Received -; | Revised -; | Accepted 31 May 2018

DOI: xxx/xxxx

SPECIAL ISSUE PAPER

A Hierarchical Key Pre-Distribution Scheme for Fog Networks

Pooneh Nikkhah Bahrami[1] | Hamid H.S.Javadi[2] | Tooska Dargahi[3] | Ali Dehghantanha[4] | Kim-Kwang Raymond Choo[5]

[1]Department of Computer Science, Tehran University, Kish Campus, Iran
[2]Department of Computer Science, Shahed University, Tehran, Iran
[3]School of Computing, Science and Engineering, University of Salford, Manchester, UK
[4]Department of Computer Science, University of Sheffield, Sheffield, UK
[5]Department of Information Systems and Cyber Security, University of Texas at San Antonio, Texas, US

Summary

Security in fog computing is multi-faceted, and one particular challenge is establishing a secure communication channel between fog nodes and end devices. This emphasizes the importance of designing efficient and secret key distribution scheme to facilitate fog nodes and end devices to establish secure communication channels. Existing secure key distribution schemes designed for hierarchical networks may be deployable in fog computing, but they incur high computational and communication overheads and thus consume significant memory. In this paper, we propose a novel hierarchical key pre-distribution scheme based on "Residual Design" for fog networks. The proposed key distribution scheme is designed to minimize storage overhead and memory consumption, while increasing network scalability. The scheme is also designed to be secure against node capture attacks. We demonstrate that in an equal-size network, our scheme achieves around 84% improvement in terms of node storage overhead, and around 96% improvement in terms of network scalability. Our research paves the way for building an efficient key management framework for secure communication within the hierarchical network of fog nodes and end devices.

KEYWORDS:
Fog Computing, Key distribution, Hierarchical Networks.

1 | INTRODUCTION

Fog computing can be broadly defined as extending the cloud to the edge of the network to enable processing data as close as possible to the data origin (i.e., end devices) [1,2]. Such an architecture reduces data processing delay and resource consumption, which are critical in time-sensitive applications, in the era of Internet-of-Things [3]. As defined by NIST [4], fog devices (also known as fog nodes) can be any device with computing and storage capabilities, deployed in a variety of environments within the end devices access network, placed between millions of end device and the cloud. Some examples of fog nodes are cellular base stations, access points, roadside units, embedded servers, and even smartphones [5,6,7].

Fog computing is increasingly prevalent, such as in oil and gas, and smart transport sectors [8]. This is partly perhaps due to the huge amount of data generated every day via smart nodes. For example, an offshore oil rig generates 2TB of data every month, and processing these data closer to its origin would provide significant performance benefits [8]. Moreover, there are several time-sensitive applications that require real-time data processing and decision making. Decisions to be made by smart nodes are unlikely to be complex, and hence many data analysis tasks in the smart networks can be performed in the fog layer where more powerful and resourceful nodes reside. This will reduce latency and communication overhead. For example, consider a scenario in which oil pipeline sensors capture an unusual pressure change. In such a scenario, processing of the generated data in the fog layer and making an appropriate decision would be more efficient and time-saving, compared to when the data is sent to the cloud for processing. One of the basic and most wildly used architectures in the fog computing is hierarchical three-layer architecture [9,10]. Figure 1 depicts an example fog computing architecture, where end devices, fog nodes, and cloud services form a three-layer hierarchical network structure [4,11,3].

The up-most layer is termed as Cloud layer which consists of cloud and data storage servers. We assume this layer to be a resourceful network component with high communicative and processing capabilities, managed by the Cloud Service Providers (CSP). The middle layer termed as fog layer, which consists of network devices, such as routers, switches, and access points that serve as a gateway to the Internet. Finally the lowest layer also known as end devices layer consists of resource-constrained sensor nodes and IoT devices.

In such a network model, end nodes, fog nodes, and cloud layer have different capabilities, in terms of processing power, communication range, and power consumption. One particular challenge in fog computing is secure communication between these three layers; specifically

1. Communication between fog nodes and Cloud service center;
2. Communication among fog nodes;
3. Communication among constraint-IoT enabled end devices;
4. Communication between fog nodes and end devices.

Therefore, there is a need for efficient authentication, encryption, and key management protocols to ensure secure communication in fog networks [12].

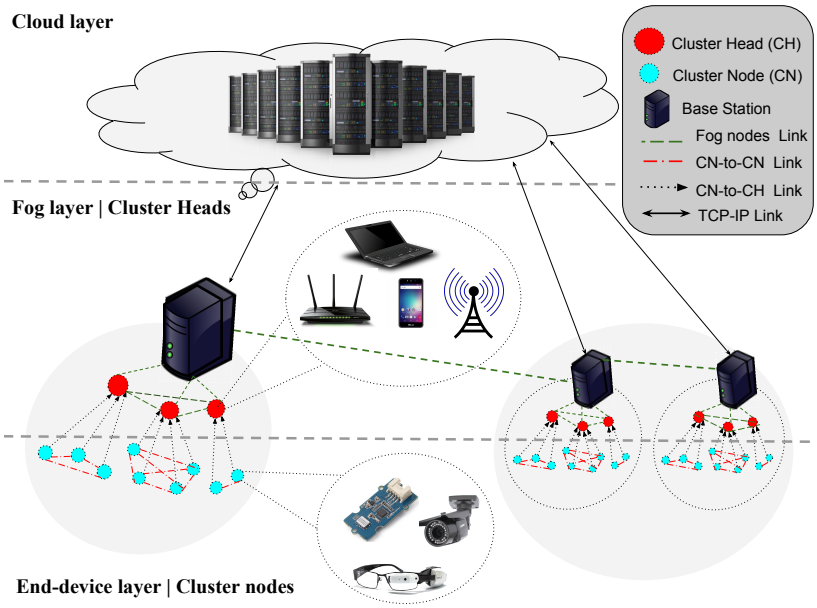

FIGURE 1 An example fog computing architecture.

A Fog network comprises devices with relatively limited computing resources, hence, it is typically not realistic to execute conventional security solutions on Fog network [12]. The challenge of establishing secure communication between end devices, and between end devices and fog nodes, is similar to that of establishing secure communication problem in a Hierarchical Wireless Sensor Networks (WSNs). The latter is a relatively mature and well understood area, where a number of symmetric key encryption schemes have been proposed in the literature [13,14,15].

In addition to a secure key establishment, there is a need to ensure that other requirements (e.g., network connectivity and scalability) can be achieved. There is a fairly rich literature in the context of key management/distribution schemes in WSNs [16,17,18,19,20,21,22,23,24,25,26]. However, most proposed schemes impose high computational, communication, and memory overhead. A number of key distribution schemes for flat networks based on combinatorial design, a scalable method that distributes key between all nodes deterministically in order to increase the network connectivity and reduce memory consumption, have also been proposed in the literature [16,18,21,25,26]. Such schemes are, however, not directly applicable to hierarchical fog networks. Therefore, there is a need for an effective “hierarchical key management and key distribution” schemes designed for fog networks, given its increasing popularity. In this paper, we present a hierarchical key pre-distribution scheme based on "residual design", which is a specific type of combinatorial design, for fog networks. The proposed scheme allows distribution of keys to the network devices/nodes prior to network deployment. The residual design would significantly reduce memory consumption on end devices. Moreover, the residual design of the key pre-distribution improves the network scalability, allowing to support large networks with almost similar amount of memory requirement. In our proposed method, we focus on the communications within the lower two layers of the fog computing architecture (refer to Figure 1), i.e., end device layer and fog layer. In particular, we consider the communication between end devices, and between fog nodes and end devices, due to the

resource limitation of the end devices. We consider these two layers, i.e., fog and end device layers, as a hierarchical network in which the end devices play the Cluster Node (CN) role and fog nodes play the Cluster Head (CH) role as we explain in Section 3.

In Section 2, we start by reviewing relevant literature, followed by a brief review of background materials on combinatorial designs and specifically residual design. In Section 3, we describe our system and attack model. Section 4 presents our proposed hierarchical key distribution scheme. In Section 5, we then conduct both analytical and experimental evaluations, as well as a comparative study with other state-of-the-art hierarchical key distribution schemes, considering different evaluation metrics, i.e., storage overhead, network scalability, connectivity and resilience against node capture attacks. We show that our proposed scheme improves network scalability and reduce storage overhead compared to the previous work, while providing a reasonable key share probability between end nodes. Finally, Section 6 concludes the paper and future research directions are also described.

2 | RELATED WORK AND BACKGROUND

In fog Networks, security and privacy is a challenging issue in all three layers (refer to Figure 1) including cloud layer, fog layer and end devices [9]. There are numerous surveys and research papers about security in the context of cloud computing. For example, in [27] researchers proposed a traditional PKI based authentication mechanism. Yu et al. [28] propose a data access control scheme based on attribute-based encryption (ABE). ABE is a one-to-many public key encryption mechanism that employs users identity as an attribute. In ABE, a set of attributes and a private key computed from the attributes are respectively used for encryption and decryption. Moreover, authors in [29,30,27] studied password-based authentication for secure communications in cloud computing network. However, due to special requirements of fog networks, i.e., mobility, heterogeneity, large-scale geo-distribution, and existence of resource constrained devices, utilization of security and privacy mechanisms that are proposed for cloud are not completely applicable to fog networks [31]. Moreover, utilizing one common password for all the devices is not suitable for securing communication between variety of devices in fog networks.

In order to address security and privacy challenges in fog computing, while considering the afore-mentioned features of fog networks, several researchers focused on establishing a reliable communication between cloud and fog nodes. Alrawais et al [32] developed an encrypted key exchange protocol based on Ciphertext-Policy Attribute Based Encryption (CP-ABE) to enable authentication and confidential communications between fog nodes and cloud. In [33] a policy-based resource access control in fog networks based on public key based solutions is proposed to support secure collaboration and interoperability between heterogeneous nodes. While the above-mentioned algorithms could be considered for securing the communication between the fog nodes and cloud, they are usually resource-consuming and are not suitable for utilization in the end device layer of the fog architecture.

The lowest layer of the fog architecture, i.e., end devices, is composed of variety of devices including sensors. Security and privacy issues in the context of wireless sensor networks is well-studied in the literature. As our concentration in this work is on the lowest layer of fog architecture, we review the most related work to our proposal in Section 2.1. Due to the resource constraint inherent of end devices in the lowest layer of the architecture, many researchers concentrated on proposing secure, lightweight and energy efficient approaches, which we discuss in the next section. The main challenges of the existing solutions are scalability and memory overhead, which was our motivation for the current work.

2.1 | Existing Schemes for End devices layer

Symmetric key schemes

In symmetric key schemes, the key sharing procedure could be performed either before the network deployment (so-called key pre-distribution) or after the network deployment. Generally, a key distribution center (KDC) performs the key generation and distribution process. Due to the constrained energy budget and limited computational and communication capacities of end devices, key pre-distribution schemes (KPSs) are the most desirable options. Hence, our concentration here is on the KPSs, in which keys are loaded into end devices' memory before their distribution in the network. In such schemes, usually every pair of nodes are able to securely communicate with each other due to their shared common credential(s) [13,14,15]. The pre-shared credentials might be produced randomly or deterministically.

Random key pre-distribution:

In these approaches, credentials (keys which are bytes) are chosen randomly from a key pool and distributed among nodes. Eschenauer and Gligor [17] were the first to propose a random key pre-distribution scheme specifically for WSNs. This scheme is simple and has low memory usage while ensuring network connectivity. However, it fails to provide pair-wise authentication and requires significant communication overhead [13]. Several solutions based on [17] were subsequently proposed [34,35,36,37]. These latter schemes modified the random key pre-distribution scheme by increasing

connectivity between nodes or decreasing storage overhead of sensor nodes. However, they are proposed for flat networks and are not effective for hierarchical heterogeneous fog scenarios.

Deterministic key distribution:

In deterministic key distribution schemes, the key pool is generated using a deterministic process. In several existing deterministic models, mathematical designs have been applied in key generation and distribution. The Combinatorial design is one known mathematical scheme used in this category [25]. In [16], the authors proposed two key pre-distribution schemes based on Symmetric Balanced Incomplete Block Design (SBIBD) and Generalized Quadrangles (GQ) [26]. However, both schemes are not scalable and are vulnerable to node capture attacks. In [25], the authors adopted the concept of set system in deterministic KPS and applied Transversal Design (TD), and proved the scheme to be more secure than that in [17] against node capture attacks [38]. Trade-KP is another scheme based on paired balanced incomplete block design [18]. Similarly, orthogonal arrays are used in [19]. As most of the explained key management schemes [17,18,19,25,26,38] are proposed for homogeneous or flat networks, they cannot be deployed for hierarchical heterogeneous fog scenarios. Recently, several KPSs for heterogeneous networks have been proposed in the literature. For example, the scheme in [21] is based on TD and has been shown to be more resilient against node compromise attack in comparison with TD [25] and BIBD schemes.

Asymmetric key schemes

Asymmetric or public key cryptography schemes are conventionally used to establish secure communication between entities. Such schemes are not suitable for deployment on low power battery operated devices, due to the need to compute expensive cryptographic operation(s), although there have also been attempts to design asymmetric key schemes for resource-constrained environments, such as IoT. For example, the proposed approaches in [39] and [40] seek to minimize the number of exchanged messages. Rabin's scheme [39] is similar to the RSA algorithm and has high energy consumption for decryption operations, although it is faster than RSA. NtruEncrypt [40] is more efficient and suitable for some resource-limited devices as the scheme consumes less energy. However, it generates large-size messages leading to communication overhead due to the requirement for re-transmission in noisy environments. Liu et. al. [41]introduced two secure traffic light control schemes in Vehicular Ad hoc NETwork (VANET) using fog computing based on the hardness of the computational Deffie-Helman puzzle. Although they propose an improved scheme, in which a traffic light in fog layer needs to perform lightweight operations, but it does not consider communication and computation overhead imposed to the lowest layer devices. Other researchers [42,43,44,45,46] argue that asymmetric solutions are actually suitable for resource-constrained devices due to their flexibility and scalability in terms of shared key management. However, most of these schemes have more or less similar challenges mentioned earlier, and are not fully applicable for our fog scenario.

Also, most asymmetric-based schemes in the literature focus on flat networks with homogeneous sensors. Recently, there have been published schemes for hierarchical networks based on asymmetric key distribution. For example, in [47] and [48], two key generation and distribution schemes based on elliptic curve cryptography (ECC) for hierarchical WSNs are presented.

Discussion

Having explained the existig methods for providing secure communication in fog networks, we can conclude that schemes that use asymetric encryption usually mandate expensive computation and communication cost on nodes. However, these approaches are more scalable and resilient against node capture attacks with less memory requirements. On the other hand, symmetric key encryption has less computational complexity,which is an important feature dues to resource-constraint nature of end devices. However, existing symmetric key pre-distribution schemes suffer from low connectivity, high communication complexity, high memory overhead, and limited scalability, and are vulnerable against node capture attacks [49]. These challenges motivated us to devise a new symmetric key pre-distribution scheme to address the scalability and memory challenges, while taking into account resilience against node capture attack. Our proposal provides a reasonable key sharing probability and significantly reduced the memory requirements.

2.2 | Combinatorial Design

Since our proposed method is based on a mathematical structure, named combinatorial design, in this section we provide the required background which is necessary to follow the rest of the paper.

Definition 1. A set design is a pair (X, A) where X is a set of v elements (points) and A is a finite set of subsets of X called blocks. The degree of a point $x \in X$ is the number of blocks containing x. The rank of a set system is the size of the largest block and (X, A) is said to be uniform of rank k if all blocks have the same size k [25].

Definition 2. Balanced Incomplete Block Design (BIBD) is a set design. It is an arrangement of v distinct objects into b blocks, where each block contains exactly k distinct objects, each object occurs in exactly r different blocks, and every two distinct objects occur together in exactly blocks"[50]. It is denoted either by (v, k, λ) or (v, b, r, k, λ) where $\lambda \times (v-1) = r \times (k-1)$ and $b \times k = v \times r$.

Definition 3. Symmetric BIBD (SBIBD) is a BIBD for which $b = v$ and, consequently, $r = k$. It is denoted by $(v, k,) - \text{SBIBD}$. For every prime $q \geq 2$ there exists a symmetric $(q^2 + q + 1, q + 1, 1)$-BIBD known as projective plane[51].

Definition 4. For every prime power $q \geq 2$, there exists a $(q^2 + q + 1, q + 1, 1)$-SBIBD (i.e., a projective plane of order q)[51].

Definition 5. A Latin square on q symbols is a $q \times q$ array such that each of the q symbols occurs exactly once in each row and in each column. The number q is called order of square. If $A = (a_{ij})$ and $B = (b_{ij})$ are any two $q \times q$ arrays, the join of A and B is a $q \times q$ array whose $(i, j) - \text{th}$ element is the pair (a_{ij}, b_{ij}). Latin squares A and B of order q are orthogonal if all entries of A join B are distinct. Latin square $A_1, A_2, ..., A_r$ are Mutually Orthogonal Latin Squares (MOLS) if they are orthogonal in pairs. For prime power q, a set of $(q-1)$ MOLS of order q can be used to construct affine plane of order q, and can be converted to a projective plane of order q[51].

Definition6. According to Definition 2, any two blocks of an SBIBD contain λ common points. The relation between affine and projective plains can be generalized to other block designs. This result provides another method of constructing new BIBDs called Residual Design.

Residual Design Theorem: Suppose (X, A) be a symmetric (v, k, λ)-BIBD where $A = \{A_1, A_2, ..., A_v\}$ and $X = \{x_1, , x_v\}$. Let for every $1 \leq i \leq v$, $A_i \in A$. Then, $\{A_1 \setminus A_i, A_2 \setminus A_i, ..., A_{i-1} \setminus A_i, A_{i+1} \setminus A_i, ..., A_v \setminus A_i\}$ are blocks of a $(v-k, v-1, k, k-\lambda, \lambda)$-BIBD from set point $X \setminus A_i$. Thus, $\text{Res}(X, A, A_0) = \{X \setminus A_0, \{A \setminus A_0 : A \neq A_0\}\}$ is residual design of BIBD based on A_0[25]. Residual design is constructed by deleting all points in A_0 and then deleting A_0. Clearly, a residual design is a BIBD, having block size at least two and at most the number of points minus one.

Example1: Suppose (7,3,1)-BIBD with the following point set and blocks: $V = \{1, 2, 3, 4, 5, 6, 7\}$ $B1 = \{1, 2, 3\}$, $B2 = \{1, 4, 5\}$, $B3 = \{1, 6, 7\}$, $B4 = \{2, 4, 6\}$, $B5 = \{2, 5, 7\}$, $B6 = \{3, 4, 7\}$, $B7 = \{3, 5, 6\}$. Then, we have seven classes to build the residual sets, each forming a (4,6,1)-BIBD, which we explain in the following.

1. $C1 = X \setminus B1 = \{4, 5, 6, 7\}, B2 \setminus B1 = \{4, 5\}, B3 \setminus B1 = \{6, 7\}, B4 \setminus B1 = \{4, 6\}, B5 \setminus B1 = \{5, 7\}, B6 \setminus B1 = \{4, 7\}, B7 \setminus B1 = \{5, 6\}.$
2. $C2 = X \setminus B2 = \{2, 3, 6, 7\}, B1 \setminus B2 = \{2, 3\}, B3 \setminus B2 = \{6, 7\}, B4 \setminus B2 = \{2, 6\}, B5 \setminus B2 = \{2, 7\}, B6 \setminus B2 = \{3, 7\}, B7 \setminus B2 = \{3, 6\}.$
3. $C3 = X \setminus B3 = \{2, 3, 4, 5\}, B1 \setminus B3 = \{2, 3\}, B2 \setminus B3 = \{4, 5\}, B4 \setminus B3 = \{2, 4\}, B5 \setminus B3 = \{2, 5\}, B6 \setminus B3 = \{3, 4\}, B7 \setminus B3 = \{3, 5\}.$
4. $C4 = X \setminus B4 = \{1, 3, 5, 7\}, B1 \setminus B4 = \{1, 3\}, B2 \setminus B4 = \{1, 5\}, B3 \setminus B4 = \{1, 7\}, B5 \setminus B4 = \{5, 7\}, B6 \setminus B4 = \{3, 7\}, B7 \setminus B4 = \{3, 5\}.$
5. $C5 = X \setminus B5 = \{1, 3, 4, 6\}, B1 \setminus B5 = \{1, 3\}, B2 \setminus B5 = \{1, 4\}, B3 \setminus B5 = \{1, 6\}, B4 \setminus B5 = \{4, 6\}, B6 \setminus B5 = \{3, 4\}, B7 \setminus B5 = \{3, 6\}.$
6. $C6 = X \setminus B6 = \{1, 2, 5, 6\}, B1 \setminus B6 = \{1, 2\}, B2 \setminus B6 = \{1, 5\}, B3 \setminus B6 = \{1, 6\}, B4 \setminus B6 = \{2, 6\}, B5 \setminus B6 = \{2, 5\}, B7 \setminus B6 = \{5, 6\}.$
7. $C7 = X \setminus B7 = \{1, 2, 4, 7\}, B1 \setminus B7 = \{1, 2\}, B2 \setminus B7 = \{1, 4\}, B3 \setminus B7 = \{1, 7\}, B4 \setminus B7 = \{2, 4\}, B5 \setminus B7 = \{2, 7\}, B6 \setminus B7 = \{4, 7\}.$

Most of the key distribution schemes which use combinatorial design to produce their key rings provide full or high connectivity between pair of nodes in the network. However, an important point to consider is that, higher connectivity leads to lower resilience[25].

On the other hand, the main problem of existing KPSs that use combinatorial methods[17,52,18] is their limited scalability. Therefore, we show that the basic mapping from residual design to KPSs, that is proposed in[53], and its usage for hierarchical network architecture provides higher scalability, good connectivity and higher resilience against node capture attack.

3 | SYSTEM MODEL

In our network mode, we consider a hierarchical fog network as shown in Figure (2). In the lowest layer, we consider end devices as cluster nodes (CNs), which are resource-constrained devices capable of communicating with each other directly or via multi-hop path, as well as with the fog nodes in the higher layer. In the physical fog layer, we consider two virtual layers (similar to[54]): 1) several fog nodes, such as smart phones, access points, or workstations, serving as cluster heads (CHs) at the upper sub-layer of the fog layer, which have more capabilities and resources compared to the end devices. These fog nodes are directly connected to end devices and are able to receive and process data for CNs. Fog nodes in sub-layer-1 are connected to a higher sub-layer, which we consider as Base Stations (BSs), through the internet backbone. We consider the BS to be a resourceful component having high communicative and processing capabilities and memory capacity, which acts as a gateway to the Internet, as well as providing human interface. Base stations are connected to the top layer, cloud servers, via TCP/IP connection. If the workloads received by sub-layer-1 fog nodes (CHs) exceed their computational capacity, the excessive amount of data will be forwarded to the higher sub-layer or BS. If needed BS will forward workload to the cloud through internet backbone. We can consider the end devices that are in a specific area, e.g., in an apartment, as a cluster. However, in case there are a large number of independent nodes in a cluster, we might have several clusters inside an area.

All of nodes in these clusters carry sensitive data, and hence the exchanged data should be encrypted and delivered only to the authorized entities. Therefore, the nodes inside one cluster require to have shared keys to communicate with each other. However, in order to communicate with other clusters, the nodes inside a cluster need to send their messages through their corresponding CH in the fog layer. The BS serves as an edge router that connects given cluster to the Internet, to provide connectivity with other authorized devices or users. Table 1 presents the notations that are used in the remaining of the paper.

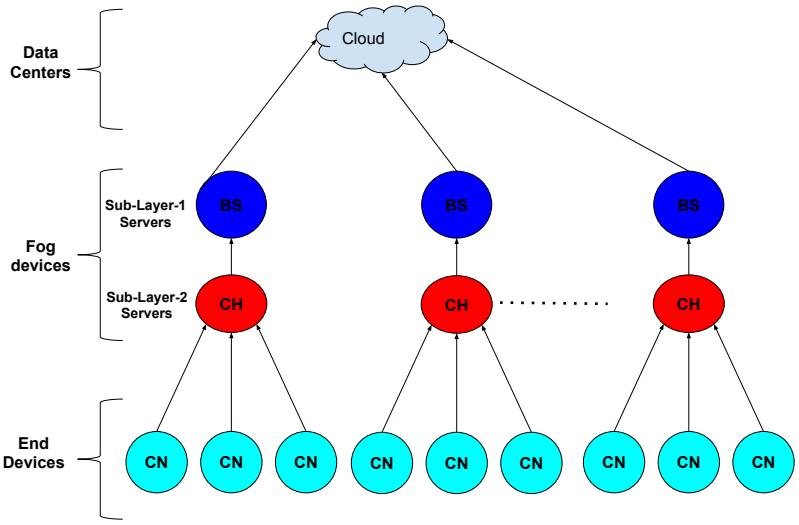

FIGURE 2 A schematic example of our considered system model. In this model, the lowest layer, is named "End Devices" layer that composed of cluster nodes (depicted as CN) which are basically sensor nodes, IoT devices and so on. The middle layer, is named "Fog" layer which consists of tow sub-layer: 1. Cluster Heads (depicted as CH) which are basically access points, routers and so on; 2. Base Station (depicted as BS) as the upper sub-layer. The topmost layer, is named "Cloud" layer.

TABLE 1 List of Used Notations.

Notation	Definition
N	The network size
N_{RD}	Number of supported nodes in Residual Design KPS
BS	Base Station
C	Total number of clusters
n_j	Number of nodes in cluster j
CN_i^j	i-th cluster node of cluster head j
CN	Cluster Node
CH	Cluster Head
$CH_i(C_i)$	i-th class of residual design
$K_{BS} - CH_j$	Pair-wise key between base station and cluster head
q	A prime number which satisfies certain conditions.

In our scheme, we consider to have N_{RD} nodes, composed of $(q^2 + q + 1)$ fog nodes (CHs), where q is a prime number, $q \geq 2$; as well as $(q^2 + q + 1) \times (q^2 + q)$ low end devices (CNs), where $q \geq 3$. Our proposed network model requires three phases of deployment: (i) Pre-deployment phase: using a safe BS or a KDC we construct a key pool from which $q^2 + q + 1$ blocks are generated using a SBIBD scheme. Then, the obtained residual design blocks are used to generate $(q^2 + q + 1)^2$ blocks from which $(q^2 + q + 1)$ blocks are assigned to CHs and $(q^2 + q + 1) \times (q^2 + q)$ blocks are assigned to CNs; (ii) Shared-key discovery and path key establishment phase: node deployment and clustering is performed in this phase. Nodes

TABLE 2 Mapping From Residual Design to Key Distribution.

Residual Design	Key pre-distribution
Point set($\|S\|$)	Key pool (KP)
Object Set size ($\|S\| = q^2 + q + 1$)	key pool size
Blocks	Key ring
Number of Blocks ($(q^2 + q + 1)(q^2 + q)$)	Number of key rings
Size of a block ($k = q$)	Size of a key ring

which are in each others' transmission range exchange a list of key identifiers to find a shared key; neighbouring nodes with no shared key rely on a mediator node to discover a path key. (iii) Post deployment authentication: after network deployment and cluster configuration, users who want to reach data inside a cluster need to authenticate to the BS and CHs.

3.1 | Attack Model

In this work, our main concentration is on secure intra-cluster and inter-cluster communication. Since not all services provided by a cluster are public, an authentication and authorization mechanism must be developed to protect against unauthorized user. We assume that the BS is secure and is in charge of performing the authentication and expensive tasks. The authentication process relies on the BS as a gateway between edge devices [55], [56]. Before the network deployment, BS acts as a KDC and generates keys (based on our proposed scheme that we explain in Section 4) to be assigned to the nodes in its corresponding cluster. In other words, each end device is pre-loaded with required secret keys for message encryption. The key pre-distribution algorithm in its pre-deployment phase is assumed to be secure since keys are loaded offline in the nodes prior to their deployment in the network. Thus, an attacker does not have access to key pool and key chains of nodes. Moreover, consistent with the literature [57], the bootstrapping time for a node, is shorter than the time required for an attacker to comprise a node. Hence, a node can not be compromised during this time. Communication is assumed to be secure in the shared-key discovery phase [17] since the nodes only exchange key identifiers in this phase and an attacker who does not know the mapping between identifiers and the keys cannot recognize the shared key(s) between two nodes by eavesdropping, without a physical access to the node. An attacker node can replace itself as a real node only when it captures a node physically and seizes its keys [58].

We also assume that every Internet user before using the data services provided by the end devices in clusters, should be registered to BS and demanded CH(s) to obtain its access authorization. This phase includes the verification of the user's password, e.g., by smart card [59]. It should be mentioned that authentication process for Internet users is independent of the authentication process of domestic nodes.

The nature of restricted end devices makes them more vulnerable against security attacks, such as node capture attack. Physical capture is a common threat that endangers inter-node links through manipulating security keys. It is considered by many as the pioneer attack that paves the way for other attack types [50]. Thus, the present study considers and enhances the network resilience against node capture attacks.

4 | THE PROPOSED APPROACH

In this section we discuss our proposed user authentication and key agreement scheme for edge network. Our main concentration is on key agreement between edge devices and also communication between devices and the fog nodes. The basic idea of the proposed key management scheme is utilisation of residual theorem by the KDC (the BS) to build key-chain in off-line pre-deployment phase. Clustering is based on a node that was used in subtraction phase of the residual design. Prior to their deployment in a target field (deployment field), nodes (end devices and fog nodes) are loaded with key chains built by the KDC (or BS) based on $res(X, A, A_i)$. In what follows we explain our proposed key generation phase based on residual theorem, and its mapping to fog network (see Table 2).

Pre-deployment phase: A residual design is used to generate initial keys. Finite projective planes of order q (a type of SBIBD with $(q^2+q+1, q^2+1, 1)$ parameter) are used for constructing residual designs, provided that $(q^2 + q + 1)^2 \geq N_{RD}$. Let us assume that i-th class of residual design is generated by using point set $X \setminus A_i$ denoted by C_i, and j-th node from class C_i with $A_j \setminus A_i$ is denoted by A_j^i $(j = 1, ..., v; i = 1, ..., i-1, i+1, ..., v)$. Since in this design, key rings are loaded before node deployment, classes should be predetermined. Consider two following properties:

1. The point sets of each class of the proposed scheme formed a BIBD with $(v, b, r, k, \lambda) = (q^2, q^2 + q, q + 1, q, 1)$ parameters. Proof: This property results from residual theorem in Section 2.2.

2. Suppose that key ring and key space sizes in symmetric key distribution design are $k = q + 1$ and $v = q^2 + q + 1$, respectively. Then, residual design supports a network with maximum size of $N_{RD} = (q^2 + q + 1)^2$. Proof: Since every class of residual design forms a $(q^2, q^2+q, q+1, q, 1)$-BIBD, and the number of classes equals to q^2+q+1, the proposed scheme supports up to $(q^2 + q + 1) \times (q^2 + q)$ nodes. Each block of the class C_i can be assigned to CH nodes.

Therefore, a total number of $(q^2 + q + 1) \times (q^2 + q) + (q^2 + q + 1) = (q^2 + q + 1)^2$ nodes are covered. The nodes (end devices) are deployed randomly in a cluster, and each CH is deployed in that cluster around the center of that cluster. The BS is located in the fog layer of the network.

Shared key discovery: After deployment of cluster nodes in their relative clusters, every node of each cluster has to identify other CNs within its communication range with which it has a shared key. To do so each two nodes exchange their own key identifiers on their key ring through a network broadcasting [17]. The motivation of this phase is to establish a routing graph of the end devices. Existence of a link between two CNs has three meanings: i) they belong to the same cluster; ii) they are placed in their communication range, and finally iii) they have a common key to communicate safe on that link by encryption.

After node deployment, CHs generate a random unique number and then transmit the encrypted number to their cluster members via a shared key. The random number is added to the end of the keys of the cluster head as well as the related cluster node. It is worth mentioning that the random number does not have any role in communications between CNs and CHs directly; it is just used to separate the key space of each cluster head into its unique spaces. Hence, revealing random number alone does not cause any sensible risk to secure communications of the network. A communication will be in risk when both random number and key chain K_{ij} are revealed by the adversary due to a node capture attack.

In our scheme, we use BIBD for key pre-distribution in CHs. Every pair of CHs share a key at some point because BIBD is a fully-connected KPS where a pair of nodes can directly connect to each other. Because number of CHs is limited, BIBD is the best choice for establishing connections between CHs. As explained, clusters are assigned different key spaces. This will reduce node capture attack effects. On the other hand, residual design enhances network scalability since a large number of blocks can be generated with a parameter q.

Registration, login and authentication phase: After deployment of end devices and fog nodes and completion of shared key discovery process and path key establishment, the remote user authentication phase starts. Since our main goal in this paper is to establish a secure communication between CNs,between CNs and its related CH in a cluster and also between CHs, we assume that when the remote user U_i wants to access data from the intended cluster, he/she can perform it using some existing secure techniques [60]. Based on proposed schemes the user U_i needs to be authenticated at both the BS and the demanded CHs within the cluster(s) that he/she wants to access data. After successful authentication, the user and authenticated CH(s) will be able to establish a secure communication link using a shared secret session key. In fact, this session key is the means of approaching user to data inside cluster via its CH.

5 | ANALYSIS OF THE PROPOSED SCHEME

In this section, the proposed scheme is analysed considering four important metrics: memory overhead, scalability, network connectivity and its resilience against node capture attacks.

5.1 | Simulation Setting

In our simulation experiment, we consider a network of nodes that are distributed uniformly in a 1×1 square. Sensor coordinates are denoted by (x,y) where x and y are random uniform digits between 0 and 1. Radio range of a node is a circle of radius r, where $0 < r < 1$. A graph can be obtained by connecting each node to the neighboring sensor in the same radio range. The node degree can substitute for location and communicating range r. The relationship between average node degree d and communication range r, as suggested in [61], is calculated as

$$d = (n-1)(\pi r^2 - 8/3r^3 + 1/2r^4) \quad (1)$$

To evaluate our scheme security, we need to determine the number of sensor nodes and nodes mean degree to create a random graph with similar properties.

The average degree of the random graph vertices is calculated by (2) to build a graph with probability of c [17]:

$$d = ((n-1)/n) \times (\ln(n) - \ln(n)(-\ln(c))) \quad (2)$$

Where c=0.9 or 0.99 or 0.999. This means the network will âĂIJalmost certainlyâĂİ be connected. In the present study, to simulate combinatorial designs in fog networks, key chains are built using residual design and then attributed to nodes. Moreover, a connected random graph with probability of 0.9 is used to establish security and key sharing between nodes and simulation is performed using C♯.NET

5.2 | Performance Evaluation

Considering that mostly the nodes in the end device layer (and possibly some nodes in fog layer) are resource constraint, we have similar limitations as HWSN in our fog scenario. As, to the best of our knowledge, there is not a hierarchical key distribution method dedicated to fog networks in the literature, we compare our proposed scheme with related work in HWSN. We compare our proposal with the state-of-the-art solutions presented in [21], [47] and [48]. In [21] a symmetric KPS for hierarchical WSNs based on TD is proposed, while, [47] adopts public key cryptography to establish secure communications in hierarchical WSNs. Finally, in [48] authors presented three key management schemes, namely SACK, SACK-P, and SACK-H, which are using symmetric, asymmetric and hybrid cryptographic algorithms, respectively. It is worth mentioning that, in [47] authors compared their model with [23] and [62] in term of resiliency against node capture attacks, memory overhead and scalability, and proved that their proposal outperforms the previous solutions; hence, we avoid providing a comparative analysis between our proposed method and the schemes in [23] and [62]. We consider the following definition for each of the considered metrics: Connectivity: probability of any given pair of neighbouring nodes sharing at least a common key. If two nodes are neighbours, it is called local connectivity; otherwise, it stands for global connectivity. Network Scalability: the ability to support different network sizes and add new nodes in the network, while having the same key pool and same key-chain length [63]. In this section, in order to evaluate the scalability of different schemes, we compute the number of keys required by each scheme to support the same network size. The smaller the required number of keys, the more scalable is the scheme. Memory overhead: the amount of memory required to store key-chain. The key-chain size is related to the number of keys in the key-chain of a node [26].

5.2.1 | Memory overhead

Number of keys in each node:

As we discussed earlier every CN is preloaded with a distinct q-key chain which is a block of a residual design. As shown in Table 2 , the number of CHs and CNs is C and N, respectively. Number of cluster nodes in each cluster is denoted by n_j. According to mapping in Table 2 , by using $(q^2+q+1, q+1, 1)$ -SBIBD and residual design we can generate q^2+q+1 clusters each of them is a $(q^2, q^2+q, q+1, q, 1)$-BIBD so $N = (q^2+q+1)(q^2+q)$, $C = q^2+q+1$. In key pre-distribution scheme based on residual design, each CH is preloaded with a key chain of length (q^2+q) to communicate with its own CNs and is also loaded with a key chain of BIBD design with length of $\sqrt{(q^2+q+1)}$ to establish secure communications with other CHs. Key pre-distribution in hierarchical networks is privileged in comparison to public key cryptography protocols [64]. To show this advantage, the proposed scheme is compared in terms of number of required keis in each node and memory usage with [47] and [48] that use elliptic curve cryptography (ECC) for generation and distribution. In [47], public key cryptography algorithm is used for establishing a secure connection in hierarchical networks, and in [48] a uniform security framework is suggested, including three key management schemes of SACKC, SACK-P, and SACK-H, from which only SACK-P uses public key cryptography. Therefore, only SACK-P is compared with our proposed scheme. In [21] a hierarchical key pre-distribution scheme that use combinatorial design based on TD is used. we show that in compare with aforementioned methods the number of required key in each node of our scheme is considerably less which reduces memory overhead of CNs. The number of required keys in each scheme having the same network size is given in Table 3 . As it is comparable, the proposed scheme requires fewer keys specifically in CNs at equal network size. Numerical results of required keys in each node to achieve 90.000 nodes (include CHs and CNs) in a hierarchical wireless sensor is given in figure (3).

TABLE 3 Number Of Required Keys In Each Node

Scheme	Number of keys in each CH	Number of keys in each CN
Proposed Scheme	$n_j + (\sqrt{(q^2+q+1)}) = (q^2+q) + (\sqrt{(q^2+q+1)})$	q
Scheme in [21]	$\sqrt{N} + \sqrt{C} = \sqrt{(q^2+q+1)(q^2+1)} + \sqrt{(q^2+q+1)}$	$C = q^2+q+1$
Scheme in [47]	$N+2 = (q^2+q+1)(q^2+q)+2$	$(C+2) + d_m = (q^2+q+3)+7$
Scheme in [48]	$n_j + C + 1 = (q^2+q) + (q^2+q+1) + 1$	$n_j + 2 = (q^2+q) + 2$

Memory usage

The memory storage requirement for loading each single CH is $((q^2+q)+\sqrt{(q^2+q+1)}) \times S_K$, where S_k is the key size in symmetric cryptography. Moreover, length of key chain for every CN in residual design is q. Therefore, memory storage requirement for CN is $q \times SK$. In order to do the comparison, networks size should suppose equal. Moreover we suppose the number of CNs in each CH is equal. Let network size to be $(q^2+q+1)^2$,

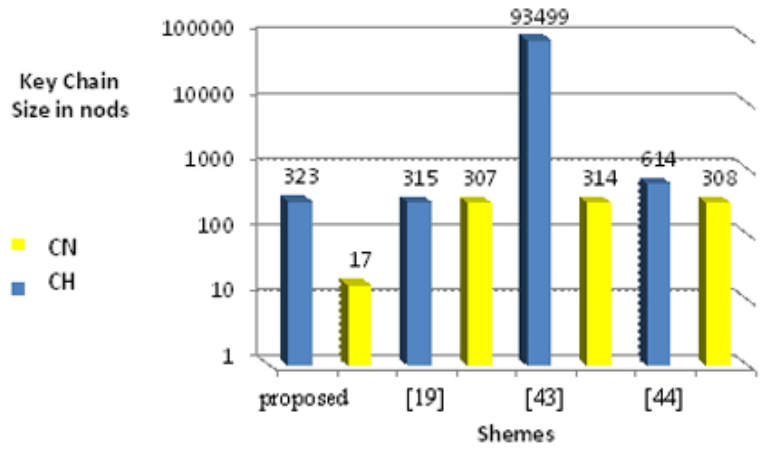

FIGURE 3 Number of required keys in each scheme at equal network size (90.000 sensor nodes)

i.e. (q^2+q+1) clusters, where each cluster has (q^2+q) nodes and one CH. A_K is the key size in public key cryptography, then, memory requirement for [47],[48] and [21] is given in Table (4) Here, d_m denotes maximum degree of neighborhood in [47] which is according to [47] equals to 7. Now, we use a numerical example to compare memory usage of the proposed scheme, [48], [21] and [47]. Let N=930 be number of CNs and C=31 be number of CHs in a hierarchical wireless sensor network. A (25,30,6,5,1)-BIBD design is used for each cluster. Also, ECC(163-bit) and RC5(80-bit) are respectively used for asymmetric and symmetric encryption. Table 5 demonstrates memory requirement for each CN and CH. Acquired values confirm that our proposed scheme reduces the memory usage in sensor nodes by $\approx$ 84%; the scheme in [48] needs memory storage 13 and 35 times more than our scheme for each CH and CN, respectively, and memory storage requirement for each CN and CH in [47] is about 15 and 52 times of our scheme. Though the memory consumption of CHs in the scheme that is proposed in [21] is close to our proposed method, but as CHs are placed in the fog layer and has more resources compared to the end devices, the memory usage in CHs is less important than CNs. Instead, the memory usage in CNs in [21] is around six times more than our proposed scheme that depicts our scheme outperforms the scheme in [21] for resource constrained end devices. Therefore, we can conclude that our proposed scheme has better performance compared to three other schemes in terms of memory consumption; it supports larger number of nodes in the network, while requiring less number of keys.

TABLE 4 Comparison of the Required Memory Storage

Scheme	Number of keys in each CH	Number of keys in each CN
Proposed scheme	$((q^2+q)+(\sqrt{(q^2+q+1)})) \times S_l$	$S_k \times q$
Scheme in [21]	$(\sqrt{(q^2+q+1)(q^2+1)} + \sqrt{(q^2+q+1)}) \times S_k$	$C=(q^2+q+1) \times S_k$
Scheme in [47]	$((q^2+q+1)(q^2+q)+2) \times A_k$	$(C+2)+d_m=((q^2+q+3) \times A_k)+(d_m \times S_k)$
Scheme in [48]	$((q^2+q)+(q^2+q+1)+1) \times A_k$	$((q^2+q)+2) \times A_k$

TABLE 5 Comparison of the Required Memory Storage (in bit)

Scheme	Required Memory by each CH	Required Memory by each CN
Proposed scheme	$(30+\sqrt{31}) \times 80 = 2880$	$50 \times 80 = 400$
Scheme in [21]	$(\sqrt{930}+\sqrt{30}) \times 80 = 2960$	$31 \times 80 = 2480$
Scheme in [47]	$(930+2) \times 160 = 149120$	$((31+2) \times 160)+(7 \times 160) = 5840$
Scheme in [48]	$(30+31+1) \times 160 = 99200$	$(30+2) \times 160 = 5120$

5.2.2 | Scalability

Network scalability can be measured by the maximum number of nodes that is supported in a scheme, given the same key pool and key chain length. The number of generated key chains is equal to the maximum number of nodes that a design can support. As each cluster forms a $(q^2, q^2+q, q+1, q, 1)-$ BIBD and the number of clusters equals to the size of points, i.e. $C = V = q^2+q+1$, and each of the clusters contains q^2+q nodes, a maximum number of $(q^2+q+1)(q^2+q)+(q^2+q+1) = (q^2+q+1)^2$ nodes can be supported by the proposed scheme. For example, if a network is composed of 2500 nodes, the smallest prime number that meets this requirement is q=7 which generates 3192 key chains (plus 57 cluster heads, totally 3249 nodes). Out of these generated key chains, 2500 key chains will be allocated to the sensor nodes and in the case of future network expansion, a key ring of size 7 remains that could be used for other 692 nodes that might be added in the future. Table (6) shows that the proposed scheme is more scalable compared to similar hierarchical schemes, such as [21], [48], and [47], as it supports larger network sizes requiring fewer number of keys in sensor nodes (I.e., CN). The proposed scheme may support 1000.000 nodes with only 31 keys, while other three schemes require more than 1000 keys for the same network size, which shows $\approx$ 96% improvement. However, the number of required CHs keys in our scheme is about one percent more than the number of required keys in [21] (see Table (7)). Analysis of this simple example shows that mapping residual design to key pre-distribution significantly improves the scalability of the scheme.

TABLE 6 Network Scalability-Comparison of required number of keys in CNs of different schemes to support Network with size N

scheme	Network Size					
	1×10^5	2×10^5	4×10^5	6×10^5	8×10^5	10×10^5
ProposedScheme	17	21	25	27	29	31
Scheme in [21]	316	447	632	775	894	1000
Scheme in [47]	323	454	639	782	901	1007
Scheme in [48]	317	448	633	776	895	1001

TABLE 7 Network scalability - comparison of required number of keys in CHs of different schemes to support Network with size N

scheme	Network Size					
	1×10^5	2×10^5	4×10^5	6×10^5	8×10^5	10×10^5
ProposedScheme	333	467	657	801	923	1031
Scheme in [21]	325	458	465	788	910	1016
Scheme in [47]	99686	199555	399369	599227	799108	999002
Scheme in [48]	632	894	1265	1549	1789	2000

5.2.3 | Network Connectivity

Two neighboring nodes with at least one shared key can directly communicate with each other. Let Pr1 be the probability of existence of a shared key between a pair of nodes in the network. To calculate Pr1, at first the local connectivity for each cluster $(Pr1_j)$ should be computed as summarized in 3 and then connectivity of the entire network can be obtained by computing weighted average of all clusters as discussed in 4. When using residual design for key pre-distribution, in each cluster, each key is contained $q+1$ key-chain from among q^2+q possible options (because each cluster is a $(q^2, q^2+q, q+1, q, 1)$ -BIBD. Consider CN_i^j and CN_z^j are randomly picked from cluster j, where CN_i^j is preloaded with q keys and CN_z^j is preloaded with another q distinct keys. Each keys in node CN_i^j is contained in other q (q+1-1=q) key-chains from q^2+q-1 existing keys. As $\lambda = 1$, so each pair of keys may share only one key-chain with each other. Therefore, key chains with two distinct keys from key ring CN_i^j are completely disjoint which means every node shares a key with $(r-1)\times k = q\times q$ ((2.2)) nodes from q^2+q-1 possible nodes. The probability of sharing a key in a cluster is calculated as follows.

$$Pr1_j = \frac{(q\times q)}{(q^2+q-1)} \tag{3}$$

As we mentioned earlier n_j is the total number of nodes in cluster j which equals to $q^2 + q$. Total network connectivity is given by 4.

$$Pr1 = \frac{\sum_{i=1}^{C} n_j \times Pr1_j}{N} = \frac{\sum_{i=1}^{C} n_j \times Pr1_j}{(q^2+q+1)(q^2+q)} = \frac{((q^2+q+1)(q^2+q) \times (q^2))}{((q^2+q+1)(q^2+q) \times (q^2+q-1))} \quad (4)$$

Finally the probability of sharing a common key can be calculated as:

$$pr1 = \frac{q^2}{(q^2+q-1)} \quad (5)$$

The evaluation of this solution shows clearly that the basic mapping from clustered residual design to key pre-distribution gives a high key sharing probability and reaches O(1) which show that probability of communication within a single hop is close to 1 (see Figure (4)). It is notable that two nodes from two distinct clusters may communicate through their CHs as they share a common key with all their CNs. Moreover, CHs may communicate with each using pre-distributed BIBD key.

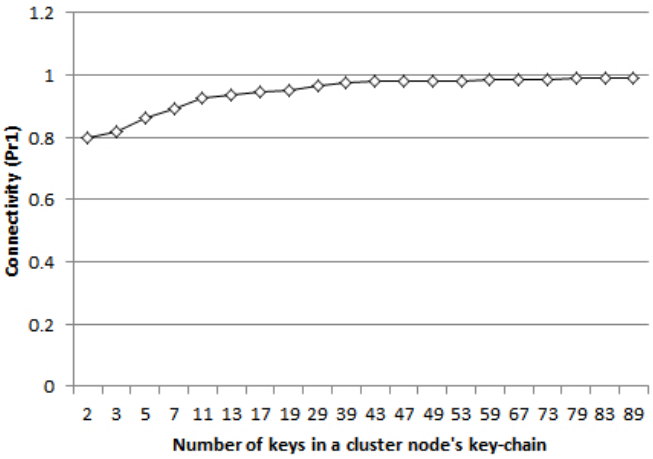

FIGURE 4 Probability of local connectivity versus the size of the key-chain

5.3 | Resilience Against Node Capture Attacks

This section examines resiliency of the proposed scheme against node capture attacks i.e. how secure is communication between benign nodes when $x(x \leq (q^2+q+1)^2)$ nodes are pawned by malicious actors. As discussed earlier, key pools in each cluster are separated after distribution. Therefore, a captured node in a cluster has no effects on other cluster members, reflecting sustainability of our scheme in node capture attacks. For example, suppose two nodes, v and u of cluster j are not captured. If x node from cluster j is attacked and decrypted, the probability that the adversary can decrypt the communication between u and v is calculated as follows. Based on definitions and symbols given in Table 8 , we seek to solve $P(L|C_x)$, i.e. what's the probability that any links between two uncaptured nodes being decrypted in case x nodes are captured in the given network. If key k is the point set of a cluster which is observed by q+1 node, the probability that a given link is secured with key k can be defined as

TABLE 8 Notation and definitions of resilience

Notation	Definition
C_x	Event that x nodes (key-rings) are captured.
D_k	Event that a block containing key k is compromised.
I_k	Event that a given link is secured with key k.
I	Event that a given link is secured.
L_k	Event that a link secured with key k is compromised.
L	Event that a link is compromised.

$$P(l_k|l) = \frac{\binom{(q+1)}{2}}{\binom{(q^2+q)(q^2+q+1)}{2}} \tag{6}$$

Also, the probability that x captured nodes in cluster j include key k is equal to

$$\mathrm{P}(\mathrm{D_k}|\mathrm{C_x}) = 1 - \mathrm{theprobabilityofnothavingkey} = 1 - \frac{\binom{(q^2+q)(q^2+q+1)-(q+1)}{x}}{\binom{(q^2+q)(q^2+q+1)}{x}} \tag{7}$$

The probability that a given link remains secure even when key k is compromised and x nodes are captured can be calculated as:

$$P(L_k|C_x) = P(l_k|l)P(D_k|C_x) \tag{8}$$

Finally, in the residual design, the probability that a link is compromised when an attacker captures x nodes can be calculated as:

$$\mathrm{P}(\mathrm{L}|\mathrm{C_x}) = \sum_{j=1}^{(q^2+q+1)} \frac{\binom{q+1}{2}}{\binom{(q^2+q+1)(q^2+q)}{2}} \left(1 - \frac{\binom{(q^2+q+1)(q^2+q)-(q+1)}{x}}{\binom{(q^2+q+1)(q^2+q)}{x}}\right) \tag{9}$$

We compare resilience of the proposed scheme against node capture attacks with a hierarchical scheme uses similar approach (combinatorial design) in key establishment, [21]. Table 9 shows a comparison between probability of node capture for [21] and the proposed scheme in case s nodes are captured with 960 and 17698 nodes in the Network. Compared to [21], with equal NetworkSize, the proposed scheme shows higher resistance against node capture attacks. This indicates that the proposed scheme promotes network resistance to node capture without restricting network connectivity.

TABLE 9 Comparison of resilience against node capture attack

Network size	Number of Captured nodes			
Scheme in [21]	10	100	200	300
960	0.3781	0.9914	0.9999	1
17689	0.2577	0.8614	0.9880	0.9973
Proposed scheme	10	100	200	300
960	0.022	0.216	0.416	0.536
17689	0.013	0.038	0.076	0.124

6 | CONCLUSION AND FUTURE WORK

In this paper, we proposed a key pre-distribution scheme for hierarchical architecture of fog networks. Our proposed key pre-distribution scheme facilitates secure communication between devices inside a fog cluster and between end devices and the fog nodes. We showed that our proposal is more efficient and scalable compared to the state-of-the-art combinatorial-based solutions for hierarchical networks. We also showed that the proposed scheme improves the memory overhead and network scalability significantly, compared to similar schemes in the literature.

Future research includes extending the proposed scheme to support mobility, and considering different attack scenarios to increase the network resilience. Another potential future research could be designing an efficient key management scheme for the cluster heads.

References

1. Osanaiye Opeyemi, Chen Shuo, Yan Zheng, Lu Rongxing, Choo Kim-Kwang Raymond, Dlodlo Mqhele. From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework. *IEEE Access*. 2017;5:8284–8300.

2. Bonomi Flavio, Milito Rodolfo, Zhu Jiang, Addepalli Sateesh. Fog computing and its role in the internet of things. In: :13–16ACM; 2012.

3. Naranjo Paola G Vinueza, Pooranian Zahra, Shojafar Mohammad, Conti Mauro, Buyya Rajkumar. FOCAN: A Fog-supported Smart City Network Architecture for Management of Applications in the Internet of Everything Environments. *arXiv preprint arXiv:1710.01801*. 2017;.

4. Iorga Michaela, Feldman Larry, Barton Robert, J. Martin Michael, Goren Nedim, Mahmoudi Charif. *The NIST Definition of Fog Computing- NIST Special Publication 800-191 (Draft).* : NIST - National Institute of Standards and Technology; 2017.
5. Yi Shanhe, Hao Zijiang, Qin Zhengrui, Li Qun. Fog computing: Platform and applications. In: :73–78IEEE; 2015.
6. Shi Weisong, Dustdar Schahram. The promise of edge computing. *Computer.* 2016;49(5):78–81.
7. *White paper: Fog Computing and the Internet of Things- Extend the Cloud to Where the Things Are.* : CISCO; 2015.
8. Computing Fog. *the Internet of Things: Extend the Cloud to Where the Things Are.* 2016.
9. Yi Shanhe, Qin Zhengrui, Li Qun. Security and privacy issues of fog computing: A survey. In: :685–695Springer; 2015.
10. Mukherjee Mithun, Shu Lei, Wang Di. Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges. *IEEE Communications Surveys & Tutorials.* 2018;.
11. Tang Bo, Chen Zhen, Hefferman Gerald, Wei Tao, He Haibo, Yang Qing. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In: :28ACM; 2015.
12. Khan Saad, Parkinson Simon, Qin Yongrui. Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing.* 2017;6(1):19.
13. Simplício Marcos A, Barreto Paulo SLM, Margi Cintia B, Carvalho Tereza CMB. A survey on key management mechanisms for distributed wireless sensor networks. *Computer networks.* 2010;54(15):2591–2612.
14. Chen Chi-Yuan, Chao Han-Chieh. A survey of key distribution in wireless sensor networks. *Security and Communication Networks.* 2014;7(12):2495–2508.
15. Conti Mauro. Secure Data Aggregation. In: Springer 2016 (pp. 101–124).
16. Çamtepe Seyit A, Yener Blent. Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Transactions on networking.* 2007;15(2):346–358.
17. Eschenauer Laurent, Gligor Virgil D. A key-management scheme for distributed sensor networks. In: :41–47ACM; 2002.
18. Ruj Sushmita, Roy Bimal. Key pre-distribution using partially balanced designs in wireless sensor networks. *International Journal of High Performance Computing and Networking.* 2011;7(1):19–28.
19. Dong Jun-Wu, Pei Ding-Yi, Wang Xue-Li. A class of key predistribution schemes based on orthogonal arrays. *Journal of Computer Science and Technology.* 2008;23(5):825–831.
20. Gupta Piyush, Kumar Panganmala R. The capacity of wireless networks. *IEEE Transactions on information theory.* 2000;46(2):388–404.
21. Javanbakht Masoumeh, Erfani Hossein, Javadi Hamid Haj Seyyed, Daneshjoo Parisa. Key predistribution scheme for clustered hierarchical wireless sensor networks based on combinatorial designs. *Security and Communication Networks.* 2014;7(11):2003–2014.
22. Huang Jen-Yan, Liao I-En, Tang Hao-Wen. A forward authentication key management scheme for heterogeneous sensor networks. *EURASIP Journal on Wireless Communications and Networking.* 2011;2011(1):296704.
23. Lu Kejie, Qian Yi, Guizani Mohsen, Chen Hsiao-Hwa. A framework for a distributed key management scheme in heterogeneous wireless sensor networks. *IEEE Transactions on Wireless Communications.* 2008;7(2).
24. Chan Haowen, Perrig Adrian, Song Dawn. Random key predistribution schemes for sensor networks. In: :197–213IEEE; 2003.
25. Lee Jooyoung, Stinson Douglas R. A combinatorial approach to key predistribution for distributed sensor networks. In: :1200–1205IEEE; 2005.
26. Camtepe Seyit A, Yener Bulent. Key distribution mechanisms for wireless sensor networks: a survey. *Rensselaer Polytechnic Institute, Troy, New York, Technical Report.* 2005;:05–07.
27. Stojmenovic Ivan, Wen Sheng. The fog computing paradigm: Scenarios and security issues. In: :1–8IEEE; 2014.

28. Yu Shucheng, Wang Cong, Ren Kui, Lou Wenjing. Achieving secure, scalable, and fine-grained data access control in cloud computing. In: :1–9Ieee; 2010.

29. Kumar Manoj. An Enhanced Remote User Authentication Scheme with Smart Card.. *IJ Network Security.* 2010;10(3):175–184.

30. Tsai Jia Lun. Efficient Nonce-based Authentication Scheme for Session Initiation Protocol.. *IJ Network Security.* 2009;9(1):12–16.

31. Mukherjee Mithun, Matam Rakesh, Shu Lei, et al. Security and privacy in fog computing: Challenges. *IEEE Access.* 2017;5:19293–19304.

32. Alrawais Arwa, Alhothaily Abdulrahman, Hu Chunqiang, Xing Xiaoshuang, Cheng Xiuzhen. An attribute-based encryption scheme to secure fog communications. *IEEE Access.* 2017;5:9131–9138.

33. Dsouza Clinton, Ahn Gail-Joon, Taguinod Marthony. Policy-driven security management for fog computing: Preliminary framework and a case study. In: :16–23IEEE; 2014.

34. Du Wenliang, Deng Jing, Han Yunghsiang S, Varshney Pramod K. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on dependable and secure computing.* 2006;3(1):62–77.

35. Chan Haowen, Perrig Adrian, Song Dawn. Random key predistribution schemes for sensor networks. In: :197–213IEEE; 2003.

36. Ito Takashi, Ohta Hidenori, Matsuda Nori, Yoneda Takeshi. A key pre-distribution scheme for secure sensor networks using probability density function of node deployment. In: :69–75ACM; 2005.

37. Hwang David D, Lai Bo-Cheng Charles, Verbauwhede Ingrid. Energy-memory-security tradeoffs in distributed sensor networks. *Ad-Hoc, Mobile, and Wireless Networks.* 2004;:630–630.

38. Lee J, Stinson DR. Article 5 (36 pages)-On the Construction of Practical Key Predistribution Schemes for Distributed Sensor Networks Using Combinatorial Designs. *ACM Transactions on Information and System Security-TISSEC.* 2008;11(2).

39. Rabin Michael O. *Digitalized signatures and public-key functions as intractable as factorization.* : MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE; 1979.

40. Gaubatz Gunnar, Kaps J-P, Ozturk Erdinc, Sunar Berk. State of the art in ultra-low power public key cryptography for wireless sensor networks. In: :146–150IEEE; 2005.

41. Liu Jian, Li Jiangtao, Zhang Lei, et al. Secure intelligent traffic light control using fog computing. *Future Generation Computer Systems.* 2018;78:817–824.

42. Kothmayr Thomas, Schmitt Corinna, Hu Wen, Brünig Michael, Carle Georg. A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. In: :956–963IEEE; 2012.

43. Raza Shahid, Shafagh Hossein, Hewage Kasun, Hummen René, Voigt Thiemo. Lithe: Lightweight secure CoAP for the internet of things. *IEEE Sensors Journal.* 2013;13(10):3711–3720.

44. Granjal Jorge, Monteiro Edmundo, Silva Jorge Sa. End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication. In: :1–9IEEE; 2013.

45. Hummen René, Ziegeldorf Jan H, Shafagh Hossein, Raza Shahid, Wehrle Klaus. Towards viable certificate-based authentication for the internet of things. In: :37–42ACM; 2013.

46. Ray Sangram, Biswas GP. Establishment of ECC-based initial secrecy usable for IKE implementation. In: ; 2012.

47. Azarderskhsh Reza, Reyhani-Masoleh Arash. Secure clustering and symmetric key establishment in heterogeneous wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking.* 2011;2011(1):893592.

48. Riaz Rabia, Naureen Ayesha, Akram Attiya, Akbar Ali Hammad, Kim Ki-Hyung, Ahmed H Farooq. A unified security framework with three key management schemes for wireless sensor networks. *Computer Communications.* 2008;31(18):4269–4280.

49. Gandino Filippo, Ferrero Renato, Montrucchio Bartolomeo, Rebaudengo Maurizio. Fast Hierarchical Key Management Scheme With Transitory Master Key for Wireless Sensor Networks. *IEEE Internet of Things Journal.* 2016;3(6):1334–1345.

50. Stinson Douglas R. Combinatorial designs: constructions and analysis.. *Sigact News.* 2008;39(4):17–21.

51. Anderson Ian. *Combinatorial designs: construction methods.* Ellis Horwood; 1990.

52. Kumar Pardeep, Porambage Pawani, Ylianttila Mika, Gurtov Andrei. A Mobile Object-based Secret Key Distribution Scheme for Wireless Sensor Networks. In: :656–661IEEE; 2013.

53. Modiri Vahid, Javadi Hamid Haj Seyyed, Anzani Mohaddese. A Novel Scalable Key Pre-distribution Scheme for Wireless Sensor Networks Based on Residual Design. *Wireless Personal Communications.* 2017;96(2):2821–2841.

54. Tong Liang, Li Yong, Gao Wei. A hierarchical edge cloud architecture for mobile computing. In: :1–9IEEE; 2016.

55. Hussen Hassen Redwan, Tizazu Gebere Akele, Ting Miao, Lee Taekkyeun, Choi Youngjun, Kim Ki-Hyung. SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN). In: :246–251IEEE; 2013.

56. Saied Yosra Ben, Olivereau Alexis. D-HIP: A distributed key exchange scheme for HIP-based Internet of Things. In: :1–7IEEE; 2012.

57. Nguyen Kim Thuat, Laurent Maryline, Oualha Nouha. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks.* 2015;32:17–31.

58. Iyengar S Sitharama, Brooks Richard R. *Distributed sensor networks: sensor networking and applications.* CRC press; 2016.

59. Yu Hong, He Jingsha, Zhang Ting, Xiao Peng, Zhang Yuqiang. Enabling end-to-end secure communication between wireless sensor networks and the Internet. *World Wide Web.* 2013;16(4):515–540.

60. Turkanović Muhamed, Brumen Boštjan, Hölbl Marko. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks.* 2014;20:96–112.

61. Vu Tuan Manh, Williamson Carey, Safavi-Naini Reihaneh. Simulation modeling of secure wireless sensor networks. In: :30ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering); 2009.

62. Traynor Patrick, Kumar Raju, Choi Heesook, Cao Guohong, Zhu Sencun, La Porta Thomas. Efficient hybrid security mechanisms for heterogeneous sensor networks. *IEEE Transactions on mobile computing.* 2007;6(6).

63. Pattanayak Anupam, Majhi B. Key Predistribution Schemes in Distributed Wireless Sensor Network using Combinatorial Designs Revisited.. *IACR Cryptology ePrint Archive.* 2009;2009:131.

64. Wang Yong, Attebury Garhan, Ramamurthy Byrav. A survey of security issues in wireless sensor networks. 2006;.