, if a compromised node is elected as a CH, this node can not only illegally obtain data from normal nodes but also forge data delivered to the sink. Furthermore, because attackers can grasp the control of the whole network by compromising a small number of CHs, compromising all CHs is a very attractive target for them [42]. Because a UAV-based WSN also suffers from the same vulnerabilities, secure CH election for the UAV-based WSN is essential for its successful operation. Up to now, a lot of research works dealing with CH election security for general WSNs [26–31, 41–45] have been proposed. However, to our best knowledge, there is no study dealing with CH election security for a UAV-based WSN. If a UAV has no impact on the secure CH election for the UAV-based WSN, we can choose one of a lot of secure CH election schemes which are employed for general WSNs.

However, in those schemes, compromised nodes declare themselves as CHs regardless of their qualification and no one can prevent them from doing so. If a UAV is employed for collecting critical data from nodes, it can deliver the collected data to the sink. Then, the sink decides which nodes should be removed from CH candidates and which nodes should be elected as CHs using the collected data. Because the election or the filtration result is broadcasted to the whole network, any CH declaration of a compromised node can be ignored by normal nodes. To this end, we devise a UAV-aided CH election framework and adapt existing security-driven CH election schemes into the framework in this paper.

The organization of this paper is as follows. We first propose a new UAV-aided CH election framework for the UAV-based WSN in Section 2. In Section 3, we classify the existing security-driven CH election schemes for the general WSN into some categories and select some representative schemes for each category. Next, we provide a brief review of each representative scheme and explain how to change each scheme in order to support the new UAV-aided CH election framework. In Section 4, we pick up some desirable security properties for CH election and compare all representative schemes of each category in terms of the properties. Then, we compare communication and computation overhead of all representative schemes. Section 5 deals with how to extend a UAV's fligth time in a UAV-based WSN. We conclude this survey paper in Section 6.

2. A UAV-Aided CH Election Framework for a UAV-Based WSN

2.1. Assumptions. Before describing the UAV-aided CH election framework, we assume the following to facilitate the quick comprehension of the framework.

First of all, we assume that the position of all nodes is known to the sink in advance. A UAV also can get the position of a node with which it will communicate via the sink before flight.

Second, any communication between a member and its CH is protected by encryption and decryption with a pairwise key which was established between them. For the key establishment, some keys are predistributed from the sink to each node before the deployment of nodes, and they are employed when a pairwise key is required between any two nodes. That is, if any two nodes share at least one predistributed key, they can establish a pairwise key using those shared keys. Even if they share no predistributed keys, a pairwise key can be indirectly established through a proxy node which shares any predistributed key with both nodes. Up to now, a lot of research works dealing with key establishment for general WSNs [46–70] have been proposed. We can use one of them to find common predistributed keys or a proxy node and to establish a pairwise key between them.

Third, we assume that a cluster-first scheme is employed for generating clusters without a CH in the network. In a CH-first scheme, whenever a new CH is elected, the cluster membership also changes accordingly. Once the cluster membership is changed, the new CH should establish a pairwise key with any other member to protect confidentiality and integrity of data exchanged between them. If these pairwise key establishments occur periodically, it is quite burdensome to an energy-constrained sensor node. Contrarily, a cluster-first scheme makes all nodes in a cluster establish pairwise keys between them during or after the cluster formation. After the initial pairwise key establishments, even though a new CH is elected in the cluster, the cluster membership is never changed. That is, because a new CH election never triggers following key establishments between nodes, a cluster-first scheme is efficient in terms of energy consumption. Up to now, some cluster formation schemes [31–34] were proposed to securely generate clusters and preserve the membership of the clusters against attackers trying to devastate the membership of the clusters. Among them, because Liu's scheme [32] assumes fixation of CH role nodes, it cannot be suited with our framework which facilitates periodic reselection of CHs. Remaining schemes only focus on formation and verification of cluster membership and never deal with pairwise key establishments for communication. Contrarily, because Wang's scheme [40] simultaneously attains the formation of clusters and the pairwise key establishments between members in each cluster, we choose it as the cluster formation scheme for our framework. After the initial cluster formation, member nodes in a cluster can generate a unique group key using common predistributed keys among them. A group key is employed for secure communication between members of a cluster and the sink.

2.2. A UAV-Aided CH Election Framework. Now, we explain the operation timeline of a UAV-aided CH election framework. First, the operation of a UAV-aided CH election framework is divided into rounds. Each round starts with an election phase during which each CH is elected and ends with a transmission phase during which data is transferred from the member nodes to their CH and to the UAV. A transmission phase is divided into multiple frames, and a frame is divided into multiple transmission slots, one UAV tour period and one broadcast time of a node list. During a frame, each member node transmits its reading to its CH in its assigned transmission slot. During a UAV period, a UAV visits all clusters to obtain data from the CHs and submits the collected data to the sink. At the end of each frame, the sink generates a list of new CHs and advertises it to all nodes. At the end of the last frame, the sink generates a list of disqualified members and advertises it to all nodes. Figure 1 shows the operation timeline of the UAV-aided CH election framework.

Based on the operation timeline, we minutely describe the UAV-aided CH election framework in the following. First, during an election phase, member nodes in each cluster elect their CH according to their embedded CH election protocol, and a UAV visits each cluster to obtain the ID of its CH from a member node. To this aim, the UAV selects a member node with which it will communicate in each cluster, and they encrypt and decrypt any message exchanged between them using the pairwise key shared with the sink. Assuming the employment of Wang's scheme [40] for initial cluster

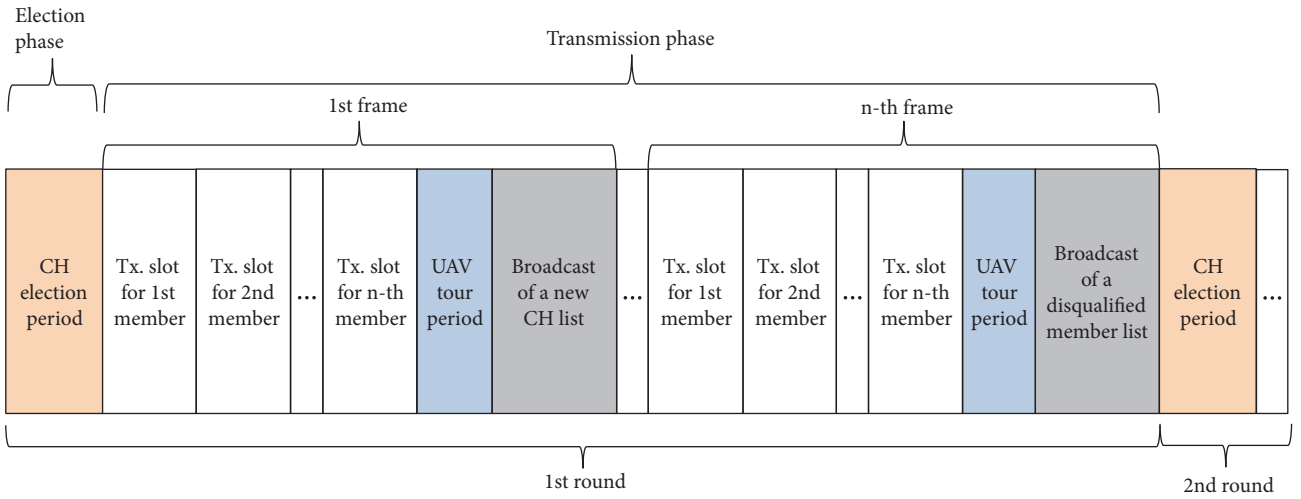


Figure 1: Operation timeline of a UAV-aided CH election framework.

formation, the UAV should communicate with the sector manager in each cluster because only sector managers are known to the sink and consequently to the UAV. Here, a sector manager means a node which was elected as a CH during the initial cluster formation. Note that a sector manager can generate a pairwise key with the sink using all predistributed keys. Besides, a UAV can acquire the pairwise key shared between the sink and the sector manager from the sink before flight.

During each transmission phase, each member node transmits a pair of {data, attribute} to its CH during its assigned time slot in a frame. Note that the transmission of a pair is protected by encryption and decryption using the shared pairwise key. Here, the attribute is a criterion for CH election or a node's residual energy. If an attribute should be considered for election of new CHs during a whole round, each member should transmit the attribute to its CH along with its data in the first frame. In the rest frames, the residual energy of nodes should be delivered to their CH to consider the energy consumption status to the election of new CHs. If an attribute takes no effect on the election of new CHs throughout all frames, each member should transmit its residual energy to its CH during all frames to consider the energy consumption distribution for the election of new CHs.

During each UAV tour period of a transmission phase, the UAV visits CHs and receives the {data, attribute} pairs of all member nodes from CHs. After visiting all CHs, the UAV delivers the pairs to the sink, and the sink selects new CHs considering the attributes. The sink encrypts the ID of each new CH with the cluster's group key and broadcasts the encrypted CH list in which each encrypted ID is enumerated in the order of the cluster's spreading code. Note that the spreading code was assigned to each cluster when each sector manager registered itself to the sink. The way of assigning a spreading code is very straightforward. The first sector manager to register was assigned the first code on a predefined code list, and the second sector manager to register was assigned the second code. Nodes pick up their cluster's item from the list and decrypt it with their group key to know which member will become the CH in the next frame. When the sink notifies that this is the last UAV tour period of a transmission phase, it takes a different action except for collecting data from nodes through a UAV. After visiting all CHs, the UAV delivers the pairs to the sink, and the sink selects disqualified members considering the attribute of all nodes. For instance, if we select residual energy as the attribute, the lowest energy nodes are selected as disqualified members.

We have two types of attributes employed in the UAV-aided CH election framework: a scheme-dependent attribute and a common attribute. The scheme-dependent attribute varies according to the election criterion of each scheme while the common attribute is fixed to the residual energy in this paper. If a scheme-dependent attribute has an impact on all frames, the sink considers the scheme-dependent attribute along with the residual energy to generate the list of new CHs or the list of disqualified members. Contrarily, if the scheme-dependent attribute has an impact on only the first frame, the sink considers the residual energy to generate the list of new CHs or the list of disqualified members. The list contains the IDs of the included members. Besides, because one list is generated for each cluster, the number of the lists is equal to the number of clusters. The sink first encrypts an ID list with the cluster's group key before appending the list to the network's list. Note that the network's list is the set of the encrypted ID lists, and each encrypted ID list is enumerated in the order of the cluster's spreading code. The sink distributes the network's list to the network in a broadcast manner. Upon receiving the network's list, nodes pick up their cluster's item from the list and decrypt it with their group key to know which member should be selected as a CH or which members should be removed from CH candidates in the next election.

Differences between the UAV-aided CH election framework and the general CH election framework are as follows. First, the UAV-aided CH election framework forces each node to transmit its attribute along with its data while the general CH election framework allows nodes to transmit its

data only. Second, the aggregation period of the general CH election framework is replaced with the UAV tour period in the UAV-aided CH election framework. In each UAV tour period, a UAV visits all clusters to collect data from the CHs while the general framework makes the CHs collect data from members and deliver the collected data to the sink. The UAV tour period also forces the sink to select new CHs for the next frame using the information provided by a UAV. To this end, during a transmission slot of a frame, each node transmits its data and attribute which is helpful for the sink to judge the qualification as a CH. The residual energy of a node is one of such attributes. Note that the attributes are collected to each CH along with data, and they are delivered to the sink through a UAV. Third, due to the second difference, the UAV-aided framework changes CH nodes after a frame period while the general framework changes CH nodes after a round period. The UAV-aided CH election framework yields the following benefits over the general CH election framework. First, because the sink selects the new CHs and advertises the new CH list to the network, an attacker can hardly predict or change any CH election result through compromised nodes. Second, because CH nodes are changed periodically even during a transmission phase, any damage caused by compromised CHs is diminished greatly. Third, because the periodic change of CH nodes is carried out by a UAV and the sink, the energy consumption of sensor nodes is greatly saved.

3. Review of Security-Driven CH Election Schemes and Application to the UAV-Aided CH Election Framework

3.1. Classification of Security-Driven CH Election Schemes for General WSNs. Because a lot of security-driven CH election schemes have been proposed up to now, we cannot review all of the schemes in this paper. For this reason, we categorize those schemes according to what they use to protect the CH elections and deal with each category and its representative schemes focusing on their adaption to the new UAV-aided CH election framework. We categorize the security-driven CH election schemes into predistributed key schemes [26, 27], random number schemes [42, 45], key chain schemes [28, 44], and cryptographic schemes [31, 41] in this paper. For each category of those schemes, we first reveal its basic operation and then explain some representative schemes. Then, we describe how to apply those schemes into the new UAV-aided CH election framework. First, we introduce an early CH election scheme which is called LEACH (Low-energy Adaptive Clustering Hierarchy) [21] to facilitate understanding of the following predistributed key schemes. In LEACH, all nodes have a CH winning probability and determine their CH role depending on this probability. The CH winning probability means a probability with which a node is elected as a CH. A node's CH winning probability increases whenever the node avoids a CH role during a given round. The probability becomes zero after the node became a CH in the previous round, and it increases again whenever the node avoids a CH role during a future round. Consequently, some nodes with a high CH winning probability declare themselves as CHs while the others join one of the declared CHs.

3.2. Predistributed Key Schemes. Ferreira et al. proposed F-LEACH [26] which protects a CH election in LEACH. Prior to the deployment, a specific number of keys are predistributed to nodes from a key pool of the sink. Then, a node with a high CH winning probability declares itself as a CH using the predistributed keys shared with the sink, and the sink authenticates the declaration using the same keys. The sink then distributes the list of the authenticated CHs to nodes using a source authentication scheme such as μTESLA [71]. Nodes which fail in declaring themselves as CHs join one of the authenticated CHs. However, this scheme does not force the CHs to authenticate the joining members. To resolve this problem, Oliveira et al. proposed SecLEACH [27] where normal nodes authenticate CH declarations using shared predistributed keys, and the CHs also authenticate the joining members using the same keys.

We can easily make the predistributed key schemes [26, 27] support the UAV-aided CH election framework by changing the following three things. First, in the transmission phase, each member transmits a pair of {data, attribute} instead of transmitting data only. In the predistributed key schemes, CH election results in the network depend on the CH winning probabilities of nodes. The reason why the predistributed key schemes choose the CH winning probability as the CH election criterion is to even energy consumption distribution among all members. That gives a rationale for us to choose the residual energy of each member as the CH election criterion in all frames. If we choose the residual energy of each member as the CH election criterion, there is no need to consider the CH winning probability in all frames. So, all members transmit their data and residual energy in their assigned time slot to reflect their energy consumption distribution. At the end of each frame, the sink considers residual energy of nodes to select a set of new CHs for the next frame. Second, a data aggregation period in the predistributed key schemes should be replaced with a UAV tour period to get benefits of the UAV-assisted CH election. That is, because the UAV visits all clusters and delivers the collected data to the sink on behalf of CHs, the CHs can avoid the energy-intensive long distant transmissions to the sink. Last, at the last UAV period of each transmission phase, the sink selects members with the lowest energy in a cluster as disqualified members to exclude them from the CH candidates. The sink securely informs each cluster of the disqualified members using encryption and decryption of a group key.

3.3. Random Number Schemes. Figure 2 shows the general operation of a basic random number scheme. In the basic random number scheme, each member in a cluster generates a random number and broadcasts it to other members as shown in Figure 2(a). As a result, all members in the cluster have the same list of random numbers and sum them up to create a common sum as shown in Figure 2(b). Each member divides the common sum by the number of

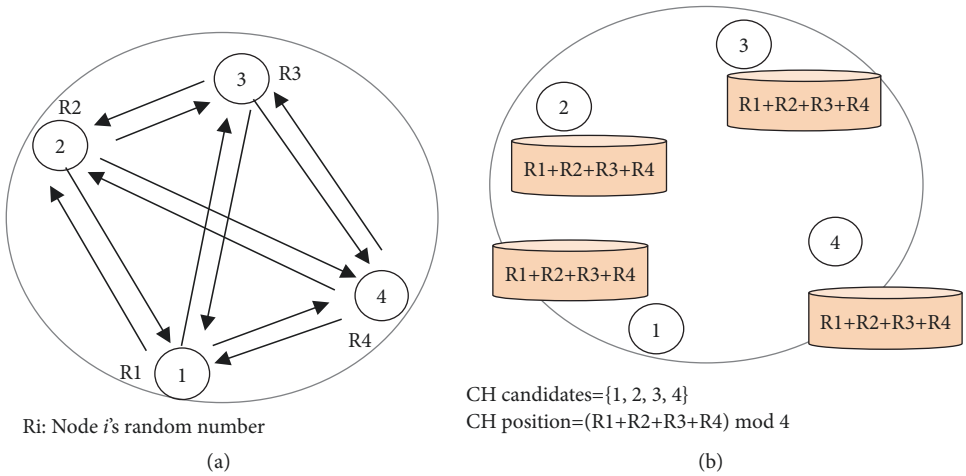


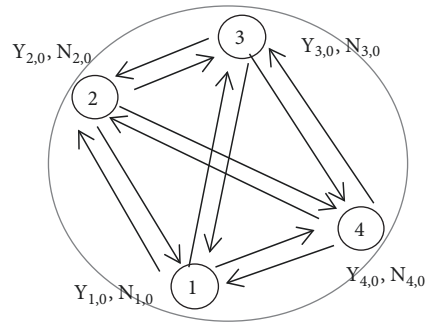
Figure 2: General operation of a basic random number scheme. (a) Broadcast of a random number. (b) Selection of a CH node.

members and settles the remainder as the position of the CH. Sirivianos et al. proposed three variations [45] of the basic scheme. They are the commitment-based scheme, the seed-based scheme, and Merkle's puzzle based scheme. They differ from each other in terms of how to create the common sum.

In the commitment-based scheme, each member encrypts a generated random number using pairwise keys shared with other members and then delivers each encrypted random number to other members in a P2P manner. Each member then delivers its random number to other members to convince them of the origin of its encrypted random number, and the common sum is generated by summing up the verified random numbers. In the seed-based scheme, each member first shares its seed number with other members by broadcasting it once. For each CH election, each member participates in the election by broadcasting a message. The message indicates the election participation of the sender in the CH election. The random number of each participant is generated by putting the election round number and the participant's seed number into a pseudo random number generator. Note that all nodes share the same pseudo random number generator. The common sum is then generated by summing up the random numbers of all participants. In Merkle's puzzle based scheme, an active CH establishes pairwise keys with other members using Merkle's puzzle [45]. Merkle's puzzle shows how any two nodes establish a pairwise key using predistributed keys without revealing the IDs of the predistributed keys. A member then generates a random number and encrypts it with a pairwise key shared with the active CH. Next, the member adds the encrypted random number to the common sum and delivers the common sum to one of other members. This procedure is repeated until all members add its encrypted random number to the common sum. The active CH then distributes the pairwise keys employed for the encryption of the random numbers to all members, and the members can transform the common sum (that is, the sum of the encrypted random numbers) into the sum of plain random numbers using only the pairwise keys.

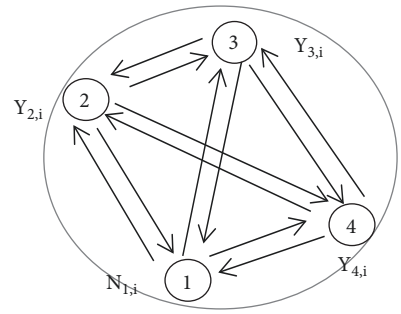
Wang et al. proposed a scheme [42] in which the trust values of all members are evaluated, and nodes with the lowest trust value are excluded from the CH candidates. Hereafter, we refer Wang's scheme to as the trust-based scheme. For each round, each member broadcasts its random number and monitors packet delivery frequency and last packet reception time of other members. Each member computes direct reputation values of other members considering the delivery success frequency and the recent delivery trends and then distributes the direct reputation values to other members. When a member evaluates an indirect reputation value of a different member, each direct reputation value that the evaluator gives to the other members is multiplied by each direct reputation value that other members give to the evaluator, and the multiplication results are averaged. In the same way, the evaluator can compute all indirect reputation values of other members. Combined reputation values are generated by adding the direct reputation values and their corresponding indirect reputation values. Each member's real reputation value is then obtained by averaging the combined reputation values that other members give to it. Finally, members whose real reputation value is lower than the average of the real reputation values are excluded from the CH candidates. The CH is selected among the survived candidates by dividing the sum of their random numbers by their population, and the remainder indicates the CH position in the survived candidates.

We can make the random number schemes work on the UAV-aided CH election framework by changing the following three things. First, in the transmission phase, each member transmits its attribute as well as its data. In the random number schemes except the trust-based scheme [42], a CH election result in a cluster depends on the random number of members. However, the random numbers are never used for the election of new CHs in all frames of a transmission phase. Therefore, in all frames, members transmit their data and residual energy to even energy consumption distribution among members. At the end of each frame, the sink selects the node with the highest residual energy as a CH in a cluster.



	Node			
	1	2	3	4
Last key of YES key chain	$Y_{1,0}$	$Y_{2,0}$	$Y_{3,0}$	$Y_{4,0}$
Last key of NO key chain	$N_{1,0}$	$N_{2,0}$	$N_{3,0}$	$N_{4,0}$

(a)



Round	Node				CH candidates
	1	2	3	4	
1	$N_{1,1}$	$Y_{2,1}$		$Y_{4,1}$	{2, 4}
2		$Y_{2,2}$	$Y_{3,2}$	$Y_{4,2}$	{2, 3, 4}
...	...	...	...	...	...

CH selection among candidates
- Select sequentially
- Comparing residual energy and distance to the sink and select the best candidate

(b)

Figure 3: General operation of a key chain scheme. (a) Sharing two last keys of all members. (b) CH election procedure in first two rounds.

In the trust-based scheme, a CH election result in a cluster relies on trust values of members. Note that the trust values have an impact on the election of new CHs in all frames. So, in the first frame, members in each cluster transmit their data and trust value to their CH. The sink selects the member with the highest trust value as a CH for the second frame in a cluster. In the rest frames, the members transmit their data and residual energy to their CH in order to reflect energy consumption distribution as well. Note that there is no need to transmit the trust values again because they have been delivered to the sink in the first frame. So, in the rest frames, the trust value is first considered for the new CH election, and the residual energy plays as a tie breaker. Second, replacing the data aggregation period with the UAV tour period mitigates the damage caused by the compromise of CHs between two CH election periods because the CHs are changed even during a transmission phase. Because a UAV relays the data collected from CHs to the sink, it saves the precious energy of sensor nodes by preventing the long distant transmissions of CHs. Third, at the last UAV period of each transmission phase, the sink selects members with the lowest energy in each cluster as disqualified members in all schemes except the trust-based scheme. In the trust-based scheme, the sink selects nodes whose trust value or residual energy is lower than a specified threshold as disqualified members. Then, the sink securely notifies each cluster of the disqualified members using encryption and decryption with a group key.

3.4. Key Chain Schemes. Before describing the key chain schemes, we reveal what a key chain is and how it is employed for secure communication. A key chain is a set of keys that are serially exploited to guarantee integrity or confidentiality of communication between any two nodes. Prior to communication, a sender first chooses a seed key K_n for a key chain and repeatedly applies a hash function F to the seed key. As a result, all other keys K_i are generated by computing $F(K_{i+1})$. The sender then securely delivers the last key of the key chain (K_0) to the receivers. Among the other keys, a key K_i is employed at the i-th transmission between the sender and the receivers to prove and verify the origin of a packet, respectively. That is, when K_i arrives at receivers, the receivers can verify if the packet's sender is the same as the originator of K_0 by checking whether $K_0 = F^i(K_i)$.

Figure 3 shows the general operation of a key chain scheme. Prior to deployment, each node i generates two key chains of Y_i and N_i. Here, Y_i is a YES key chain, and the j-th key of the key chain ($Y_{i,j}$) is used for expressing its participation intention in the j-th CH election round. Contrarily, N_i is a NO key chain, and the single NO key ($N_{i,1}$) is employed for requesting its exclusion from the CH candidates. Because the last keys of the two key chains ($Y_{i,0}$ and $N_{i,0}$) are concatenated to create the ID of node i, they are employed to verify the legality of each node in a CH election. First, each member advertises the last keys of the two key chains, as shown in Figure 3(a). In each election round, each member broadcasts the next YES key only when it wants to participate in the CH election. Otherwise, it broadcasts the single NO key. Members receiving the single NO key completely remove the sender from their CH candidates. If a member's message has not arrived at rest members during this election round, the rest of members temporarily exclude this member from the candidates. Note that the temporarily excluded member can join the next-round election unless its transmission failure exceeds a threshold. For example, even

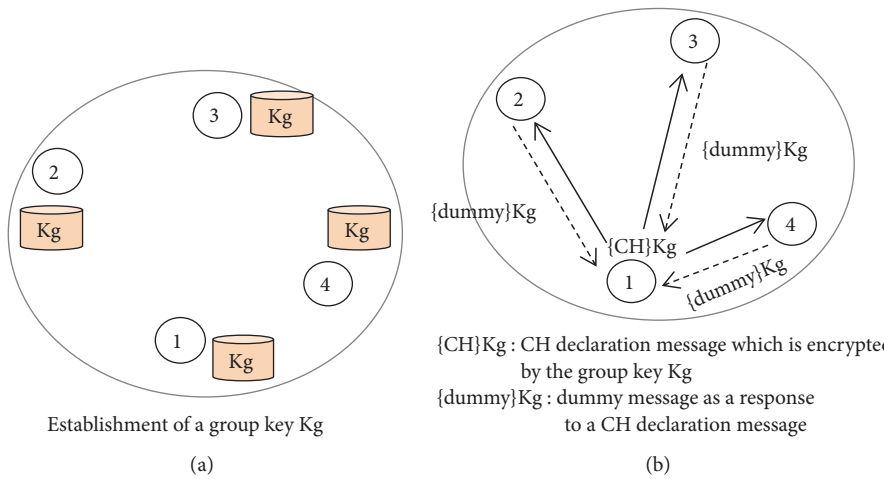


Figure 4: General operation of the symmetric cryptography scheme. (a) Group key generation. (b) CH election procedure in each round.

though a message from member 3 has not reached the rest members in the first round, it can participate in the second round by transmitting its next YES key ($Y_{3,2}$), as shown in Figure 3(b). The selection of a CH among the candidates is implemented by two methods. Paper [44] picks up a CH node sequentially in the candidate list. We refer this scheme to as Dong's scheme hereafter. Contrarily, paper [28] considers the residual energy and the distances from members to the sink and picks up the best candidate as a CH. We refer this scheme to as Han's scheme in this paper.

We can also make the key chain schemes work on the UAV-aided CH election framework by changing a few things. First, in a transmission phase, each member transmits its data and its attribute to its CH instead of transmitting its data only. In the key chain schemes, CH election results in the network depend on the election willingness of nodes, which is represented as a YES key in the original scheme. That is, the nodes which have transmitted a YES key are only selected as CH candidates. Therefore, in the first frame, members in each cluster transmit their data and election willingness indicator. The election willingness indicator can be represented as one bit (that is, one or zero). Among the CH candidates whose election willingness indicator is one, a CH is selected in a random manner by the sink at the end of the first frame. In the rest of frames, because the election willingness indicator plays the role of selecting CH candidates, we need to consider the indicators in later frames as well. However, there is no need to transmit them again because they have been delivered to the sink in the first frame. So, in the rest frames, members transmit their data and residual energy to even energy consumption distribution. At the end of each frame, the sink first picks up the CH candidates from members using the election willingness indicators and selects the node with the highest residual energy as a CH among the candidates. Second, replacing the data aggregation period with the UAV tour period facilitates the benefits of the UAV-aided CH election. Last, at the last UAV tour period of each transmission phase, the sink selects the members whose willingness indicator is zero and the members with the lowest energy in a cluster as disqualified members. Then, the sink securely reports the disqualified members to their affiliated cluster using encryption and decryption of a group key.

3.5. Cryptographic Schemes. In this paper, we classify cryptographic schemes into symmetric cryptography scheme [31] and discrete logarithm scheme [41]. The symmetric cryptography scheme [31] hides a CH election from external observers, whereas the discrete logarithm scheme [41] hides a CH election from both members and external observers.

In the symmetric cryptography scheme, each member first establishes a group key with other members jointly and employs the group key to hide an election process from external observers. Figure 4 shows the operation of the symmetric cryptography scheme. First, all members in a cluster establish a group key jointly, as shown in Figure 4(a). Next, a member which received no CH declaration message encrypts a CH declaration message with the group key and broadcasts it, as shown in Figure 4(b). The CH election procedure follows the "first-come-first-served" rule. If a member first declares itself as a CH, any other member which hears the declaration becomes the member of the declarer. The members receiving the CH declaration message decrypt the message, and settle the broadcaster as the CH. Next, the receivers encrypt a dummy message with the group key and broadcast it to prevent external observers from recognizing the CH through the previous declaration message, as shown in Figure 4(b). However, this scheme hides the election process from only external observers, and thus the election process is disclosed to an internal compromised member.

The discrete logarithm scheme highly relies on the difficulty of the discrete logarithm problem as its name suggests. The discrete logarithm problem is to find an integer n, if it exists, such that $g^n = h$ when we have a multiplicative group G and elements $g, h \in G$. Under this group, the difficulty in the discrete logarithm problem means that computing an integer n is difficult. In the discrete logarithm scheme, each member declares itself as a CH depending on a probability of being a CH which is called the self-election probability. However,

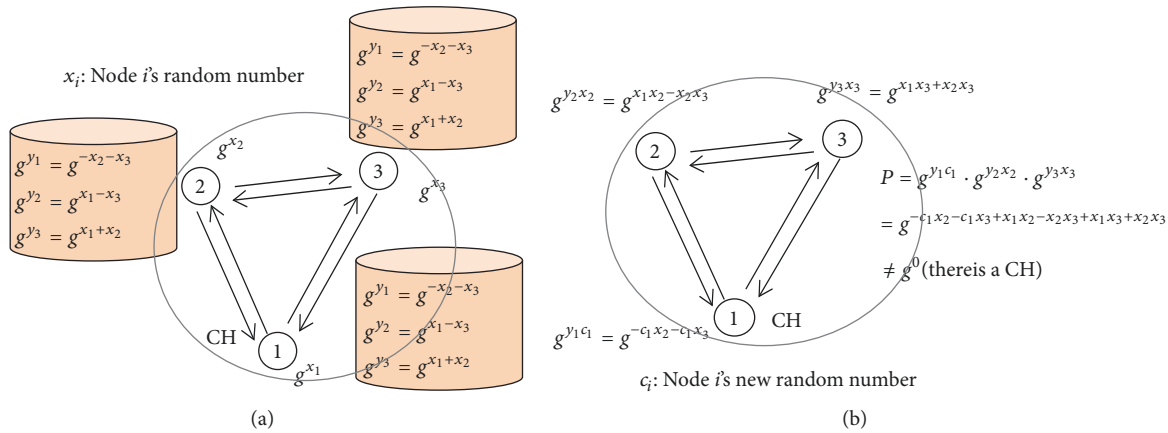


FIGURE 5: CH declaration check in the discrete logarithm scheme. (a) Generating public keys and evidences of holding public keys. (b) Existence check of a CH node.

because the CH declaration is never advertised outside of the node, both internal and external observers cannot know which node will be a CH, and they can check only if there is a CH declaration node. Figure 5 shows the procedure through which each member checks the existence of a CH in its cluster. During the procedure, whenever each member broadcasts a message, the message includes a knowledge proof that ensures the legal membership of the broadcaster in the cluster. However, we omit the real knowledge proof in each message for the sake of simplicity. First, each member i generates a random number (x_i) and broadcasts the random number's public key (g^{x_i}). Upon receiving it, a member computes the evidence of holding the public key (g^{y_i}) using (1). Thus, after receiving all public keys, members 1, 2, and 3 have g^{y_1}, g^{y_2}, and g^{y_3} as shown in Figure 5(a). Now, each member decides to elect itself as a CH or to give up a CH role according to its self-election probability. If it elects itself as a CH, it assigns a new random number to c_i as shown in Figure 5(b). Otherwise, it assigns the original random number (that is, x_i) to c_i. Next, each member i computes $g^{y_ic_i}$ and broadcasts it as shown in Figure 5(b). After receiving all $g^{y_ic_i}$, the members execute the determination equation of (2), and the result is not one if there is a CH as shown in Figure 5(b). We disclose the reason why a compromised member cannot know whether node 1 is a CH or not, even though it conforms to the protocol shown in Figure 5. If the compromised node wants to check whether node 1 declares as a CH or not, it should first extract x_1, x_2, and x_3 from g^{x_1}, g^{x_2}, and g^{x_3}. Then, the compromised node can check whether node 1 declared as a CH by computing the value of $g^{y_1c_1}$. In other words, if $g^{y_1c_1}$ is $g^{-x_1\ x_2-x_1\ x_3}$, the node 1 did not declare as a CH. Otherwise, it means that the node 1 declared itself as a CH. However, extracting x_1, x_2, and x_3 from g^{x_1}, g^{x_2}, and g^{x_3} is quite difficult owing to the difficulty of the discrete logarithm, as described earlier.

$$g^{y_i} = \frac{\prod_{j=1}^{i-1} g^{x_j}}{\prod_{j=i+1}^{N} g^{x_j}} \tag{1}$$

$$P = \prod_{i=1}^{N} g^{y_ic_i} \tag{2}$$

In the symmetric cryptography scheme, a CH election result in a cluster relies on the CH announcement probability of members. If a member obtains a chance to declare itself as a CH earlier than other members, it becomes a CH while other members become members of the CH. Because a CH is randomly selected by the CH announcement probabilities of members, the probabilities should have no impact on the election of new CHs in all frames. So, there is no reason to consider them again in all frames. For this reason, in all frames, members transmit their data and residual energy to even energy consumption distribution among them. At the end of each frame, the sink selects members with the highest residual energy in each cluster as CHs.

The discrete logarithm scheme determines a CH role considering the self-election probability of members in a cluster. If a node's self-election probability is higher than a threshold, it becomes a CH but never advertises its CH declaration. Therefore, in the first frame, members in each cluster transmit their data along with their self-election probability as an attribute. The sink selects the node with the highest self-election probability as a CH for the second frame. However, because the self-election probability relies on the randomness of the probability, it should have no impact on the election of new CHs in later frames. So, there is no reason to consider the self-election probability in later frames again. Therefore, in the rest frames, members send their data and residual energy to even energy consumption distribution over members. At the end of each frame, the sink selects the node with the highest residual energy as a CH among the members. For both cryptographic schemes, the data aggregation period is replaced with the UAV tour period to get various benefits of the UAV-assisted CH change. For both schemes, at the last UAV period of each transmission phase, the sink picks up members whose residual energy is lower than a specified threshold and designates them as disqualified members. Then, the sink securely notifies the

Table 2: Adaption of security-driven CH election schemes into the UAV-aided CH election framework.

Category of CH election schemes		Attribute transmitted during the first frame	Criterion for selecting a new CH for post-1st frame	Criterion for selecting disqualified members
Pre-distributed key schemes	F-LEACH	Residual energy	Residual energy	Residual energy
	SecLEACH	Residual energy	Residual energy	Residual energy
Random number schemes	Commitment-based scheme	Residual energy	Residual energy	Residual energy
	Seed-based scheme	Residual energy	Residual energy	Residual energy
	Merkle's puzzle based scheme	Residual energy	Residual energy	Residual energy
	Trust-based scheme	Trust value	Trust value, residual energy	Trust value, residual energy
Key chain schemes	Dong's scheme	Election willingness indicator	Election willingness indicator, residual energy	Election willingness indicator, residual energy
	Han's scheme	Election willingness indicator	Election willingness indicator, residual energy	Election willingness indicator, residual energy
Cryptographic schemes	Symmetric cryptography scheme	Residual energy	Residual energy	Residual energy
	Discrete logarithm scheme	Self-election probability	Residual energy	Residual energy

disqualified members to each cluster using encryption and decryption with a group key.

3.6. Adaption into the UAV-Aided CH Election Framework. In this subsection, we reveal the differences in adapting the security-driven CH election schemes into the UAV-assisted CH election framework in Table 2. First of all, the attribute transmitted during the first frame varies in line with each scheme while the attribute transmitted during the rest frames (that is, residual energy) remains unchanged regardless of scheme. Even if an attribute affects all frames of a transmission phase, it is enough for members to transmit it only in the first frame. During the UAV tour period of the first frame, a UAV gets the attributes of all nodes visiting all CHs and delivers the attributes to the sink. Consequently, the sink repeatedly employs them to select new CHs for the next frame. Most of the schemes employ the residual energy as the criterion for selecting disqualified members while only a few schemes employ an extra attribute as well as the residual energy. The reason why they employ an extra attribute for selecting disqualified members is that the extra attribute has an impact on qualification of members as well. For instance, the trust-based scheme removes some members whose trust value is lower than a specified threshold while two key chain schemes (that is, Dong's scheme and Han's scheme) remove some members whose election willingness indicator is zero. Contrarily, because a probability or a random number never affects qualification of members in other schemes, the residual energy is enough for selecting disqualified members in those schemes.

4. Security Comparison

4.1. Desirable Security Properties for a CH Election. Desirable security properties for a CH election are independent from the type of a WSN. That is, desirable security properties for CH election in a general WSN can be also employed for a UAV-based WSN. This is because electing a CH in a cluster is finally to pick up the best member by comparing a specific attribute value among the cluster members. For this reason, we have identified some of the desirable security properties through a careful review of some previous work which were proposed for general WSNs [41, 42, 45, 72]. Considering these properties, we chose some essential properties that are employed in this survey according to the following criteria. First, we chose properties with commonalities out of all properties. Next, we excluded reliability-driven properties such as termination and completeness to focus on the security of CH elections.

First, unpredictability means that a member in a cluster cannot predict the winner of a CH election before the end of the election. Without this property, malicious nodes can compromise the predicted CHs during the election process. Besides, they can try to change a predicted CH or to make multiple CHs in a cluster. Next, nonmanipulability means that a member in a cluster cannot modify a CH election result. Without this property, malicious nodes can prevent a legitimate node from being a CH, or they can make one of the malicious nodes become a CH through their collusion. The agreement property expresses that all members in a cluster have the same CH election result. A cluster having multiple CHs damages the agreement property. If some members accept a new CH besides an existing CH, it also means that the nonmanipulability of their election result is paralyzed. Immunity against loss means that message losses during a CH election have little impact on the nonmanipulability and the agreement property. Without this property, a malicious node can hide its misbehaviour through the message losses. Finally, privacy means that the CH election process is hidden from observers within and around a cluster. Without this property, malicious nodes can identify an elected CH during an election process and try to compromise the identified CH during the transmission phase. Besides, because they know an anticipated CH, they try to manipulate the election result

so that a different member becomes a CH, or multiple nodes play as CHs in the cluster.

4.2. Security Comparison of the Security-Driven CH Election Schemes. In this subsection, we provide security comparison of the security-driven CH election schemes which we explained in the Section 3. We compare them considering how well they satisfy the desirable security properties for CH elections.

4.2.1. Predistributed Key Schemes. In the predistributed key schemes, as long as a node shares at least one key with potential CHs, it can easily identify that the potential CHs will become a CH. Their unpredictability is therefore low. In addition, a malicious node can declare itself as a CH even though it has a CH within its vicinity. Thus, their agreement property is also low. Because they have no recovery mechanism when messages are often lost, their immunity against loss is also low. Finally, because a CH election process is revealed to nodes that have common keys, their privacy is low.

4.2.2. Random Number Schemes. Among the random number schemes, the commitment- and the seed-based scheme provide a low level of unpredictability and nonmanipulability, whereas their agreement property is medium. The low unpredictability and nonmanipulability are caused by the fact that a compromised node can easily recognize which node will be a CH by delaying its message transmission. In other words, if a compromised node delays its message transmission to the last, it can predict the election result by summing up random numbers of all members and dividing the sum by the number of members. The compromised node may try to change the result by modifying its random number before transmitting it. If a malicious member transmits its message to only a part of members, the CH election result is split into two different results. However, if members share their received messages, such a malicious behaviour can be easily exposed to normal members. Then, the normal members can expel those malicious members from the cluster. That is the reason why we set their level of agreement property to medium. Furthermore, because the commitment- and the seed-based scheme have no mechanism to compensate for message losses, their immunity against message loss is low. Merkle's puzzle based scheme provides higher security than the commitment- and the seed-based scheme. This is because only the active CH knows a CH election result, and it is hard for the active CH to change the CH election result in such a short time. Because a CH election result is induced through an encrypted sum shared among members, it is hard to break the agreement property of the encrypted sum. Furthermore, because it has a mechanism to compensate consecutive reception failure of keys required for decryption of the encrypted sum, it provides better immunity against message loss than the commitment- and the seed-based scheme. In the trust-based scheme, compromised nodes trying to damage the nonmanipulability and the agreement property are given a low trust value, and they hardly get a chance to modify any CH election result. Because the scheme allows a delivery failure member to participate in the next election again, its immunity against message loss is high. However, because the trust values of members are distributed in plaintext, an observer can easily predict which node will be a CH node. In all random number schemes, because the election process is exposed to all members in a cluster, their privacy is low.

4.2.3. Key Chain Schemes. In Dong's scheme, all members in a cluster have a common list of CH candidates, and a CH is selected sequentially from the list. Therefore, the scheme's unpredictability is low. In addition, a compromised node can even change a CH election result by avoiding its message transmission, and thus the scheme's nonmanipulability is also low. Dong's scheme lowers the agreement property because it allows a compromised node to split a CH election result into multiple results by selectively transmitting a message. Fortunately, it has an algorithm for merging multiple results into a single result to compensate its low agreement property. To deal with a message loss, Dong's scheme allows a transmission failure node to get a second chance to transmit its message. Because Han's scheme combines Dong's scheme into LEACH, it inherits the properties of both schemes. That is, Han's scheme allows a normal node to easily predict a CH node as in LEACH, and thus its unpredictability is also low. In addition, because Han's scheme allows a compromised node to change a CH election result by avoiding its message transmission as in Dong's scheme, its nonmanipulability is low as well. In terms of immunity against message loss, Han's scheme compensates a message loss through redundant message transmissions and providing multiple chances as in Dong's scheme. However, unlike Dong's scheme, Han's scheme has no election merge algorithm, and thus its agreement property is lower than that of Dong's scheme. Finally, the privacy of both schemes is low because the election process is open to all nearby observers in Dong's scheme and to the internal members in Han's scheme.

4.2.4. Cryptographic Schemes. In the symmetric cryptography scheme, because a compromised member that holds a shared group key can easily predict which member will become a CH, the scheme's unpredictability is low. However, an internal member holding the group key as well as an attacker cannot modify a CH election result because the election result depends on the self-election probability of a member. Because a compromised member that holds a shared group key can illegally declare itself as a CH, the scheme's agreement property is also low. Moreover, because the scheme has no recovery mechanism when messages are lost, its immunity against message loss is low. The privacy of the symmetric cryptography scheme is low because its election process is revealed to all internal members. The discrete logarithm scheme offers the highest unpredictability over other schemes because the identity of a CH node is never revealed even to internal members. For the same reason, a compromised member cannot prevent a legal member from declaring itself as a CH, and thus its nonmanipulability is high. Contrarily, a compromised member can

Table 3: Security comparison of security-driven CH election schemes.

Category of security-driven CH election schemes		Unpredictability	Non-manipulability	Agreement property	Immunity against message loss	Privacy
Pre-distributed key schemes	F-LEACH	Low	High	Low	Low	Low
	SecLEACH	Low	High	Low	Low	Low
Random number schemes	Commitment-based scheme	Low	Low	Medium	Low	Low
	Seed-based scheme	Low	Low	Medium	Low	Low
	Merkle's puzzle based scheme	Medium	High	High	Medium	Low
	Trust-based scheme	Low	High	High	High	Low
Key chain schemes	Dong's scheme	Low	Low	Medium	Medium	Low
	Han's scheme	Low	Low	Low	Medium	Low
Cryptographic schemes	Symmetric cryptography scheme	Low	High	Low	Low	Low
	Discrete logarithm scheme	High	High	Low	Low	High

illegally declare itself as a CH without any permission, and thus it can easily break the agreement property. Moreover, because it has no compensation mechanism against message losses, its immunity against message loss is low. The privacy of the discrete logarithm scheme is high because it hides the election process not only from external observers but also from internal members. Table 3 summarizes the security comparison of the security-driven CH election schemes.

4.3. Overhead Comparison of the Security-Driven CH Election Schemes. In this section, we provide overhead comparison of the security-driven CH election schemes which we explained in Section 3. The overhead of the security-driven CH election schemes is divided into the communication and the computation overhead. For the sake of convenience of comparison, the magnitude of any overhead is restricted to extent that each scheme causes during a CH election in a cluster. The communication overhead indicates the number of messages sent between members in a cluster during a CH election period, and it is represented as the big O notation. Note that the communication overhead includes only the communications caused during the election phase and excludes the communications caused during the transmission phase. This is because the communications caused during the transmission phase are equal over all schemes even though the size of a message which is sent to the CH may vary according to each scheme.

The computation overhead indicates the number of computational operations caused by members in a cluster during an election, and it is also represented as the big O notation. Note that the computation overhead includes only computational operations performed by members during the election phase, and it excludes computational operations caused during the transmission phase. Before starting the comparison, we assume the following. First, a symmetric key encryption or decryption in this paper indicates the encryption or the decryption caused by a stream cipher such as RC4. Next, a hash operation in this paper indicates the hash operation caused by SHA-1.

4.3.1. Predistributed Key Schemes. F-LEACH causes the communication overhead of $O(n)$ where n is the number of members in a cluster. SecLEACH brings about the same amount of communication overhead as F-LEACH. Concerning the computation overhead, both F-LEACH and SecLEACH induce $O(n)$ hash operations.

4.3.2. Random Number Schemes. The commitment scheme induces $O(n^2)$ of communication overhead while other schemes of the same category down their communication overhead to $O(n)$. In terms of computation overhead, the commitment scheme causes $O(n^2)$ symmetric key encryptions/decryptions and $O(n)$ random number generations while the seed-based scheme induces $O(n^2)$ arithmetic operations and $O(n^2)$ random number generations. The computation overhead of Merkle's puzzle based scheme consists of $O(n)$ symmetric key encryptions/decryptions, $O(n)$ arithmetic operations, and $O(n)$ random number generations. The trust-based scheme induces $O(n^2)$ arithmetical operations and $O(n)$ random number generations.

4.3.3. Key Chain Schemes. Dong's scheme and Han's scheme cause the equal communication overhead of $O(n)$. In terms of computation overhead, Dong's scheme causes $O(n^2)$ hash operations and $O(n^2)$ arithmetical operations while the overhead of Han's scheme consists of $O(n^2)$ hash operations, $O(n^2)$ symmetric key encryptions/decryptions, and $O(n)$ arithmetic operations.

4.3.4. Cryptographic Schemes. The discrete logarithm scheme and the symmetric cryptography scheme cause the communication overhead of $O(n)$. In terms of computation overhead, the symmetric cryptography scheme requires $O(n)$ symmetric key encryptions/decryptions while the overhead

Table 4: Communication overhead of security-driven CH election schemes.

Category of security-driven CH election schemes		Communication overhead
Pre-distributed key schemes	F-LEACH	$O(n)$
	SecLEACH	$O(n)$
Random number schemes	Commitment-based scheme	$O(n^2)$
	Seed-based scheme	$O(n)$
	Merkle's puzzle based scheme	$O(n)$
	Trust-based scheme	$O(n)$
Key chain schemes	Dong's scheme	$O(n)$
	Han's scheme	$O(n)$
Cryptographic schemes	Symmetric cryptography scheme	$O(n)$
	Discrete logarithm scheme	$O(n)$

Table 5: Computation overhead of security-driven CH election schemes.

Category of security-driven CH election schemes		Computation overhead
Pre-distributed key schemes	F-LEACH	$O(n)$ hash operations
	SecLEACH	$O(n)$ hash operations
Random number schemes	Commitment-based scheme	$O(n^2)$ symmetric key encryptions/decryptions+ $O(n)$ random number generations
	Seed-based scheme	$O(n^2)$ arithmetic operations+ $O(n^2)$ random number generations
	Merkle's puzzle based scheme	$O(n)$ symmetric key encryptions/decryptions+ $O(n)$ arithmetic operations+ $O(n)$ random number generations
	Trust-based scheme	$O(n^3)$ arithmetic operations+ $O(n)$ random number generations
Key chain schemes	Dong's scheme	$O(n^2)$ hash operations + $O(n^2)$ arithmetic operations
	Han's scheme	$O(n^2)$ hash operations+ $O(n^2)$ symmetric key encryptions/decryptions+ $O(n)$ arithmetic operations
Cryptographic schemes	Symmetric cryptography scheme	$O(n)$ symmetric key encryptions/decryptions
	Discrete logarithm scheme	$O(n)$ modular exponentiations+ $O(n)$ hash operations+ $O(n)$ arithmetic operations

of the discrete logarithm scheme consists of $O(n)$ modular exponentiations, $O(n)$ hash operations, and $O(n)$ arithmetic operations.

Table 4 shows the communication overhead of security-driven CH election schemes. As shown in Table 4, the commitment-based scheme induces much higher overhead than other schemes, and the overhead of other schemes is similar in terms of the big O notation.

Table 5 shows the computation overhead of security-driven CH election schemes. We first take into account computation overhead that a single computational operation causes. Then, we consider the total computation overhead that each scheme causes. Note that the computation overhead of a modular exponentiation is significantly higher than that of a hash operation [41]. Besides, the computation overhead of a hash operation such as SHA-1 is higher than that of a symmetric key encryption/decryption such as RC4 [41]. In addition, the computation overhead of a symmetric key operation is higher than that of an arithmetic operation while the computation overhead of a random number generation is least compared to other computational operations. Thus, we have an inequality with respect to the computation overhead of the computational operations: modular exponentiation $\gg$ hash operation > symmetric key encryption/decryption > arithmetic operation $\gg$ random number generation. According to the inequality, the discrete logarithm scheme causes the highest computation overhead over other schemes, and other schemes compete with each other. Because a hash operation induces more overhead than a symmetric key encryption/decryption, Han's scheme, Dong's scheme, and the commitment-based scheme follow the discrete logarithm scheme serially. Even though F-LEACH and SecLEACH seem to induce the same amount of computation overhead, their real overhead is n hash operations and $2n - 2$ hash operations, respectively. Therefore, SecLEACH causes more overhead than F-LEACH. It is very straightforward that the commitment-based scheme induces more overhead than the symmetric cryptography scheme and Merkle's puzzle based scheme. When it comes to the symmetric key encryptions/decryptions, Merkle's puzzle based scheme causes $((m+3)n - m - 1)$ encryptions/decryptions while the symmetric cryptography scheme causes $3n - 2$ encryptions/decryptions. Because m is larger than one, Merkle's puzzle based scheme causes more overhead than the symmetric cryptography

scheme. In terms of the arithmetic operation, the seed-based scheme induces n^2 arithmetic operations while the trust-based scheme causes $(3n^3 + 5n^2 - 3n)$ arithmetic operations. It makes the computation overhead of the trust-based scheme bigger than that of the seed-based scheme.

5. Extension of UAV Flight Time

Because a UAV's flight time is less than 60 minutes due to its high energy consumption and a limited battery power, extending the UAV's flight time is very important to a UAV-based WSN. One way to extend the UAV's flight time is to make an optimal flight plan of the UAV to save its precious energy. Papers [9, 17] deal with this issue. Another way to extend a UAV's flight time is to make multiple UAVs serve for the UAV-based WSN simultaneously. Because each UAV's served area is reduced, it flight time and energy consumption also can be reduced. Paper [20] covers the use of multiple UAVs and impact of their mobility patterns on the performance. Besides above two solutions, deploying some recharging stations and making them recharge UAVs having a little amount of energy is an interesting alternative. Recently, paper [73] presents a scheme which deploys some recharging stations in a mission area and makes an optimal flight plan including those recharging stations. Employment of the recharging stations is also helpful when a UAV is suffering from a bad weather condition such as strong wind.

6. Conclusions

In this survey, we introduce UAS and its promising applications and reveal the importance of integrating the UAS into a WSN which is called the UAV-based WSN. Then, we explain the benefits of clustering in a UAV-based WSN and the necessity of secure CH election in such a network. Then, we propose a UAV-aided CH election framework to employ a UAV for securing the CH elections in the UAV-based WSN. First, we classify the security-driven CH election schemes for a general WSN into several categories and explain the principle of each category and some representative schemes of each category. Then, we deal with how to adapt them into the UAV-aided CH election framework. We identify some desirable security properties that a CH election scheme should have and compare the security-driven CH election schemes in line with the desirable security properties. Last, we compare the communication and the computation overhead of the security-driven CH election schemes in terms of the big O notation.

According to the desirable security properties, we found that some schemes are not suitable in terms of security as shown in Table 3. More specifically, the commitment- and the seed-based scheme, Dong's scheme, and Han's scheme provide a low security level for almost all properties and a medium security level for just one or two properties. Alike, F-LEACH, SecLEACH and the symmetric cryptography scheme provide a low security level for almost all properties while guaranteeing a high level security for just one property. Consequently, Merkle's puzzle based scheme, the trust-based scheme, and the discrete logarithm scheme are remained, and they have respective superiority over other competitors. First, Merkle's puzzle based scheme and the trust-based scheme are superior to the discrete logarithm scheme in terms of the agreement property and the immunity against message loss. Contrarily, the discrete logarithm scheme is superior to Merkle's puzzle based scheme and the trust-based scheme in terms of the unpredictability and the privacy. Note that none of all schemes guarantees all desirable security properties at a high level. We therefore need to select an election scheme according to which security properties should be first considered in the environment where sensor nodes work.

As a future work item, we plan to implement the security-driven CH elections schemes that we dealt with in this paper in a simulation environment and to do a series of simulations to compare their security and overhead rigorously.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the ICT R&D program of MSIP/IITP [R0126-18-1005, Development of High Reliable Communications and Security SW for Various Unmanned Vehicles].

References

[1] D. C. Iannicca, J. H. McKim, D. H. Stewart, S. K. Thadhani, and D. P. Young, "Control and Non-payload Communications (CNPC) Prototype Radio – Generation 2 Security Architecture Lab Test Report," Tech. Rep. NASA/TM-2015-218453, May 2015.

[2] D. C. Iannicca, J. A. Ishac, and K. A. Shalkhauser, "Control and Non-payload Communications (CNPC) Prototype Radio – Generation 2 Security Flight Lab Test Report," Tech. Rep. NASA/TM-2015-218821, Security Flight Lab, 2015.

[3] N. Hossein Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[4] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[5] T. Torri and Y. Sanada, "Radiation Measurement by Unmanned Aircraft after Fukushima Daiich Nuclear Power Plant Accident," in *Proceedings of the of Symp. ICAO*, Montreal, QC, Canada, 2015.

[6] T. Andre, K. A. Hummel, A. P. Schoellig et al., "Application-driven design of aerial communication networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 129–137, 2014.

[7] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-Based IoT Platform: A Crowd Surveillance Use Case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[8] E. Ackerman, "U.S. Marines Testing Disposable Delivery Drones," IEEE Spectrum, 2017, http://spectrum.ieee.org/automaton/robotics/drones/marines-testing-disposable-gliding-delivery-drones.

[9] J. Xu et al., "Animal Monitoring with Unmanned Aerial Vehicle-Aided Wireless Sensor Networks," in *Proceedings of the 40th Annual IEEE Conf. on Local Computer Networks (LCN, 2015)*, pp. 334–341, Clearwater Beach, Fla, USA, 2015.

[10] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[11] A. Orsino, A. Ometov, G. Fodor et al., "Effects of Heterogeneous Mobility on D2D- and Drone-Assisted Mission-Critical MTC in 5G," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 79–87, 2017.

[12] P.-V. Mekikis, E. Kartsakli, L. Alonso, and C. Verikoukis, "Flexible aerial relay nodes for communication recovery and D2D relaying," in *Proceedings of the 5th IEEE Global Conference on Consumer Electronics, GCCE 2016*, pp. 1-2, Kyoto, Japan, October 2016.

[13] P.-V. Mekikis, A. Antonopoulos, E. Kartsakli, L. Alonso, and C. Verikoukis, "Communication recovery with emergency aerial networks," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 3, pp. 291–299, 2017.

[14] K. Namuduri, "Flying cell towers to the rescue," *IEEE Spectrum*, vol. 54, no. 9, pp. 38–43, 2017.

[15] S. Say, H. Inata, J. Liu, and S. Shimamoto, "Priority-Based Data Gathering Framework in UAV-Assisted Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5785–5794, 2016.

[16] M. Dong, K. Ota, M. Lin, Z. Tang, S. Du, and H. Zhu, "UAV-assisted data gathering in wireless sensor networks," *The Journal of Supercomputing*, vol. 70, no. 3, pp. 1142–1155, 2014.

[17] D. Ho, E. I. Grotli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of Wireless Sensor Network and UAV Data Acquisition," *Journal of Intelligent Robotic Systems*, vol. 78, no. 1, pp. 159–179, 2015.

[18] D. T. Ho, E. I. Grølti, and T. A. Johansen, "Heuristic Algorithm and Cooperative Relay for Energy Efficient Data Collection with a UAV and WSN," in *Proceedings of the IEEE Intl Conf. on Computing, Management Telecommunications (ComManTel*, pp. 346–351, Ho Chi Minh City, Vietnam, 2013.

[19] C. Zhan, Y. Zeng, and R. Zhang, "Energy-Efficient Data Collection in UAV Enabled Wireless Sensor Network," *IEEE Wireless Communications Letters*, 2017.

[20] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[22] P. Ren, J. Qian, L. Li, Z. Zhao, and X. Li, "Unequal clustering scheme based LEACH for wireless sensor networks," in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing, ICGEC 2010*, pp. 90–93, chn, December 2010.

[23] V. Katiyar, N. Chand, G. C. Gautam, and A. Kumar, "Improvement in LEACH Protocol for Large-scale Wireless Sensor Networks," in *Proceedings of the 2011 Intl Conf. on Emerging Trends in Electrical and Computer Technology (ICETECT*, pp. 1032–1036, Tami Nadu, India, 2011.

[24] M. Saadat, R. Saadat, and G. Mirjalily, "Improving threshold assignment for cluster head selection in hierarchical wireless sensor networks," in *Proceedings of the 2010 5th International Symposium on Telecommunications, IST 2010*, pp. 409–414, irn, December 2010.

[25] S. H. Kang and T. Nguyen, "Distance based thresholds for cluster head selection in wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1396–1399, 2012.

[26] A. C. Ferreira, M. A. Vilaça, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. Loureiro, "On the Security of Cluster-Based Communication Protocols for Wireless Sensor Networks," in *Proceedings of the 4th IEEE Int'l Conf. on Networking, LNCS 3420*, vol. 3420 of *Lecture Notes in Computer Science*, pp. 449–458, Reunion Island, France, 2005.

[27] H. C. Leonardo B. Oliveira, "SecLEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks," in *Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)*, pp. 145–154, Cambridge, Mass, USA.

[28] Y. Han, M. Park, and T. Chung, "SecDEACH: Secure and Resilient Dynamic Clustering Protocol Preserving Data Privacy in WSNs," in *Computational Science and Its Applications – ICCSA 2010*, vol. 6018 of *Lecture Notes in Computer Science*, pp. 142–157, Springer, Berlin, Germany, 2010.

[29] N. Pissinou and G. V. Crosby, "Cluster-Based Reputation and Trust for Wireless Sensor Networks," in *Proceedings of the 2007 4th IEEE Consumer Communications and Networking Conference*, pp. 604–608, Las Vegas, NV, USA, January 2007.

[30] G. V. Crosby, N. Pissinou, and J. Gadze, "A Framework for Trust-based Cluster Head Election in Wireless Sensor Networks," in *Proceedings of the 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS)*, pp. 13–22, Columbia, MD, USA, April 2006.

[31] L. Buttyán and T. Holczer, "Private cluster head election in wireless sensor networks," in *Proceedings of the IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 1048–1053, Macau, China, October 2009.

[32] S. Shi, X. Liu, and X. Gu, "An energy-efficiency Optimized LEACH-C for wireless sensor networks," in *Proceedings of the 2012 7th International ICST Conference on Communications and Networking in China (CHINACOM '12)*, pp. 487–492, August 2012.

[33] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[34] O. Younis, S. Fahmy, and P. Santi, "Robust communications for sensor networks in hostile environments," in *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS '04)*, pp. 10–19, IEEE, Montreal, Canada, June 2004.

[35] K. Sun, P. Peng, P. Ning, and C. Wang, "Secure Distributed Cluster Formation in Wireless Sensor Networks," in *Proceedings of the 2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, pp. 131–140, Miami Beach, FL, USA, December 2006.

[36] D. Liu, "Resilient Cluster Formation for Sensor Networks," in *Proceedings of the 27th International Conference on Distributed*

Computing Systems (ICDCS '07), pp. 40-40, Toronto, ON, Canada, June 2007.

[37] H. Rifà-Pous and J. Herrera-Joancomartí, "A fair and secure cluster formation process for Ad hoc networks," *Wireless Personal Communications*, vol. 56, no. 3, pp. 625–636, 2011.

[38] G. Wang, D. Kim, and G. Cho, "A Secure Cluster Formation Scheme in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 301750, 14 pages, 2012.

[39] M. E. Elhdhili, L. Ben Azzouz, and F. Kamoun, "CASAN: clustering algorithm for security in ad hoc networks," *Computer Communications*, vol. 31, no. 13, pp. 2972–2980, 2008.

[40] M. Qin and R. Zimmermann, "An Energy-Efficient Voting-Based Clustering Algorithm for Sensor Networks," in *Proceedings of the 6th Inl Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS Int'l Workshop on Self-Assembling Wireless Networks (SNPD/SAWN)*, pp. 444–451, May 2005.

[41] T. Holczer and L. Buttyán, "Anonymous aggregator election and data aggregation in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 828414, 19 pages, 2011.

[42] G. Wang and G. Cho, "Reputation-based cluster head elections in wireless sensor networks," *Simulation*, vol. 89, no. 7, pp. 829–845, 2013.

[43] G. Wang and G. Cho, "Secure cluster head sensor elections using signal strength estimation and ordered transmissions," *Sensors*, vol. 9, no. 6, pp. 4709–4727, 2009.

[44] Q. Dong and D. Liu, "Resilient Cluster Leader Election for Wireless Sensor Networks," in *Proceedings of the 2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 1–9, Rome, Italy, June 2009.

[45] M. Sirivianos, D. Westhoff, F. Armknecht, and J. Girao, "Non-Manipulable Aggregator Node Election Protocols for Wireless Sensor Networks," in *Proceedings of the 2007 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pp. 1–10, Limassol, Cyprus, April 2007.

[46] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the the 9th ACM conference*, p. 41, Washington, DC, USA, November 2002.

[47] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 1, IEEE, Hong Kong, March 2004.

[48] K. Lu, Y. Qian, and J. Hu, "A framework for distributed key management schemes in heterogeneous wireless sensor networks," in *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC '06)*, p. 520, Phoenix, Ariz, USA, April 2006.

[49] P. Traynor, H. Choi, G. Cao, S. Zhu, and T. La Porta, "Establishing Pair-Wise Keys in Heterogeneous Sensor Networks," in *Proceedings of the Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pp. 1–12, Barcelona, Spain, April 2006.

[50] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," in *Proceedings of the the 10th ACM conference on Computer and Communications Security (CCS '03)*, pp. 62–72, Washington, DC, USA, October 2003.

[51] D. Liu, P. Ning, and W. Du, "Group-based key pre-distribution in wireless sensor networks," in *Proceedings of the the 4th ACM workshop on Wireless Security (WiSe '05)*, pp. 11–20, Cologne, Germany, September 2005.

[52] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 197–213, Washington, DC, USA, May 2003.

[53] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM conference on Computer and Communications Security (CCS '03)*, pp. 52–61, Washington, DC, USA, October 2003.

[54] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the the 10th ACM conference on Computer and Communications Security (CCS '03)*, pp. 42–51, Washington, DC, USA, October 2003.

[55] W. Gu, X. Bai, S. Chellappan, and D. Xuan, "Network Decoupling for Secure Communications in Wireless Sensor Networks," in *Proceedings of the 14th IEEE Int'l Workshop on Quality of Service (IWQoS 2006)*, pp. 189–198, New Haven, CT, USA, June 2006.

[56] G. Li, J. He, and Y. Fu, "A Group-Based Dynamic Key Management Scheme in Wireless Sensor Networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, pp. 127–132, Niagara Falls, Canada, May 2007.

[57] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2006.

[58] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122–130, 2006.

[59] G. Jolly, M. C. Kuscu, P. Kokate, and M. Younis, "A low-energy key management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC '03)*, pp. 335–340, Kemer-Antalya, Turkey, July 2003.

[60] G. Gupta and M. Younis, "Performance evaluation of load-balanced clustering of wireless sensor networks," in *Proceedings of the 10th International Conference on Telecommunication. ICT'2003. Conference Proceedings*, pp. 1577–1583, Papeete, Tahiti, French Polynesia.

[61] M. Chorzempa, J.-M. Park, and M. Eltoweissy, "Key management for long-lived sensor networks in hostile environments," *Computer Communications*, vol. 30, no. 9, pp. 1964–1979, 2007.

[62] C. Castelluccia and A. Spognardi, "RoK: A robust key pre-distribution protocol for multi-phase wireless sensor networks," in *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm*, pp. 351–360, Niece, France, September 2007.

[63] A. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP '03)*, pp. 326–335, Atlanta, Ga, USA, November 2003.

[64] H. Chan and A. Perrig, "PIKE: peer intermediaries for key establishment in sensor networks," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, pp. 524–535, Miami, Fla, USA, March 2005.

[65] R. Riaz, A. Ali, K. Kim, H. Ahmad, and H. Suguri, "Secure Dynamic Key Management for Sensor Networks," in *Proceedings of the 2006 Innovations in Information Technology*, pp. 1–5, Dubai, United Arab Emirates, November 2006.

[66] Z. Qingguang, C. Yanling, and L. Juan, "A Lightweight Key Management Protocol for Hierarchical Sensor Networks," in *Proceedings of the 2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*, pp. 379–382, Taipei, Taiwan, December 2006.

[67] T. Landstra, M. Zawodniok, and S. Jagannathan, "Energy-efficient hybrid key management protocol for wireless sensor networks," in *Proceedings of the 32nd IEEE Conference on Local Computer Networks, LCN 2007*, pp. 1009–1016, irl, October 2007.

[68] B. Panja, S. Madria, and B. Bhargava, "Energy and Communication Efficient Group Key Management Protocol for Hierarchical Sensor Networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06)*, pp. 384–393, Taichung, Taiwan, 2006.

[69] M. Eltoweissy, M. H. Heydari, L. Morales, and I. H. Sudborough, "Combinatorial optimization of group key management," *Journal of Network and Systems Management*, vol. 12, no. 1, pp. 33–50, 2004.

[70] M. Eltoweissy, A. Wadaa, S. Olariu, and L. Wilson, "Group key management scheme for large-scale sensor networks," *Ad Hoc Networks*, vol. 3, no. 5, pp. 668–688, 2005.

[71] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[72] P. Schaffer, K. Farkas, Á. Horváth, T. Holczer, and L. Buttyán, "Secure and reliable clustering in wireless sensor networks: a critical survey," *Computer Networks*, vol. 56, no. 11, pp. 2726–2741, 2012.

[73] C. Tseng, C. Chau, K. Elbassioni, and M. Khonji, "Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones," https://arxiv.org/abs/1703.10049.