Journal of Optimization in Soft Computing (JOSC)

Vol. **2**, Issue **2**, pp: (23-35), **Summer**-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Paper Type (Research paper)

Key Pre-distribution Based on Block Complementation Design in Internet of Things Security

Vahid Chegeni[1*], Hamid Haj Seyyed javadi[2], Mohammad Reza Moazami Goudarzi[3]

1. Department of Computer Engineering, Khorramabad Branch, Islamic Azad University, Khorramabad, Iran
2. Department of Mathematics and Computer Science, Shahed University, Tehran, Iran
3. Department of Mathematics, Borujerd Branch, Islamic Azad University, Borujerd, Iran

Article Info

Article History:
Received: 2024/08/12
Revised: 2024/09/18
Accepted: 2024/09/22

DOI:

Keywords:
Internet of Things (IoT), Combinatorial Designs, Residual Design, Block Complementation, key management, cryptography.

**Corresponding Author's Email Address: vahid.chegeni@iau.ac.ir*

Abstract

The Internet of Things is a network of smart devices that can connect and exchange data with other things. Due to the heterogeneous nature of IoT devices and constrained resources, creating a secure connection between IoT devices is very important. The use of previous algorithms for encryption, such as RSA and AES, involves complex and heavy computation and is unsuitable. Therefore, lightweight encryption methods are required. This paper presents a new and essential pre-distribution scheme proposed to attain high security. This scheme is based on a design derived from combinatorial algebra, namely the residual design. According to this scheme, each device in IoT will have a set of keys called the key ring from a key pool assigned to it. It should be noted that the residual design that is built from block complementation is being used in the IoT for the first time. A basic mapping from residual design to key pre-distribution is illustrated. Another advantage of this approach is improving the IoT resilience while maintaining high scalability. The evaluations performed indicate that our approach leads to an improvement in secure connectivity and an increase in IoT scalability with high resilience.

1. Introduction

The Internet of Things (IoT) is defined as a network of smart devices that share information through interacting with one another. "Things" refers to any physical object with a device with a unique IP address. This device can connect to a network to send and receive data. The IoT is now used to define several things, such as the convergence of multiple technologies, real-time analytics, machine learning, object sensors, and implant systems. The IoT can be considered a framework for the Smart City and Smart Energy Management Systems that are widely used today. Each IoT has various devices, such as sensors, actuators, RFID tags, and smartphones or backend servers, which vary in size, capability, and functionality. The 6LoWPAN has recently been used to help even the smallest devices connect to the Internet. The idea behind 6LoWPAN is that everything is expected to support the TCP/IP protocol stack and join the IoT. Making the IPv6/RPL connected 6LoWPANs secure is challenging since the devices are connected to the untrusted Internet. Also, the resources used are constrained, and the communication links are lossy. A second challenge is related to the limitations and constraints of IoT devices in terms

of memory and processing power. These limitations, in turn, indicate a restriction in the size of keys, IDs, and Rings [1].

The sensors on the IoT, in addition to the senses, can process and store sensed events. They can even intelligently recognize if a sensed event is a repeating one. IoT combines multiple technologies such as RFID, wireless sensor networks (WSN), NFC, etc. WSN is a subset of IoT. IoT is responsible for data processing, manipulation, and decision-making. In IoT, the data is sent to the Internet in only one hop. First, in IoT, routing is not implemented. Sensors send their data directly to the Internet because they have an Internet connection. In IoT, each device is identifiable with a unique ID: its IP address [2].

IoT is gradually becoming a significant part of different aspects of our lives. It is used in smart homes, wearable devices, healthcare, etc. Its wide range of applications yields common data, such as the enormous value of user's private information. Hence, the security of this information is very consequential. Several factors, such as data confidentiality, data integrity, authentication, access control, and privacy, are required to provide security for the IoT. It should also be noted that the authentication of IoT devices is of particular significance [3].

We need to use proper cryptographic methods to achieve high security in IoT. Cryptographic methods are divided into two categories, including symmetric key and asymmetric key cryptography. Asymmetric key cryptography, such as RSA and ECC (Elliptic Curve Cryptography), requires high computational cost, more processing time, and larger key sizes. Thus, for these reasons, the use of asymmetric key cryptography in IoT is limited, and as a result, symmetric-key cryptography is recommended [4].

Since IoT security depends on the method used to distribute keys between IoT devices, an effective key establishment method is required to distribute the cryptographic keys between the IoT devices. As already mentioned, public-key cryptography requires high computational costs. So, key pre-distribution is a solution to the key establishment problem in IoT, where each device is pre-loaded with a finite set of keys before deployment. The key pre-distribution scheme (KPS) determines which nodes store which keys [5]. Traditional cryptographic methods cannot provide authentication in their present form for the expected 50 billion devices. Using related technologies such as DES, 3DES, and AES to encrypt resource-constrained IoT devices requires too much energy [6] [7].

An IoT has various criteria that analyze key distribution solutions, such as memory overhead, connectivity, scalability, resilience, and communication overhead. Memory overhead is the required memory to store keys in every device. Connectivity refers to the probability of a shared key between two nodes. Resilience is the persistence of the IoT against node capture. Scalability is the maximum network size supported by a KPS, and communication overhead is the number of messages sent between nodes [8].

A KPS uses three methods: random, deterministic, and hybrid. The first schemes require that the keys be selected randomly from a key pool and stored in each object. This method will not guarantee the direct communication of each two nodes. Lack of direct communication creates a path between the two nodes, reducing communication speed. Also, a deterministic method should be used to design a key pool and key rings to achieve better key connectivity. A combination of both the deterministic and random approaches creates a hybrid method that can be used to improve scalability and resiliency [9].

The key pre-distribution scheme is a good solution for IoT security and is used in most research studies. Each KPS has three phases:

1) Pre-distribution
2) Shared-key discovery
3) Path-key establishment

A key pool is produced during the first phase. Subsequently, a subset of the key pool, namely the *key ring*, is assigned to each sensor node. The second phase is carried out after the deployment of sensor nodes. Each pair of nodes must communicate with each other to find at least one shared key between them. In the final phase, the two nodes without a shared key that want to communicate with each other may create a secure path using one or more intermediate nodes in which each pair of nodes shares a standard key [10].

Applying combinatorial designs in KPS with proper parameters can cause a decrease in the length of the key path and increase the maximum connectivity. The solid mathematical structure of combinatorial designs results in the communicational algorithms that can be reduced to $O(1)$ in the path-key establishment and shared-key discovery phases.

The IoT wants to convert traditional devices into connected devices by using interchanging data and communications to monitor and control the devices. To get the required security on the IoT, we must consider the following challenges [11] [12]:

- **Resource constraints:** IoT devices frequently operate on channels with low-bandwidth communication. Therefore, it is impossible to execute directly standard conventional security protocols of the Internet in the context of IoT.
- **Resilience to attacks:** IoT devices are typically small and inexpensive, with low physical protection. For example, a mobile device can be stolen, or fixed devices can be moved.
- **Scalability:** The IoT is universally composed of a large number of devices. The proposed security approach must be able to scale all those included.

This paper focuses on combinatorial constructions for key pre-distribution schemes in IoT. To improve IoT

resilience while maintaining high scalability and secure connectivity, we illustrated a novel basic mapping from residual design based on block complementation to key pre-distribution. The residual design is constructed from block complementation and is used on the IoT for the first time. The new approach has been analyzed and compared analytically and experimentally with other state-of-the-art KPSs, examining various evaluation criteria. It was indicated that the proposed scheme amended network scalability and decreased memory overhead compared to other works [13].

The rest of the paper is organized as follows. In section 2, related work is summarized. In section 3, we provide a brief overview of key pre-distribution and combinatorial design theory. In sections 4 and 5, we introduce and analyze the proposed scheme and present how to map the residual design to key distribution. Section 6 presents the implementation and simulation results. Finally, Section 7 concludes the paper.

2. RELATED WORK

A q-composite random key pre-distribution scheme has been proposed by Chan et al. [14]. This scheme can enhance the security of communication between the two nodes. Based on this scheme, every two nodes may create a secure link on condition that they have at least q shared keys. Qian proposed A key pre-distribution scheme [15]. This scheme contained a hash function to improve the resilience against node capture attacks. Recently, binational design in key pre-distribution has been proposed as a solution. In this study, a -PBIBD combinatorial design is introduced and constructed, and the mapping of such design as a key pre-distribution scheme in the resource-constrained IoT network is explained. Using such a pre-distribution scheme, more keys are obtained for communication between two devices in the IoT network [16] [17].

A new key pre-distribution scheme named POK (adaPtive and rObust Key pre-distribution) is presented in [18]. POK improves the way keys are generated and pre-loaded in the sensor nodes. The main idea of the POK is that newly added sensor nodes will be pre-loaded with pairwise keys computed by using a hash function and having knowledge of the number of future post-deployments. A comparison study with related works concludes that POK offers less communication overhead and doesn't require time synchronization, leading to an energy-efficient scheme.

Different encryption and hash algorithms were proposed by Vinayaga et al. [19] to enhance the security of smart home systems. Their algorithms were designed to secure any communication between the devices within an IoT System. Thus, a hash algorithm was created based on RC4, and its efficiency was measured against the existing hash algorithms.

In [20], a key pre-distribution scheme has been proposed based on the combinatorial design for IoT. This scheme has increased the scalability of the network. For the proposed scheme, a kind of mapping from the unital design to the key establishment has been proposed, which yields a network with high scalability. The results indicate that the proposed scheme increases network scalability considerably with high resilience.

To combine security in an IoT-based Smart Home System, Santoso et al. [21] proposed a method to maintain user comfort. Their paper explained how to implement a WiFi-based IoT Smart Home system, including IoT devices such as sensors, actuators, and equipment. These devices were connected to the Home Gateway over the Home network. They designed a user device to control and monitor the system. This device was connected to a Home gateway over the Internet. The home gateway made it possible for IoT devices to communicate securely. Also, it allowed users to access, configure, and control the system via the user interface. It is an open-source IoT framework containing various libraries from cryptography (ECC, AES, etc).

In [22], the current cryptographic methods, such as the Advanced Encryption Standard (AES) and the Elliptic Curve Cryptography (ECC), are expounded, and their functionality, together with their advantages and disadvantages, are discussed. Also, this paper highlights the need for more flexible cryptographic suites.

The term security is a vital issue in any sensor network. In these networks, key management is considered the main security service. Due to the limitations on sensor nodes, traditional key management techniques do not fit with sensor networks. A new key pre-distribution scheme was proposed in [23] using multivariate polynomials to establish the pairwise key in sensor networks. Based on this approach, the combinatorial design theory must be applied in the multivariate key pre-distribution scheme. In this scheme, the common multivariate polynomials can be stored in sensor nodes before deploying the network. This idea is done using the identifier of sensors and the combinatorial design. Also, compared to previous schemes, the proposed approach receives better security in terms of resilience against node capture with the exclusion of additional communication overhead.

In [24], an advanced key administration framework for remote sensor networks is proposed, consolidating fuzzy logic and AES encryption to improve the performance of the WSNs. The proposed framework uses fuzzy logic for cluster formation and head rotation and utilizes the AES algorithm to encode the information. It falls in the classification of techniques that depend on hierarchical structures, in which the sensor nodes use pre-distribution and post-deployment mechanisms to distribute keys. The proposed key management uses fuzzy logic, which enhances security and energy effi-

ciency. Thus, the energy utilized by the network is reduced, and the network's lifetime is improved.

A key pre-distribution scheme was presented in [25] for a clustered heterogeneous WSN using transversal designs. In this novel scheme, key rings are assigned to sensor nodes before the network is deployed, and a key pool of each cluster is separated by adding a pseudo-random-generated number after the network is deployed. The efficiency evaluation and security analysis results suggested that the proposed scheme, compared to other key management schemes, can provide better security and considerably reduce communication overhead and memory space without losing connectivity.

The residual design, a novel combinatorial approach, was proposed in [26] for key establishment. This approach requires that WSNs have a highly scalable key management scheme. The scheme is intended to provide highly secure connectivity. This scheme implies that the residual design undergoes a basic key pre-distribution mapping with high network scalability. It should be noted that this mapping lacks high resilience. Accordingly, a new approach should be designed for key pre-distribution based on the residual design to improve network resilience, maintain connectivity, and high scalability. Results suggest that the use of this approach leads to a reduction in computational cost and memory overhead. Although this approach provides the same connectivity based on the first scheme, the analysis and numerical results suggest that the optimized approach yields better network resilience. At the same time, it leads to lower network scalability against the residual design key pre-distribution scheme at an equal key-ring size [27].

In [28], the authors propose a new key pre-distribution scheme for wireless sensor networks based on combinatorial design. The proposed scheme divides the WSN into cells of the same size, where the sensor nodes are distributed evenly. Each cell has two types of sensor nodes, including the cluster head and the sensor node. The communication within the cell is direct; the communication between the nodes of the different cells is done through the cluster head. This scheme would reduce the key storage overhead and increase overall network resistance.

With a symmetric key, shared key allocation methods could be accomplished in cryptography before or after the network deployment. The one that occurs before the deployment is called the key pre-distribution. The Key Pre-distribution Schemes (KPSs) are the most desirable choices due to their limited computational costs and constrained energy and communication capacities of end devices. Therefore, keys are assigned to the end device's memory before their distribution in the network. According to these schemes, every pair of nodes can usually communicate securely because of the shared common credential(s) [29].

Cryptography schemes such as asymmetric or public keys normally facilitate secure communication between objects. Of course, it is not advisable to use these schemes for the sake of deployment on low-power battery operating devices. This is because they are required to compute costly cryptographic operation(s). However, the approaches proposed in [30] to reduce the number of exchanged messages are intended to design asymmetric key schemes for environments with resource constraints, such as the IoT. Other researchers [31] revealed that asymmetric solutions should be used for resource-constrained devices. The reason is that they have acceptable flexibility and scalability regarding shared key management.

Camtepe and Yener have proposed combinatorial designs for key pre-distribution in WSNn [32]. Their paper presented a new deterministic KPS based on the Symmetric Balanced Incomplete Block Design (SBIBD). The SBIBD is mapped onto the key pre-distribution to create $m^2 + m + 1$ key-rings from a key pool S of $m^2 + m + 1$ keys. There are $k = m + 1$ keys in each key-ring. Also, precisely one common key is shared by every two key rings. The main advantage of the Camtepe scheme is that every two nodes share exactly one common key. However, SBIBD schemes do not match extensive networks. To construct roughly $m^2 + m + 1$ key-rings, key rings of $m+1$ keys should be used. In the article [33], the SBIBD-based key pre-distribution was used to guarantee intra-region secure communications in grid group WSNs.

In [14], a perfect network resilience was proposed by Chan et al. aimed at obtaining network scalability of $O(k)$ where k is the key-ring size. The SBIBD [32] could also obtain network scalability of $O(k^2)$. For this reason, the unital design theory was used to pre-distribute keys. Their paper proposed mapping from units to key pre-distribution to achieve a good trade-off between scalability and connectivity. Hence, the method proposed in their paper was designed to improve network resilience against node capture attacks.

Contrary to wireless sensor network security, security in the IoT involves end-to-end communications. The IoT devices deny the possibility of defining static client and server roles. The devices in IoT act alternatively as a client and a server. Every IoT device has four criteria: the number of exchanged messages, the required bandwidth, the complexity of computations, and the possibility of pre-computations. These criteria are important in the cryptographic protocol. They only matter when they have to be implemented by highly resource-constrained devices. A good metric for these nodes is the overall energy consumption induced by both computations and message exchanges. Fig. 1 shows some applications of IoT devices. As can be seen from Fig. 1, secure communication is vital in every IoT device.

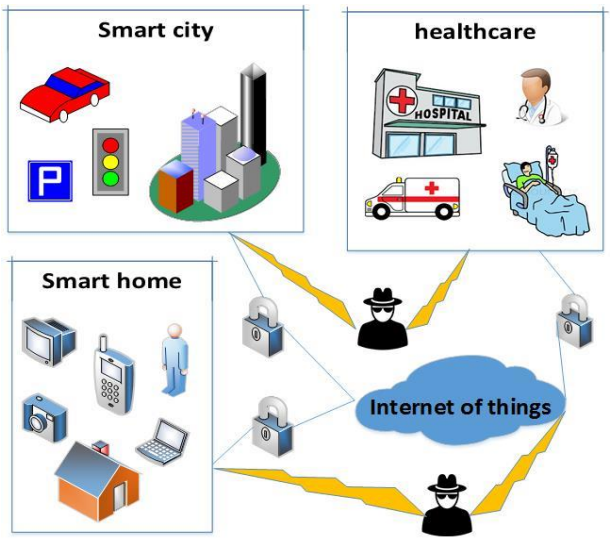

Fig. 1. Secure communication is essential in the IoT

3. Key Pre-Distribution and Combinatorial Design

3.1. Key Pre-distribution

Key management is the techniques and procedures for establishing secure communications between authorized parties. It is vital for a secure connection in IoT. Creating secret keys between sensor nodes is exceptionally challenging due to resource constraints (energy, CPU, and memory) on the nodes.

Key management includes four essential functions: analysis, assignment, generation, and distribution of network keys, such as Fig 2. A central server is responsible for storing and distributing the key pool. In a symmetric key algorithm, the keys must be chosen carefully, distributed, and stored securely.

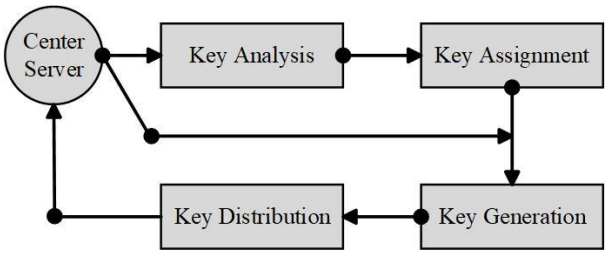

Fig. 2. Key Management process.

1. **Key analysis:** First, the number of keys required for the network, as well as the number of keys needed for each node, are analyzed.
2. **Key assignment:** This step refers to mapping keys to different parts. In this case, a key assignment manager assigns the key to the parts that want to create a secure communication channel. In this case, a key manager determines how many keys are assigned to each node to create a secure communication channel.
3. **Key generation:** This step may occur once or several times over the network's life. In the static key distribution scheme, the keys are generated by a central server and loaded in nodes before network development.
4. **Key distribution:** This step involves delivering the generated keys to predefined nodes.
   When an attack occurs, the above steps are repeated to ensure the network's security.

Key pre-distribution is the most effective technique to establish secure communication between nodes. Based on the key pre-distribution scheme, we should assign determined keys to each sensor before deployment. Before deployment, each sensor node should be preloaded with a set of keys from the large pool. Based on key pre-distribution, every two nodes with at least one shared key can create a communication path with another node. Given the features of the IoT, the use of key pre-distribution yields better results. A KPS includes three phases: key pre-distribution, shared key discovery, and path-key establishment. Security keys should be created and allocated to nodes during these three phases. Two nodes should detect one or more shared keys to make a secure connection. These keys are diverse in each KPS, and then communication is done between nodes using these shared keys.

The IoT includes devices with constrained resources that suffer from low memory capacity. Nonetheless, in most methods, the size of the key rings is related to the network size. Most of the existing techniques suffer from low scalability and memory overload. This problem led us to use a combinatorial design, especially residual design theory. To further expand, we will start with the definition of *block complementation* and the features of *residual design* theory. Afterward, we will propose the basic mapping from residual design to key pre-distribution and evaluate its performance metrics [34].

3.2. Combinatorial Design

Combinatorial design theory deals with arranging elements into subsets satisfying some generalized concepts of balance and symmetry. We focus primarily on the definition and properties of a particular kind of design, Balanced Incomplete Block Designs (BIBD) and symmetric BIBD. This paper defines a projective plane and blocks its complementation. Then, we build a residual design from block complementation.

3.2.1 Symmetric BIBD

A BIBD is a design (X, A) with positive integer parameters v, k, and λ such that $v > k$. Therefore, a $(v, k, \lambda) - BIBD$ is a design that $|X| = v$, and each block includes exactly k elements (points), and every both distinct points is included in precisely λ block. In definition, X is a set of points $X = \{x_1, x_2, \dots, x_v\}$, and $A = \{A_1, A_2, \dots, A_b\}$ is a collection of non-empty subsets of X called blocks. Generally, a BIBD contains v

distinct objects into *b* blocks with size *k*, so each object includes exactly r various blocks, and every different point occurs together in exactly λ blocks. Then, the design is explained as (v, k, λ), or equivalently (v, b, r, k, λ), where [35] [36]:

$$\begin{aligned} \lambda(v-1) &= r(k-1) \\ bk &= vr \end{aligned} \tag{1}$$

A *Symmetric BIBD* or *Symmetric Design* is a *BIBD* with $b = v$ and therefore $k = r$. In *Symmetric Design*, every block includes $k = r$ points, every object is contained in $r = k$ blocks, and every pair of objects is included in λ blocks, and finally, every pair of blocks intersects in λ objects. This paper uses a subset of Symmetric Designs called a Projective Plane.

3.2.2 block complementation

In this study, we use a *projective plane* with parameters $(q^2 + q + 1, q + 1, 1)$ where $q \geq 2$ and q is a prime number. Here, we state one method of constructing new BIBDs from old BIBD that is called *block complementation*. Suppose (X, A) is a $(v, b, r, k, \lambda) - BIBD$, where $k \leq v$-2. Then *block complementation* is done by replacing every block $A_i \epsilon A$ by $X \backslash A_i$ for $1 \leq i \leq v$. This created design is a $BIBD$ with parameters $(v, b, b - r, v - k, b - 2r + \lambda)$ [37].

Example 1: consider a *projective plane* with order $q = 2$, $(7,3,1) - BIBD$; then we construct *block complementation* with parameters $(7,4,2) - BIBD$. The element set and blocks of both designs are as follows:

Projective plane $(7,7,3,3,1) - BIBD$:

$$\begin{aligned} &X = \{1,2,3,4,5,6,7\} \\ &A_1 = \{1,2,4\}, A_2 = \{2,3,5\}, A_3 = \{3,4,6\}, \\ &A_4 = \{4,5,7\}, A_5 = \{1,5,6\}, A_6 = \{2,6,7\}, \\ &A_7 = \{1,3,7\} \end{aligned} \tag{2}$$

Block complementation $(7,7,4,4,2) - BIBD$:

$$\begin{aligned} &X = \{1,2,3,4,5,6,7\} \\ &A'_1 = \{3,5,6,7\}, A'_2 = \{1,4,6,7\}, A'_3 = \{1,2,5,7\}, \\ &A'_4 = \{1,2,3,6\}, \quad A'_5 = \{2,3,4,7\}, \quad A'_6 = \{1,3,4,5\}, \\ &A'_7 = \{2,4,5,6\} \end{aligned} \tag{3}$$

Then, using block complementation, we build residual design sets, as described in the following section.

3.2.3 Residual Design

Given a symmetric $(v, k, \lambda) - BIBD$ with elements $X = \{x_1, x_2, \dots, x_v\}$ and blocks $A = \{A_1, A_2, \dots, A_v\}$, then for every $1 \leq i \leq v$, fixing a block A_i Deleting this block and its elements from all other blocks of SBIBID constructs a new BIBD called *Residual Design*. That is, for any i, $\{A_1 \backslash A_i, A_2 \backslash A_i, \dots, A_v \backslash A_i\}$ are the blocks of a $(v - k, v - 1, k, k - \lambda, \lambda) - BIBD$ of the element set $X \backslash A_i$ Provided that $\lambda \neq k - 1$ [38] [39].

Based on the projective plane with parameters $(q^2 + q + 1, q + 1, 1)$ and the definition of the block complementation, we create a $BIBD$ with parameters $(q^2 + q + 1,\ q^2, q^2 - q)$. Then, we create a residual design based on the new $BIBD$.

Example 2: Consider $(7,4,2) - SBIBD$ in example 1. Then, we can create seven classes of residual sets for any A_i, where each building is a $(3,6,4,2,2) - BIBD$ over the element set $X \backslash A_i$. Therefore, by considering A_i as a fixed block in each class C_i, we have seven classes of residual sets, including the following blocks:

$$\begin{aligned} &C_1 = X \backslash A_1 = \{1,2,4\}, \ A_2 \backslash A_1 = \{1,4\}, \ A_3 \backslash A_1 = \{1,2\}, \\ &A_4 \backslash A_1 = \{1,2\}, \quad A_5 \backslash A_1 = \{2,4\}, \quad A_6 \backslash A_1 = \{1,4\}, \\ &A_7 \backslash A_1 = \{2,4\}. \\ &C_2 = X \backslash A_2 = \{2,3,5\}, \ A_1 \backslash A_2 = \{3,5\}, \ A_3 \backslash A_2 = \{2,5\}, \\ &A_4 \backslash A_2 = \{2,3\}, \quad A_5 \backslash A_2 = \{2,3\}, \quad A_6 \backslash A_2 = \{3,5\}, \\ &A_7 \backslash A_2 = \{2,5\}. \\ &C_3 = X \backslash A_3 = \{3,4,6\}, \ A_1 \backslash A_3 = \{3,6\}, \ A_2 \backslash A_3 = \{4,6\}, \\ &A_4 \backslash A_3 = \{3,6\}, \quad A_5 \backslash A_3 = \{3,4\}, \quad A_6 \backslash A_3 = \{3,4\}, \\ &A_7 \backslash A_3 = \{4,6\}. \\ &C_4 = X \backslash A_4 = \{4,5,7\}, \ A_1 \backslash A_4 = \{5,7\}, \ A_2 \backslash A_4 = \{4,7\}, \\ &A_3 \backslash A_4 = \{5,7\}, \quad A_5 \backslash A_4 = \{4,7\}, \quad A_6 \backslash A_4 = \{4,5\}, \\ &A_7 \backslash A_4 = \{4,5\}. \\ &C_5 = X \backslash A_5 = \{1,5,6\}, \ A_1 \backslash A_5 = \{5,6\}, \ A_2 \backslash A_5 = \{1,6\}, \\ &A_3 \backslash A_5 = \{1,5\}, \quad A_4 \backslash A_5 = \{1,6\}, \quad A_6 \backslash A_5 = \{1,5\}, \\ &A_7 \backslash A_5 = \{5,6\}. \\ &C_6 = X \backslash A_6 = \{2,6,7\}, \ A_1 \backslash A_6 = \{6,7\}, \ A_2 \backslash A_6 = \{6,7\}, \\ &A_3 \backslash A_6 = \{2,7\}, \quad A_4 \backslash A_6 = \{2,6\}, \quad A_5 \backslash A_6 = \{2,7\}, \\ &A_7 \backslash A_6 = \{2,6\}. \\ &C_7 = X \backslash A_7 = \{1,3,7\}, \ A_1 \backslash A_7 = \{3,7\}, \ A_2 \backslash A_7 = \{1,7\}, \\ &A_3 \backslash A_7 = \{1,7\}, \quad A_4 \backslash A_7 = \{1,3\}, \quad A_5 \backslash A_7 = \{3,7\}, \\ &A_6 \backslash A_7 = \{1,3\}. \end{aligned} \tag{4}$$

In this study, we build the residual design by symmetric BIBD with parameters $(q^2 + q + 1,\ q^2, q^2 - q)$. Consider the *i*th class of the residual design that is created by the select block A_i as a fixed block, therefore, the element set of each class builds a BIBD with parameters $(v, b, r, k, \lambda) = (q + 1, q^2 + q, q^2, q, q^2 - q)$. In this paper, the focus is on a residual design that runs for q as a prime power. The $v \times b$ incidence matrix, named *M*, may define a residual. In this matrix, rows represent the x_i points and columns represent the A_j blocks. Subsequently, matrix *M* can be defined as:

$$M = \delta_{ij} = \begin{cases} 1 & if\ x_i \in A_j \\ 0 & otherwise \end{cases} \tag{5}$$

3.2.4 properties

- **Maximum network size that is supported in residual design is $N = (q^2 + q + 1)(q^2 + q)$:** Since the number of classes in residual design is exactly $q^2 + q + 1$, and each class forms $(q + 1, q^2 + q, q^2, q, q^2 - q) - BIBD$, therefore, we can have in total $(q^2 + q + 1)(q^2 + q)$ blocks for support nodes.
- **Any two classes have only one common element:** whereas the element set for each class C_i is the

same corresponding with the block A_i in the projective plane. Therefore, as defined for the projective plan, every two blocks have exactly one common element.

- **Each element in the residual design is included in exactly $q^2(q+1)$ block:** Due to the point set each class in residual design with parameters $(q+1, q^2+q, q^2, q, q^2-q)$, each element is included in $q+1$ classes, and each element is repeated in q^2 block in each class; therefore, each element is included in $q^2(q+1)$ blocks.

Block designs are precisely relevant to key pre-distribution schemes. Key rings are assigned to devices in a KPS system proposed for an IoT. We suppose a key matches with a point, and a key ring matches with a block. For example, a residual design based on KPS creates $(q^2+q+1)(q^2+q)$ key rings from a key pool with q^2+q+1 keys. In IoT, if two key rings have at least one shared key, the corresponding two devices can be directly and securely connected since they have at least one common key.

4. The Proposed Approach

4.1. mapping from residual design to key pre-distribution in IoT

In our scheme, we consider an IoT of N nodes (devices), where each node is assigned a key ring from a key pool. As already mentioned, we build a residual design using a form of symmetric BIBD, including parameters $(q^2+q+1, q+1, 1)$ where q is a prime number. We proposed a basic mapping in which a distinct key matches a residual point. the key ring also corresponds to each block, and the key pool matches the global set of points. Then, we can create $N = (q^2+q+1)(q^2+q)$ key rings from a key pool with $|X| = q^2+q+1$ keys. The size of each key ring is $k = q$ keys. We select q as a prime number in such a way that $(q^2+q+1)(q^2+q) \geq N$.

This residual design contained q^2+q+1 classes where each class has q^2 elements. In total, it creates $(q^2+q+1)(q^2+q)$ blocks of size q. Then constructed blocks as key-rings are assigned to N devices.

We have indicated basic mapping from the residual design to key pre-distribution in Table 1. First, we create the residual blocks according to key-rings. Then, we allocate a distinct key ring to each node in which each key ring has a key identifier. After the assignment, every two adjacent devices exchange their key identifiers to determine a shared key. In our approach, every two devices share at most one common key. According to residual features, every two points are included together in exactly one block, which results in the fact that two blocks cannot have more than one common point. Therefore, if two adjacent devices have one common key, the key is selected as a pairwise key that is further used to create secure communication. Otherwise, devices are required to determine secure paths, including some secure communications.

Table 1 Mapping from residual design to key pre-distribution

Residual Design	Key pre-distribution
Point Set (S)	Key-Pool (P)
Blocks	Key-Rings
Object Set Size ($\|X\| = v = q^2+q+1$)	Key-Pool Size $\|P\|$
Number of Blocks $b = (q^2+q+1)(q^2+q)$	Number of Key-Rings (N)
Size of a Block $k = q$	Size of a Key-Ring (K)
Number of Blocks that an Object is in $r = q^2$	Number of Key-Rings that a Key is in
Two Blocks share $\lambda = q^2 - q$ Objects	Two Key-Rings share λ Keys

To create an IoT with N devices, we require N key-rings; therefore, a residual design with $b = N$ blocks and set X with $|X| = v$ points need to be constructed. Hence, with prime number q, we have $v = q^2+q+1$ and $b = (q^2+q+1)(q^2+q)$. Each point in X can be related to a distinct random key, and each block can be associated with a key ring. Residual design guarantees that every two blocks have λ points in common; each key-rings (or device) has λ common keys. Table 2 indicates notations that are used in the remainder of the paper. The key pre-distribution approach proposed in this paper for an IoT of size N can be explained briefly in Algorithm 1.

Table 2 List of used notations.

Notation	Definition
N	Total number of nodes in the IoT
N_{CRD}	Number of supported nodes in CRD
l	The key size
k	Key-ring size & Block size of a given design
q	The design order (a prime number)
C_i	*i-th class of residual design*
B_{ij}	*j-th block in class i*
P_{CRD}	The probability that two nodes can establish a secure link
$P(L\|C_x)$	The network resiliency when x nodes are captured

The most important advantage of our approach is improving the probability of a shared key. As explained in the next section, our approach allows us to obtain highly secure connectivity coverage and network scalability since a block with $k = q$ disjoint keys is assigned to each device. Also, this solution provides good network resiliency due to the pairwise secret keys, which augment secure communications. Moreover, this approach illustrates that our solution can achieve higher network scalability than the existing solutions.

5. Analyses of the proposed scheme

5.1. Theoretical analysis

This section analyzes the proposed scheme, considering four important metrics: network connectivity,

memory overhead, network scalability, and *resilience* against node capture attacks.

1) ***Connectivity*:** Connectivity is the probability of every two nodes sharing at least one common key. We assume that B_{ij} (block j in class i) and $B_{i'j'}$ they are two blocks of the residual design. These two blocks are either in the same class or in different classes. We continue to consider the probability of a shared key in both cases as follows:
   i. **Same class:** any two blocks are included in the same class ($i = i'$, e.g. C_i). In this case, the probability that every two blocks from the same class have at least a common key is 1.

Proposition 1. The probability Q_{SC} that each pair of blocks is in the same class is calculated as follows:

$$Q_{SC} = \frac{\binom{q^2+q}{2}}{\binom{(q^2+q)(q^2+q+1)}{2}} \quad (6)$$

Proof. We have $q^2 + q + 1$ classes and each class has $q^2 + q$ blocks. □

   ii. **Different Classes:** Each of the two blocks is included in different classes ($i \neq i'$, e.g. C_i, $C_{i'}$). In this case, the probability of a shared key of block B_{ij} is checked with blocks of two categories of classes:
   1. Classes whose point set includes exactly one point of the block B_{ij}.
   2. Classes that do not contain any points in the block B_{ij}.

In case 1: Each block in class C_i has exactly one common point with point set q^2 other classes. Therefore, the probability of selecting one of these classes is $\frac{q^2}{q^2+q}$.

Given the definition of the residual design, in each of these classes, there are q^2 blocks that contain one common key with block B_{ij} from the class C_i. That is, the probability of choosing a block with a shared key in each of these classes is $\frac{q^2}{q^2+q}$.

Proposition 2: in case of 1**,** let's assume that B_{ij} is a block of class C_i, and block $B_{i'j'}$ is of another class, then we can calculate the probability of B_{ij} and $B_{i'j'}$ having at least one common key using the following relation:

$$P_{dcA} = \frac{q^2}{q^2+q} \times \frac{q^2}{q^2+q} \quad (7)$$

Algorithm 1. Mapping from Residual Design based on Block Complementation to key pre-distribution in IoT

Require: N {Total number of devices}

1. Find the minimum prime number q such that $(q^2 + q + 1)(q^2 + q) \geq N$.
2. Generate the projective plane of order q (Symmetric Design) with parameters $(q^2 + q + 1, q + 1, 1)$.
   - $v = q^2 + q + 1$ element $X = \{x_1, x_2, \dots, x_v\}$.
   - $b = v$ blocks $A = \{A_1, A_2, \dots, A_v\}$ of size $q + 1$.
3. Build block complementation, based on the projective plane, with parameters $(q^2 + q + 1,\ q^2, q^2 - q)$.
   - $v = q^2 + q + 1$ element $X = \{x_1, x_2, \dots, x_v\}$.
   - $b = v$ blocks $A' = \{A'_1, A'_2, \dots, A'_v\}$ of size q^2.
4. Create a residual design based on block complementation containing $q^2 + q + 1$ class, each class with parameters $(q + 1, q^2 + q, q^2, q, q^2 - q)$.
   - Blocks $A_{ij} = A'_i \backslash A'_j$: j-th block in class i.
5. Delete repeated blocks in all classes.
6. Assign blocks to specified devices.

Proof. The probability of selecting one class in case 1 is $\frac{q^2}{q^2+q}$. Also, by definition of the residual design, the probability of choosing a block with a common key in case 1 is $\frac{q^2}{q^2+q}$. □

In case 2: in this case, the probability that two blocks in different classes have at least one common key is zero.

Proposition 3: the probability Q_{DC} that different classes own every two blocks can be calculated as follows:

$$Q_{DC} = \frac{\binom{q^2+q}{1}\binom{q^2+q}{1}}{\binom{(q^2+q)(q^2+q+1)}{2}} \quad (8)$$

Proof. We have $q^2 + q + 1$ classes and each class has $q^2 + q$ blocks. Therefore, we select two blocks in two different classes. □

Proposition 4 The probability that each pair of blocks has one or more common keys in residual design is calculated as follows:

$$P_{CRD} = 1 \times Q_{SC} + \left(\frac{q^2}{q^2+q} \times \frac{q^2}{q^2+q}\right) \times Q_{DC} \quad (9)$$

Proof it follows from Proposition 1 to 3. □

2) ***Memory overhead*:** In the key pre-distribution

scheme, each node needs the memory to store keys. When using the proposed residual design of order q, one key ring was assigned to each node. Therefore, there were q disjoint keys in each node. Also, to store keys in each node, the memory required was $l \times q$, where the key size was denoted by l.

3) ***Network scalability*:** network scalability means the maximum number of nodes a network can support. This number is similar to the key rings that design can support. Hence, based on our approach, we can use the relation $N = (q^2 + q + 1)(q^2 + q)$ calculate the total number of possible key rings when the residual design is used. Since each block of residual design is repeated q times in each class, and disjoint blocks from N possible blocks of residual design are assigned to each node; the maximum number of nodes that we can support is equal to:

$$N_{CRD} = \frac{(q^2 + q + 1)(q^2 + q)}{q} = (q^2 + q + 1)(q + 1) \qquad (10)$$

This section will calculate the maximum network size based on the compromised nodes to keep the network secure.

4) **Resilience:** Network resiliency is defined as the fractions of secure external links that are uncompromised when x sensor nodes are captured. In terms of resilience, we are interested in solving the probability $P(L|C_x)$ that is calculated as:

$$P(L|C_x) = \sum_{\forall j} P(l_j|l)P(D_j|C_x) \qquad (11)$$

Equation 11 shows the probability of a link L being compromised when x randomly selected nodes and their related key-rings are captured by an adversary. In Equation 11, C_x is the number of times that x nodes are captured, l_j is the number of times that a given link is secured with key j, and finally, D_j is the event that a key ring, including key j is compromised.

First, we assume two nodes, v, and u are not captured. If x node is attacked and decrypted, the probability of an attacker's decrypting the communication between v and u can be calculated using Equation 11. The probability that communication is secured with key j can be calculated as:

$$P(l_j|l) = \frac{\binom{q^2(q+1)}{2}}{\binom{(q^2+q+1)(q+1)}{2}} \qquad (12)$$

Also, the probability that D_j key-ring includes key j is compromised, with x captured nodes computed as:

$$P(D_j|C_x) = 1 - \frac{\binom{(q^2+q+1)(q+1)-q^2(q+1)}{x}}{\binom{(q^2+q+1)(q+1)}{x}} \qquad (13)$$

Finally, the probability that a link is compromised when x nodes are captured by an adversary can be computed using the following relation:

$$P(L|C_x) = \sum_{j-1}^{q^2+q+1} \frac{\binom{q^2(q+1)}{2}}{\binom{(q^2+q+1)(q+1)}{2}} \left(1 - \frac{\binom{(q^2+q+1)(q+1)-q^2(q+1)}{x}}{\binom{(q^2+q+1)(q+1)}{x}}\right) \qquad (14)$$

In our approach, the resilience against node capture is a significant parameter. An attacker may attack our proposed idea in two ways. First, the attacker agrees with the link key between nodes without capturing them. Second, sensor nodes may be captured by the attacker to prevent creating pairwise keys. Therefore, our main metrics of interest include the fraction of compromised secure links between pairs of uncompromised nodes and the fraction of compromised keys.

6. Implementation

The Contiki Operating System was used to develop the simulation. Contiki is created to run on hardware devices that are severely limited in terms of memory, power, processing power, and communication bandwidth. There is a network simulator called Cooja in every Contiki system, which simulates network nodes. Contiki is designed for small-scale systems. It has merely a few *kilobytes* of memory available. The recently standardized IETF protocols for low-power IPv6 networking, including the 6LoWPAN adaptation layer, the RPL IPv6 multi-hop routing protocol, and the CoAP RESTful application-layer protocol, are supported by this system.

Eschenauer and Gligor proposed A key pre-distribution algorithm [40]. In the context of RPL, the performance was examined using a simulation experiment. The experiment explicitly explores the percentage of leaves sharing a key in the RPL routing table.

We compared essential metrics of the proposed scheme (CRD) against other methods such as SBIBD, Combinatorial Trade, and Residual Design (RD) with a similar approach (combinatorial design) in a key establishment.

6.1. Performance Comparison

In this section, the CRD approach proposed is compared with the existing schemes in terms of different criteria. The parameters of different existing schemes, such as symmetric BIBD, combinatorial trade and residual design (RD) are summarized in table 3.

Table 3 parameters of SBIBD, Trade, RD, CRD

Design	v	b	r	k	λ
SBIBD	q^2+q+1	q^2+q+1	$q+1$	$q+1$	1
Trade	q^2+q+1	$2(q^2+q+1)$	$2(q+1)$	$q+1$	1
RD	q^2+q+1	$(q^2+q+1)(q+1)$	$q(q+1)$	q	1
CRD	$q+1$	$(q^2+q+1)(q+1)$	q^2	q	q^2-q

Combinatorial trade: In each $t-(v,k)$ trade (also known as the combinatorial trade), there are $T = \{T_1, T_2\}$ collections where T_i $(i = 1,2)$ is a collection of m blocks with the size of k (*k-subsets*) that are selected from X that the T_1 blocks are different from the T_2 $(T_1 \cap T_2 = \emptyset)$ blocks. Also, each *t-set* that is selected from X happens in the same number of blocks of T_1 similar to those of T_2. Therefore, upon noticing a $t-(v,k)$ Steiner trade of volume m, all the k-subsets of

$T_1 \cup T_2$ as blocks of the design can be considered. It should be noted that any *t*-subset of elements happens in either 2 or no blocks. As soon as it is mapped onto the key pre-distribution, v is the size of the key pool, and T_1 and T_2 are the sensors holding k keys [41].
The key pre-distribution scheme will have a key pool size of $q^2 + q + 1$, provided that q is a prime power, the maximum number of nodes in the network is $2(q^2 + q + 1)$, and the key-ring size is $q + 1$.

1) *Scalability*

We compared the scalability of our proposed scheme with three existing methods, namely SBIBD [32], Trade [41] and RD [26] in Fig 3. As can be seen from the figure, considering a similar key-ring size, the scheme proposed here leads to a significant increase in scalability compared to the two methods, namely SBIBD and Trade. However, its scalability is the same as the RD method. Therefore, simulation results suggest that considering similar network sizes, using RD and CRD schemes reduces the key-ring size compared to the other schemes.
In our proposed scheme, an equal number of key rings and devices that can be supported by design was used. For instance, in our scheme and the RD scheme, in case a network requires 2500 devices, the smallest prime number that satisfies this requirement is $q = 13$, which results in 2562 nodes. However, in SBIBD and Trade, to support this network with 2500 devices, the smallest prime number must be $q = 53$ and $q = 37$, respectively.

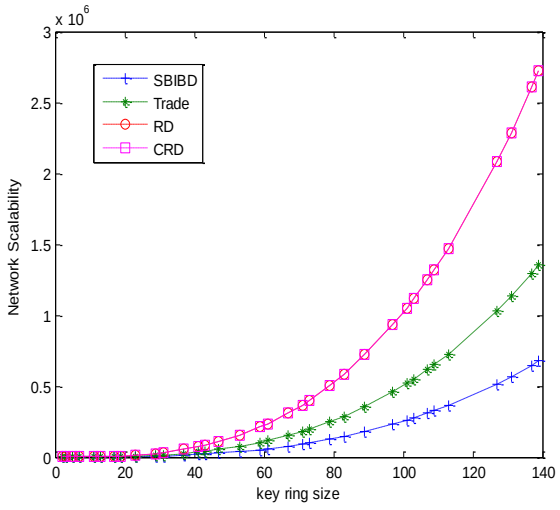

Fig. 3. A comparison of various schemes regarding Scalability

2) *Connectivity*

Two neighboring nodes can have at least one shared key to communicate with each other directly on an IoT device. Fig. 4 compares the probability of a shared key between nodes for four specified methods. It can be seen from the figure that the SBIBD method has a perfect probability of key sharing. The reason is that every two nodes in this method have at least one shared key. Results suggest that our proposed scheme improves connectivity compared to the RD and the Trade methods.

3) *Resilience*

With a similar approach, the resilience of our proposed scheme was compared against node capture attacks using the SBIBD, Trade and RD methods.
In Fig 5, all four methods are compared at an equal number of compromised nodes for a key-ring size of $k = 24$ and $k = 42$. To calculate the resilience of three methods SBIBD, Trade, and RD, we used [32], [41] and [26], respectively. It was found that our scheme, compared to the three other methods for a compromised node number (CNN) larger than 25, provides good resilience considering the same key-ring size $k = 24$. Also, the CRD resilience is compared with that of other schemes considering the same key-ring size $k = 42$. For compromised nodes numbers bigger than 47, CRD has an advantage in terms of resilience.

6.2. Discussion

In Tables 4 and 5, we illustrated the numerical results comparing scalability, connectivity, and resilience of the four schemes, namely SBIBD, Trade, RD, and CRD, considering similar key-ring sizes. The proposed scheme provides the maximum number of supported nodes for network scalability. For example, if the key-ring size were equal to $k = 90$, the CRD method would generate nodes more than 90 times the SBIBD and more than 45 times the Trade. Also, numerical results show that the scheme proposed in this paper is better than the other three schemes regarding network resilience. For example, considering key-ring size $k = 42$ and CNN=85, the CRD resilience=0.773, SBIBD=0.835, Trade=0.872, and RD=0.783. Also, our scheme increases the probability of shared key compared to the two methods, namely Trade and RD.

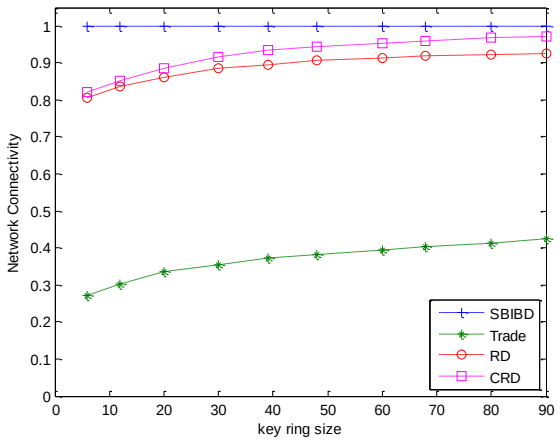

Fig. 4. Connectivity comparison our scheme with three existing methods. SBIBD has perfect connectivity and our scheme is better than Trade and RD.

Table 4 Simulation results of different schemes in terms of connectivity and scalability

Method	key-ring size=20		key-ring size=30		key-ring size=48		key-ring size=68		key-ring size=90	
	Number of nodes	P_c	Number of nodes	P_c	Number of nodes	P_c	Number of nodes	P_c	Number of nodes	P_c
SBIBD	381	1	871	1	2257	1	4557	1	8011	1
Trade	762	0.335	1742	0.356	4514	0.381	9114	0.403	16022	0.425
RD	7620	0.860	26130	0.886	108336	0.908	309876	0.920	720990	0.927
CRD	7620	0.885	26130	0.916	108336	0.945	309876	0.960	720990	0.971

Table 5 Simulation results of different schemes in terms of resilience

KRS-CNN	10	25	40	55	70	85	100
SBIBD							
24	0.331	0.532	0.658	0.764	0.856	0.913	0.945
42	0.254	0.432	0.557	0.660	0.758	0.835	0.877
Trade							
24	0.275	0.523	0.657	0.772	0.884	0.947	0.975
42	0.176	0.374	0.520	0.671	0.783	0.872	0.913
RD							
24	0.346	0.521	0.637	0.734	0.812	0.885	0.910
42	0.241	0.426	0.533	0.630	0.715	0.783	0.817
CRD							
24	0.353	0.523	0.625	0.728	0.802	0.870	0.896
42	0.263	0.423	0.527	0.625	0.704	0.773	0.794

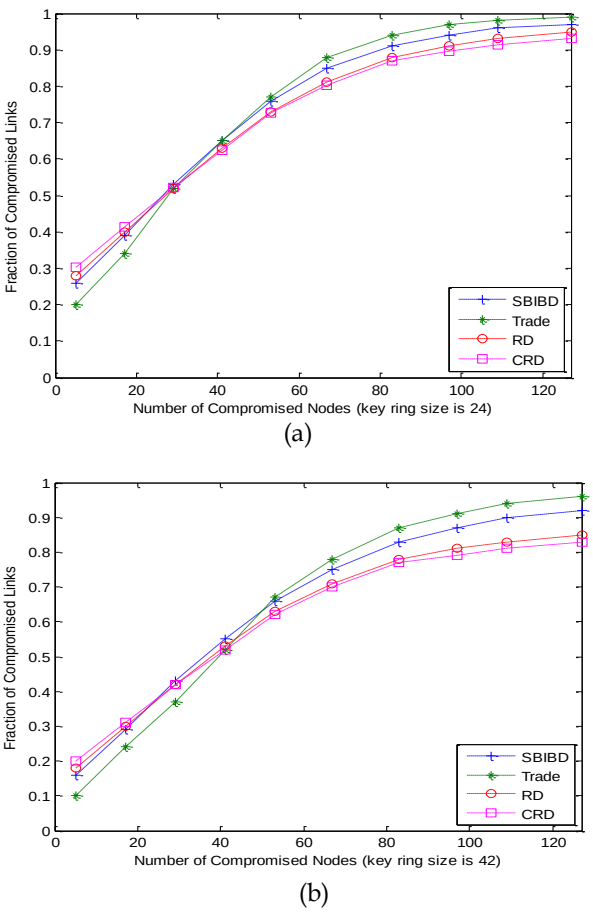

Fig. 5. Resilience Comparison. (a) Resilience of our proposed is compared with SBIBD, Trade and RD with same key-ring size k=24. (b) Resilience of our proposed is compared with other existing methods with same key-ring size k=42.

7. Conclusion

This paper proposed a developing and highly scalable key pre-distribution scheme for an IoT device. The block complementation theory was used to build a residual design for the first time. We showed that a mapping from residual design to key pre-distribution is needed to achieve profoundly high network scalability while at the same time degrading the key sharing probability. In Figs 3 to 5, we demonstrated comparison results of scalability, connectivity, and resilience of four methods (SBIBD, Trade, RD, and CRD) considering similar key-ring sizes. The maximum network scalability was obtained using the RD and CRD methods. Also, the proposed scheme is better than the other three schemes regarding network resilience. Our proposed scheme increases the probability of network connectivity more than the two methods, Trade and RD, but its connectivity is less than that of the SBIBD method.

8. References

[1] T. Gomes, F. Salgado, S. Pinto, J. Cabral and A. Tavares, "A 6LoWPAN Accelerator for Internet of Things Endpoint Devices," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 371-377, 2018, **DOI:** 10.1109/JIOT.2017.2785659.

[2] M. Husamuddin and M. Qayyum, "Internet of Things :A Study on Security and Privacy Threats," in *2nd International Conference on Anti-Cyber Crimes (ICACC)*, Abha, 2017,

DOI: 10.1109/Anti-Cybercrime.2017.7905270.

[3] M. Saadeh, Z. Sleit, M. Qatawneh and W. Almobaideen, "Authentication Techniques for the Internet of Things: A Survey," in *Cybersecurity and Cyberforensics Conference*, Amman, Jordan, 2016, **DOI:** 10.1109/CCC.2016.22.

[4] M. Malik, M. Dutta and J. Granjal, "A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things," *IEEE Access,* vol. 7, pp. 27443 - 27464, 2019, **DOI:** 10.1109/ACCESS.2019.2900957.

[5] M. Anzani, H. Haj Seyyed Javadi and V. Modiri, "Key-management scheme for wireless sensor networks based on merging blocks of symmetric design," *Wireless Networks,* vol. 24, no. 8, p. 2867–2879, 2017, DOI:10.1007/s11276-017-1509-y.

[6] M. Tajeri, H. H. S. Javadi, M. Bayat and M. E. Shiri, "Pre-Distribution Encryption Key Scheme for Communicating between IoT Device Layer and Fog Layer," *Cybernetics and Systems,* pp. 1-25, 2022, DOI:10.1080/01969722.2022.2145665.

[7] T. Dargahi, H. H. S. Javadi and M. Hosseinzade, "Application-specific hybrid symmetric design of key pre-distribution for wireless sensor networks," *Security and Communication Networks,* vol. 8, no. 8, pp. 1561-1574, May 2015, DOI:10.1002/sec.1104.

[8] A. Pattanayak and B. Majhi, "Key pre-distribution schemes in distributed wireless sensor network using combinatorial designs revistied," 2009.

[9] S. Akhbarifar, H. Haj Seyyed Javadi, A. M. Rahmani and M. Hosseinzadeh, "Hybrid Key Pre-distribution Scheme Based on Symmetric Design," *Iranian Journal of Science and Technology, Transactions A: Science,* vol. 28, no. 39, p. 1–8, 2019, DOI:10.1007/s40995-019-00703-7.

[10] H. Haj Seyyed Javadi and M. Anzani, "Hybrid Key Pre-distribution Scheme for Wireless Sensor Network Based on Combinatorial Design," *Journal of Advances in Computer Engineering and Technology,* vol. 1, no. 3, pp. 33-38, 2015, DOI:10.13140/RG.2.2.13619.63520.

[11] S. H. Erfani, H. Haj Seyyed Javadi and A. M. Rahmani, "Analysis of Key Management Schemes in Dynamic Wireless Sensor Networks," *Advances in Computer Science: an International Journal,* vol. 4, no. 1, pp. 117-121, 2015.

[12] S. H. Erfani, H. Haj Seyyed Javadi and A. M. Rahmani, "A dynamic key management scheme for dynamic wireless sensor networks," *Security and Communication Networks,* vol. 8, no. 6, pp. 1040-1049, 2014, DOI:10.1002/sec.1058

[13] D. Chen, G. Chang, D. Sun, J. Jia and X. Wang, "Lightweight key management scheme to enhance the security of internet of things," *International Journal of Wireless and Mobile Computing,* vol. 5, no. 2, pp. 191-198, 2012, DOI:10.1504/IJWMC.2012.046773.

[14] H. Chan, A. Perrig and D. Song, "Random Key pre-distribution Schemes for Sensore Networks," in *Symposium on Security and Privacy*, Berkeley, CA, USA, 2003, **DOI:** 10.1109/SECPRI.2003.1199337.

[15] S. Qian, "A Novel Key Pre-distribution for Wireless Sensor Networks," in *International Conference on Solid State Devices and Materials Science*, 2012, DOI: 10.1016/j.phpro.2012.03.368.

[16] N. Solari Esfehani and H. Haj Seyyed Javadi, "A survey of key pre-distribution schemes based on combinatorial designs for resource-constrained devices in the IoT network," *Wireless Networks,* vol. 27, no. 4, pp. 3025-3052, 2021, DOI: 10.1007/s11276-021-02629-8.

[17] A. Morshed Aski and H. Haj Seyyed Javadi, "A novel key pre-distribution scheme based on -PBIBD combinatorial design in the resource-constrained IoT network," *arXiv preprint arXiv:2102.07137,* 2021, DOI:10.48550/arXiv.2102.07137.

[18] M.-L. Messai, "A Self-Healing Pairwise Key Pre-Distribution Scheme in IoT-based WSNs," in *International Wireless Communications and Mobile Computing (IWCMC)*, Marrakesh, Morocco, 2023, **DOI:** 10.1109/IWCMC58020.2023.10183198.

[19] B. Vinayaga Sundaram, M. Ramnath, M. Prasanth and J. Varsha Sundaram, "Encryption and Hash based Security in Internet of Things," in *3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, Chennai, 2015, **DOI:** 10.1109/ICSCN.2015.7219926.

[20] V. Chegeni, H. Haj Seyyed Javadi, M. R. Moazami Goudarzi and A. Rezakhani, "A scalable key pre-distribution scheme based on the unital design for the internet of things security," *IETE Journal of Research,* pp. 1-12, 2021, DOI: 10.1080/03772063.2021.1933626.

[21] F. K. Santoso and N. C. H. Vun, "Securing IoT for smart home system," in *International Symposium on Consumer Electronics (ISCE)*, Madrid, 2015, **DOI:** 10.1109/ISCE.2015.7177843.

[22] N. Sklavos and I. D. Zaharakis, "Cryptography and Security in Internet of Things (IoTs) : Models, Schemes and Implementations," in *8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Larnaca, 2016, **DOI:** 10.1109/NTMS.2016.7792443.

[23] M. Anzani, H. Haj Seyyed Javadi and A. Moeni, "A deterministic Key Predistribution Method for Wireless Sensor Networks Based on Hypercube Multivariate Scheme," *Iranian Journal of Science and Technology, Transactions A: Science,* vol. 42, no. 2, p. 777–786, June 2018, DOI:10.1007/s40995-016-0054-3.

[24] Y. Kim, E. Lim and T. Kwon, "On the Impact of Deployment Errors in Location-Based Key Pre-distribution Protocols for Wireless Sensor Networks," *IEEE Access,* vol. 12, pp. 35765 - 35778, 2024, **DOI:** 10.1109/ACCESS.2024.3372653.

[25] M. Javanbakht, H. Erfani, H. Haj Seyyed Javadi and P. Daneshjoo, "Key Pre-distribution Scheme for Clustered Hierarchical Wireless Sensor Networks based on Combinatorial Designs," *Security and Communication Networks,* vol. 7, no. 11, pp. 2003-2014, November 2014, DOI: 10.1002/sec.914.

[26] V. Modiri, H. Haj Seyyed Javadi and M. Anzani, "A Novel Scalable Key Pre-distribution Scheme for Wireless Sensor Networks Based on Residual Design," *Wireless Personal Communications,* vol. 96, no. 2, p. 2821–2841, September 2017, DOI:10.1007/s11277-017-4326-9.

[27] V. Modiri, H. Haj Seyyed Javadi and M. Anzani,

"Using Residual Design for Key Management in Hierarchical Wireless Sensor Networks," *Journal of Information Systems and Telecommunication (JIST),* vol. 8, no. 1, pp. 53-61, 2020, https://civilica.com/doc/1352392/.

[28] A. Kumar, N. Bansal and A. R. Pais, "New key pre-distribution scheme based on combinatorial design for wireless sensor networks," *IET Communications,* vol. 13, no. 7, p. 892 – 897, 2019, DOI:10.1049/iet-com.2018.5258.

[29] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks," *Security and Communication Networks,* vol. 7, pp. 2495-2508, 2014, DOI: 10.1002/sec.354.

[30] G. Gaubatz, J.-P. Kaps, E. Ozturk and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, Kauai Island, 2005, **DOI:** 10.1109/PERCOMW.2005.76.

[31] S. Raza, H. Shafagh, K. Hewage, R. Hummen and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," *IEEE Sensors Journal,* vol. 13, no. 10, pp. 3711-3720, 2013, **DOI:** 10.1109/JSEN.2013.2277656.

[32] S. A. Camtepe and B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking,* vol. 15, no. 2, pp. 346 - 358, April 2007, **DOI:** 10.1109/TNET.2007.892879.

[33] S. Ruj and B. Roy, "Key pre-distribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks," *ACM Trans. Sensor Netw,* vol. 6, no. 4, pp. 1-4, Jan 2010, DOI:10.1145/1653760.1653764.

[34] A. Morshed Aski, H. Haj Seyyed Javadi and G. H. Shirdel, "A Full Connectable and High Scalable Key Pre-distribution Scheme Based on Combinatorial Designs for Resource-Constrained Devices in IoT Network," *Wireless Personal Communications,* vol. 114, no. 3, 2020, DOI: 10.1007/s11277-020-07466-0.

[35] A. Dey, Theory of block designs, J. Wiley, 1986.

[36] C. J. Colbourn and J. H. Dinitz, Handbook of combinatorial designs, CRC press, 2010.

[37] D. R. Stinson, Combinatorial Designs: Constructions and Analysis, New York: Springer, 2004, DOI: 10.1145/1466390.1466393.

[38] H. R. Sharifi, H. Haj Seyyed Javadi, A. Moeini and M. Hosseinzadeh, "Residual design of sink localization algorithms for wireless sensor networks," *Journal of High Speed Networks,* vol. 25, no. 1, pp. 87-99, 2019, DOI:10.3233/JHS-190605.

[39] P. Nikkhah Bahrami, H. Haj Seyyed Javadi, T. Dargahi, A. Dehghantanha and K. K. Raymond , "A Hierarchical Key Pre-Distribution Scheme for Fog Networks," *Concurrency and Computation: Practice and Experience,* 2018, DOI: 10.1002/cpe.4776.

[40] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA, 2002, DOI:10.1145/586110.586117.

[41] S. Ruj, A. Nayak and I. Stojmenovic, "Pairwise and Triple Key Distribution in Wireless Sensor Networks with Applications," *IEEE Transactions on Computers,* vol. 62, no. 11, pp. 2224 - 2237, 2013, **DOI:** 10.1109/TC.2012.138.