A Deterministic Multiple Key Space Scheme for Wireless Sensor Networks via Combinatorial Designs

Ibrahim Qasemzadeh Kolagar*, Hamid Haj Seyyed Javadi* and Shahriar Bijani*
*Department of Mathematics and Computer Science, Shahed University, Tehran, Iran
{i.qasemzadeh,h.s.javadi,bijani}@shahed.ac.ir
Corresponding author: Hamid Haj Seyyed Javadi

***Abstract*- The establishing of a pairwise key between two nodes for encryption in a wireless sensor network is a challenging issue. To do this, we propose a new deterministic key pre-distribution scheme which has modified the multiple key space scheme (MKSS). In the MKSS, the authors define two random parameters to make better resilience than existing schemes. Instead of a random selection of these parameters, our proposal provides a combinatorial framework by substituting appropriate parameters to satisfy certain properties. We show that the proposed scheme enhances storage memory and improves performance or security by carefully choosing a combinatorial design. In addition, we propose a new key agreement mechanism by using *derivative operation* on univariate polynomials to aim a desired computational overhead. In this case, the computational overhead of our approach lower than the general case of the MKSS's. If we choose the *primitive element* in our general formula as the special case of the MKSS, this new process has the same computational overhead as the MKSS.**

***Index Terms*-** Combinatorial design theory, Key pre-distribution, Security, Wireless sensor networks.

I. INTRODUCTION

The concept of micro-sensing and wireless connection in wireless communications have been led many new application areas of wireless sensor networks (WSNs). A sensor network is composed of a large number of sensor nodes. These small devices have limited memory, battery power, bandwidth, transmission range, and computation power. Sensor nodes are randomly distributed in large numbers in a region and operate unattended for a long period of time [5]. Typically, sensor nodes consist of sensing, processing, power and communicating components. The WSNs collect various information (e.g. mechanical, thermal, biological, chemical) and have many applications in military operations, environment and habitat monitoring, healthcare and emergency response, etc [32].

Manuscript received 5-Jan.-2017 and revised 18-April-2017,
Accepted on 7-June-2017
P- ISSN: 2322-4088
E- ISSN: 2322-3936

Under adversarial conditions, data in sensor nodes deployed in such hostile environments need to be securely operated. Thus, the security becomes a crucial issue in the WSNs. Wireless nature of communication in WSNs, resource limitation, and lack of a fixed infrastructure are various security challenges. The security services are key management, authentication, and pairwise key establishment [5]. The cryptographic techniques enable the sensor nodes to communicate securely with each other. Traditional public key protocols are infeasible in WSNs due to the resource constraints of sensor nodes. One of the important problems in WSNs is the key agreement mechanism, i.e. how to set up secret keys between communicating nodes.

Three key agreement approaches are *arbitrated keying*, *self-enforcing*, and *key pre-distribution* schemes to establish secret keys [32]. Arbitrated keying schemes which depend on infrastructures using trusted third parties are impractical. Self-enforcing approaches in WSNs which use asymmetric cryptography are inappropriate due to the limited computational power and energy resources of sensor nodes. Nowadays, key pre-distribution schemes are the best practical solution for pre-distributing keys between sensor nodes. In this approach, secret keys are preloaded in each node prior to deployment [6], [24], [29], [30].

Key pre-distribution solutions can be probabilistic, deterministic, or hybrid [5]. In probabilistic approaches, a list of keys (*key chain*) is randomly selected from a key pool and distributed to a sensor node. Deterministic solutions are proposed to provide better performance. Hybrid solutions combine a deterministic core with a probabilistic approach to inherit benefits of both.

A. Main Contributions

Du et al. [13] suggest a matrix-based key pre-distribution as defined in Blom's scheme [3]. They define a multiple key space scheme (MKSS) as a pair of matrices (*private matrix*, *public matrix*). The MKSS randomly constructs ω key spaces and selected τ $(2 \leq \tau < \omega)$ key spaces for each sensor node. If two nodes have at least a common key space, they can generate at least a pairwise key. Motivated by the random selection of the two parameters τ and ω in the MKSS, we propose the combinatorial structure for these parameters. Similarly to the MKSS, our scheme defines a *public set of arrays* and ω random symmetric *private sets of arrays*. The idea is to use a *set system* by substitution of its parameters with ω and τ. Using suitable combinatorial designs as set systems, our new scheme obtains better connectivity or resilience than the MKSS's. Furthermore, our approach enables reducing storage memory.

For establishing a pairwise key in the key agreement phase, we use the *derivative operation* on univariate polynomials to achieve a desired computational overhead. The computational overhead of our scheme is $O(\lambda log \lambda)$. This provides better the computational overhead than the general case of the MKSS. Especially, similar to the MKSS, if the *primitive elements* are selected in the proposed approach, then the computational overhead in our scheme and the MKSS are equal to $O(\lambda)$.

The remaining of the paper is organized as follows. Section II explains the related work and provides the background. In Section III, we explain our system model and adversarial model. Section VI describes our proposed approach with two phases in a WSN. In Section V, we evaluate performance and security properties of our scheme and compare it with the MKSS. We also analyze the main results of our proposal in section VI. Finally, we conclude the paper in Section VII.

II. Related Work and Background

The uncontrolled environments for WSNs and the existence of various attacks motivate the designing of security protocols. Many types of research have been addressed designing security protocols in WSNs. A few examples have been discussed the surveys in [5], [8], [32], [34]. In most security protocols, all cryptographic operations involve keys. Therefore, key establishment is the first step to establishing a security infrastructure. Key pre-distribution schemes are assumed the best solution for key establishment in WSNs. There are three categories for key pre-distribution schemes: random, deterministic, and hybrid.

A. Random Key Pre-distribution Schemes

In the random key pre-distribution schemes, each node is preloaded with a list of keys from a key pool. In such sensor networks, we imagine a graph in which each node is a vertex. There exists an edge between two nodes if they have a common shared key. Such a graph is called *random graph* [17]. Based on the random graph theory, Eschenauer et al. [16] propose a random key pre-distribution (AKA *basic scheme*). In this scheme, any two neighboring use a single common key to establishing a secure link. The basic scheme is generalized by Chan et al. [7] in which two nodes share at least q common keys ($q > 1$) for computing a pairwise key. Ehdaie et al. [14] propose another random key distribution by increasing resilience for WSNs to node capture without additional computation and communication overheads rather than existing schemes. Other examples of random key pre-distribution schemes have been studied in [21], [22], [33], [35].

B. Deterministic Key Pre-distribution Schemes

Various types of deterministic key pre-distribution schemes have been proposed: matrix-based, polynomial-based, and combinatorial-based schemes.

Blom [3] proposes a threshold matrix-based key pre-distribution which generates a public matrix and a private symmetric matrix for establishing pairwise keys. To increase the resilience of Blom's scheme, Du et al. [13] use multiple private symmetric matrices, instead of just one. This scheme is called *multiple key space scheme (MKSS).*

Another class of deterministic schemes is the polynomial-based approach. It can be considered an equivalent between polynomial-based and MKSS schemes. In polynomial-based approach, a

symmetric bivariate polynomial is used for establishing a common key. The elements in the second row of a public matrix in the MKSS can be considered as the IDs of sensor nodes in polynomial-based approach. Each private matrix can be considered as the coefficients of a bivariate polynomial. For the first time, Blundo et al. [4] suggest a λ-secure polynomial-based key pre-distribution scheme. In this work, the *setup server* generates a symmetric bivariate λ-degree polynomial from which each node establishes a common key with its neighbors. Mitra et al. [25] propose another polynomial-based scheme which uses three disjoint sets of distinct symmetric bivariate polynomials in the *triangular grid*. Therefore, each node enables to communicate with all nodes lying on the same straight line. In [2], Anzani et al. apply the combinatorial design theory in the multivariate key pre-distribution scheme. In this scheme, using the IDs of sensors as an d tuple of positive integers and the combinatorial design, the shares of the multivariate polynomials store in sensor nodes before the network deployment.

In the third class of deterministic schemes, there are several key pre-distribution schemes: *projective planes, generalized quadrangles, transversal designs, and partially balanced incomplete block designs*. For example, Camtepe and Yener [6] propose novel deterministic approaches which use projective planes and generalized quadrangles are mapped to key distribution schemes. Another combinatorial-based key pre-distribution scheme based on partially balanced incomplete block designs was proposed by Ruj et al. [31]. For more examples of combinatorial-based schemes, see [1], [19].

C. Hybrid Key Pre-distribution Schemes

To enhance the desired metrics for WSNs, the hybrid schemes merge random and deterministic approaches to inherit benefits of both. Camtepe et al. [6] propose two hybrid designs: *hybrid symmetric* and *hybrid GQ* designs. These approaches combine a deterministic core with a probabilistic extension. Dargahi et al. [12] modify the hybrid symmetric design to improve key share probability and scalability yet providing the same resilience against node capture attack. They use two similar key pools with some different keys in contrast to the hybrid symmetric scheme, which utilizes one key pool and its complement. Liu et al. [23] propose another hybrid key pre-distribution scheme to improve resilience and scalability in Blundo's scheme [4]. This scheme merges polynomial-based approach and key pool idea in the basic scheme [16]. A few examples of hybrid schemes are [18], [20].

D. Design-theoretic background

Definition 1. *A set system is a pair* (X, A)*, where* 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* (X, A) *is the size of the largest block.*

Definition 2. *A balanced incomplete block design (BIBD) or* (v, b, r, k, μ)*-BIBD is a set system with the following properties [9]:*

1. $|X| = v, |A| = b$,

2. Each block of A contains exactly k *elements,*

3. Each element occurs in exactly r *blocks,*

4. Each pair of elements comes in exactly μ *blocks of A.*

In a (v, b, r, k, μ)-BIBD, we have: $\mu(v-1) = r(k-1)$ and $bk = vr$. A BIBD is called symmetric design or symmetric BIBD denoted by (v, k, μ)-SBIBD when $b = v$ and therefore $r = k$ [9].

Definition 3. *An* $(n^2 + n + 1, n + 1, 1)$*-SBIBD with* $n \geq 2$ *is called a projective plane of order n.*

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

Definition 5. *Let* $n \geq 2$. *An* $(n^2, n^2 + n, n + 1, n, 1)$*-BIBD is called an affine plane of order n.*

Definition 6. *For every prime power* $q \geq 2$*, there exists a* $(q^2, q, 1)$*-SBIBD (i.e., an affine plane of order q).*

Definition 7. *An* (s, t)*-generalized quadrangle is a pair* (X, A) *in which X and A are disjoint (nonempty) sets of objects called points and lines (respectively), which satisfies the following axioms:*

(i) Each point is incident with exactly $t + 1$ *lines* $(t \geq 1)$.

(ii) Each line is incident with $s + 1$ *points* $(s \geq 1)$.

(iii) if x *is a point and L is a line not incident with* x*, then there is a unique point* $y \in L$ *such that* x *and* y *occur on a line.*

The pair (X, A) is denoted by $GQ(s, t)$ and the integers s and t are the parameters of the GQ. The GQ with parameters s and t is said to have order (s, t).

Definition 8. *Let* $GQ(s, t)$ *be a generalised quadrangle of order* (s, t)*, and put* $|X| = v$, $|A| = b$. *Then, in a* $GQ(s, t)$*, there are* $v = (s + 1)(st + 1)$ *points and* $b = (t + 1)(st + 1)$ *lines where each line includes* $s + 1$ *points and each point appears on* $t + 1$ *lines.*

Definition 9. *A projective space* $PG(d, q)$ *of dimension d over a field F (order q) is constructed from the vector space of dimension* $d + 1$ *over F such that objects are all subspaces of the vector space and two objects are incident if one contains the other.*

Remark 1. *There are three known GQs as defined in [28]: 1)* $GQ(q, q)$ *from projective space* $PG(4, q)$*; 2)* $GQ(q, q^2)$ *from projective space* $PG(5, q)$*; and 3)* $GQ(q^2, q^3)$ *from projective space* $PG(4, q^2)$ *with q as an arbitrary prime power.*

III. SYSTEM MODEL AND ADVERSARIAL THREAT

A. System model

We consider a WSN with N sensor nodes which randomly distributed in the field. The base station

first generates a key pool of different *arrays* and then constructs the *public* subset and the *private* subsets of the key pool. Each sensor node is preloaded with a list including the node's id and various *key spaces* which will be explained in the next section. Our scheme is divided into two phases: *setup* and *key agreement*. In the first phase, one array of the public set and one array of some private sets are preloaded in each node. After deployment, any pair of two neighboring nodes exchanges a list including the node's id, the indices of their key spaces, and the array of the public set to establish at least one common key. If two neighboring nodes do not share any common key, they can find a secure path in the network to establish a common key through other nodes. This approach will be discussed in more detail in Section IV.

B. Adversarial threat

The setup phase of our scheme is performed before the deployment of the network. Therefore, an adversary cannot recover the key pool and any subset of it. Thus, the setup phase is secure. After deployment, any pair of two nodes exchanges a list which includes the node's id, the indices of their key spaces, and the array of the public set. In a standard attack assumption (e.g. [6]), an adversary chooses a number of sensor nodes *randomly*. To obtain any information about the keys, he or she needs to capture the key spaces to access the keys stored in those nodes. Therefore, all links of captured node, which were communicated to other nodes, will be broken which do not include *specific array*. For example, suppose that N_i, N_j, N_k are three nodes, where they have a common array U. Suppose that the node N_k is compromised. Then the secrecy of the common key between N_i and N_j which is generated by the common array U is broken. In this situation, the capture of the common key corresponded with U affects the links from N_i to N_j.

In other attack assumption, we assume that a wise attacker monitors the whole network and captures a number of sensor nodes *selectively*. Then, he or she has the ability to compromise the same *specific array* in their key spaces of selected nodes and recovers the key pool.

The unattended operation of sensors in uncontrolled environments increases the various types of attacks. For example, the adversary might reveal all the keys or information stored in the nodes. Then, the adversary can use this information to eavesdrop on other links between uncompromised nodes. The adversary can obtain some secret information by appending additional hostile nodes into the sensor network. Thus, an adversary can control the entire network with clones of the captured node. This is the *clone attack* [16]. Another scenario of a particularly harmful attack on WSN is the *Sybil attack* [15], where an attacker generates multiple sensor identities and handles the Sybil identities to the sensor networks. To do this, he may attempt to capture a set of legitimate nodes and extracts the keys for communicating with the rest of the network. However, the first step to further attacks is that the adversary gains the full physical control of a sensor node and removes that node from the network in a way such that the node cannot communicate with the other nodes in the network. This attack is

TABLE I. LIST OF USED NOTATION

Notation	Definition
Σ	A finite nonempty set of elements
Σ^*	A set of all arrays with different lengths over Σ
$GF(q')$	A Galois field with a prime order q'
s	A primitive element which each nonzero element in $GF(q')$ can be represented by some power of s
N	The total number of sensor nodes in the network
P_{GQ}	The probability of establishing a common key between two nodes in the GQ design
P_{MHS}	The probability of establishing a common key between two nodes in the modified hybrid symmetric design
P_{actual}	The probability of establishing a common key between two nodes in the MKSS
L	Event that a link is compromised
C_x	Event that the adversary captures x nodes and thus x key chains
$P\{L\|C_x\}$	The probability that a link is compromised when an attacker captures x key chains

called the *node capture attack* as a serious threat in WSNs [10].

IV. OUR PROPOSED SCHEME

In this section, we have modified the MKSS and investigate various metrics such as performance, security, storage memory, and computational overhead. The used notations are summarized in Table I. Our framework for key pre-distribution is divided into two phases: *setup* and *key agreement*.

A. Setup phase

In this phase, the base station uniquely assigns identifiers and key spaces to sensor nodes. Let Σ be a finite non-empty set of elements. We set $\Sigma = GF(q') = Fq'$ where $q' = p^n$ for some prime number p. An array is defined to be a finite string of elements in Σ. We define the length of an array $u = a_1 \dots a_m$ to be $|u| = m$. Let Σ^* be a set of all arrays with different lengths over Σ. We pick a key pool $P \subseteq \Sigma^*$. In our framework, we choose a random parameter λ such that $\lambda + 1$ is the length of each array. The base station constructs a set G as a subset of key pool P with N arrays.

$$G = \left\{ \underbrace{b_{11}b_{12} \dots b_{1(\lambda+1)}}_{G(1)}, \dots, \underbrace{b_{N1}b_{N2} \dots b_{N(\lambda+1)}}_{G(N)} \right\}, \quad (1)$$

where $G(i) \neq G(j)$ for $1 \leq i, j \leq N (i \neq j)$. Similar to the MKSS, the set G is public information. An adversary and each sensor are able to know the contents of G.

An Example of Set G

We present an example of set G. Let s be a primitive element of the finite field $GF(q')$ (q' is a prime power); namely, each nonzero element in $GF(q')$ can be expressed by some power of s. Also,

each key is represented by a primitive element in $GF(q')$, where $|q'|$ is larger than the desired key size. An example of G can be designed as follows.

$$G = \left\{ \underbrace{1ss^2 \dots s^{\lambda}}_{G(1)}, \underbrace{1s^2(s^2)^2 \dots (s^2)^{\lambda}}_{G(2)}, \dots, \underbrace{1s^N(s^N)^2 \dots (s^N)^{\lambda}}_{G(N)} \right\}. \quad (2)$$

Based on the property of primitive elements, $s^i \neq s^j$ if $i \neq j$ mod q'. Therefore, $s, s^2, s^3, \dots, s^N$ are all distinct. Also, G can be generated by the primitive element s of $GF(q')$. For storing the ith array of G at node i, we only need to store the element s^i at this node.

Du et al. [13] randomly generate ω symmetric matrices $D_1, \dots, D_\omega$ of size $(\lambda+1) \times (\lambda+1)$. The tuple $S_i = (D_i, G)$, where $i = 1, \dots, \omega$, is called a *key space* in which G is a public matrix of size $(\lambda+1) \times N$. To achieve better resilience than Blom's scheme [3], they set two parameters ω and τ, where $2 \leq \tau < \omega$. They randomly pick τ distinct key spaces from the ω key spaces for each sensor node.

Instead of a random selection of the parameters τ and ω, we now use combinatorial structures for these parameters to satisfy certain properties. According to Definition (1), we obtain a set system (X, A) by substitution of the parameters $\omega = v$ and $\tau = k$. By this definition, the base station creates v random symmetric sets D_l as follows.

$$D_l = \left\{ \underbrace{a^l_{11}a^l_{12} \dots a^l_{1(\lambda+1)}}_{D^l(1)}, \dots, \underbrace{a^l_{(\lambda+1)1}a^l_{(\lambda+1)2} \dots a^l_{(\lambda+1)(\lambda+1)}}_{D^l(\lambda+1)} \middle| a^l_{ts} = a^l_{st} \in GF(q') \right\}, \quad (3)$$

where $l \in \{1, \dots, v\}$ and $1 \leq t, s \leq \lambda + 1$. For each l, the D_l is a symmetric set of arrays. Similar to the MKSS, adversaries or sensor nodes should not disclose the content of the set D_l. This set has $\lambda + 1$ arrays of length $\lambda + 1$.

Example 1. *Assume that a network with $N = 6$ nodes. An (7,3,1)-SBIBD can be used as a set system. This design generates $b = 7$ blocks out of $v = 7$ objects where block size is $k = 3$. The associated blocks of this design for sensor nodes can be $D_B = \{\{D_1, D_2, D_3\}, \{D_1, D_4, D_5\}, \{D_1, D_6, D_7\}, \{D_2, D_4, D_6\}, \{D_2, D_5, D_7\}, \{D_3, D_4, D_7\}, \{D_3, D_5, D_6\}\}$. Therefore, the base station uses 6 of 7 blocks for 6 nodes, randomly.*

Let M be a finite set of arrays. We now introduce the following map on the set M.

$$\begin{cases} \phi_i : M \to F_{q'}[x] \\ w_{ij} = a_{i1}a_{i2} \dots a_{ij} \xrightarrow{yields} \sum_{s=1}^{j} a_{is}x^{s-1} \end{cases} \quad (4)$$

where $F_{q'}[x]$ is a *polynomial ring* over a finite field $F_{q'}$ [25]. The map ϕ_i can be used on the sets G and D_l as follows.

$$\phi_i|_G = g_i(x) = \sum_{s=1}^{\lambda+1} b_{is}x^{s-1} \qquad s.t. \qquad 1 \leq i \leq N,\ b_{is} \in Fq', \quad (5)$$

where $g_i(x)$ is the corresponding polynomial to the array $G(i)$ in G. Since $G(i) \neq G(j)$, then two arbitrary outputs of the function $\phi_i|_G$ will not be equal. As a result, it must be to uniquely assign any node with the key array. Similarly,

$$\phi_j\Big|_{D_l} = f_j^l(x) = \sum_{t=1}^{\lambda+1} a_{jt}^l x^{t-1} \qquad s.t. \qquad 1 \le j \le \lambda+1,\ a_{jt}^l \in Fq^{'}, \tag{6}$$

where $f_j^l(x)$ is the corresponding polynomial to the array $D_l(j)$ in D_l.

We now construct multiplication of two polynomials $g_i(x)$ and $f_j^l(x)$. In what follows, the derivative operation is used for this multiplication to achieve a desired computational overhead. Therefore, $h_{ij}(x) = [g_i(x) \cdot f_j^l(x)]' = g_i(x) \cdot (f_j^l(x))' + g'_i(x) \cdot f_j^l(x)$ for $1 \le i \le N,\ 1 \le j \le \lambda+1$. We obtain new sets A_l by computing $h_{ij}(x)$ in $x = 1$ as follows.

$$A_l = \left\{ \underbrace{h_{11}(1)h_{12}(1) \dots h_{1(\lambda+1)}(1)}_{A^l(1)}, \dots, \underbrace{h_{N1}(1)h_{N2}(1) \dots h_{N(\lambda+1)}(1)}_{A^l(N)} \right\}, \tag{7}$$

where $1 \le l \le v$. Note that each A_l has N arrays of length $\lambda + 1$. According to Equation (4), every array in A_l corresponds with a polynomial.

Theorem 1. Let $h_i(x)$ and $g_j(x)$ be the corresponding polynomials for arrays $A^l(i)$ and $G(j)$ in A_l and G, respectively. Then,

$$\big[h_i(x) \cdot g_j(x)\big]'\Big|_{x=1} = \big[h_j(x) \cdot g_i(x)\big]'\Big|_{x=1} \qquad \forall\ 1 \le i,j \le N. \tag{8}$$

Proof. Consider two nodes m and n want to find a common pairwise key. Therefore, the base station generates the polynomials $h_m(x) = \sum_{i=1}^{\lambda+1}[g_m(1).f_i'(1) + g_m'(1).f_i(1)]x^{i-1}$ and $h_n(x) = \sum_{i=1}^{\lambda+1}[g_n(1).f_i'(1) + g_n'(1).f_i(1)]x^{i-1}$, respectively, where $f_i(x) = \sum_{t=1}^{\lambda+1} a_{it}x^{t-1}$. Suppose that $g_m(x)$ and $g_n(x)$ are their corresponding polynomials to the arrays $G(m)$ and $G(n)$ in G, respectively. We must show $[h_m(x) \cdot g_n(x)]'|_{x=1} = [h_n(x) \cdot g_m(x)]'|_{x=1}$.

$$[h_m(x) \cdot g_n(x)]'|_{x=1} = h_m(1).g_n^{'}(1) + h_m^{'}(1).g_n(1) = \underbrace{\sum_{i=1}^{\lambda+1}\sum_{t=1}^{\lambda+1} a_{it}g_m^{'}(1).g_n^{'}(1)}_{A} +$$

$$\sum_{i=1}^{\lambda+1}\sum_{t=2}^{\lambda+1}(t-1)a_{it}g_m(1).g_n^{'}(1) + \sum_{i=2}^{\lambda+1}\sum_{t=1}^{\lambda+1}(i-1)a_{it}g_m^{'}(1).g_n(1)$$

$$+ \underbrace{\sum_{i=2}^{\lambda+1}\sum_{t=2}^{\lambda+1}(i-1)(t-1)a_{it}g_m(1).g_n(1).}_{A'}$$

Similarly,

$$[h_n(x) \cdot g_m(x)]'|_{x=1} = h_n(1).g_m^{'}(1) + h_n^{'}(1).g_m(1) = \underbrace{\sum_{i=1}^{\lambda+1}\sum_{t=1}^{\lambda+1} a_{it}g_n^{'}(1).g_m^{'}(1)}_{B} +$$

$$\sum_{i=1}^{\lambda+1}\sum_{t=2}^{\lambda+1}(t-1)a_{it}g_n(1).g_m^{'}(1) + \sum_{i=2}^{\lambda+1}\sum_{t=1}^{\lambda+1}(i-1)a_{it}g_n^{'}(1).g_m(1)$$

$$+ \underbrace{\sum_{i=2}^{\lambda+1}\sum_{t=2}^{\lambda+1}(i-1)(t-1)a_{it}g_n(1).g_m(1).}_{B'}$$

In above two expressions, the summands A and A' are equal to B and B', respectively. Since $a_{it} = a_{ti}$, the other summands are equal to each other. Therefore, the proof is complete.

Example 2. Assume that $\Sigma = GF(7) = F_7$ and let $P \subseteq \Sigma^* = \bigcup_{n=0}^{\infty} F_7^n$ be a key pool and

$4 = N,\ q' = 7,\ \lambda = 2, s = 3$. According to Equation (2), we can generate a set $G = \{\underbrace{132}_{G(1)}, \underbrace{124}_{G(2)}, \underbrace{161}_{G(3)}, \underbrace{142}_{G(4)}\}$ with $N = 4$ arrays of length $\lambda + 1 = 3$.

Set $l = 1$. Next, we randomly generate a symmetric set $D = D_1 = \{\underbrace{120}_{D^1(1)}, \underbrace{245}_{D^1(2)}, \underbrace{052}_{D^1(3)}\}$ with $\lambda + 1 = 3$ arrays of length $\lambda + 1 = 3$. We now compute the polynomials in Equations (5) and (6).

$g_1(x) = 1 + 3x + 2x^2, \quad g_2(x) = 1 + 2x + 4x^2,$

$g_3(x) = 1 + 6x + x^2, \quad g_4(x) = 1 + 4x + 2x^2,$

$f_1(x) = 1 + 2x, \quad f_2(x) = 2 + 4x + 5x^2, \quad f_3(x) = 5x + 2x^2.$

According to Equation (7), we now calculate the arrays $A(i), 1 \le i \le 4$, of set $A = A_1$. For instance, for $A^1(1)$ we have:

$h_{11}(1) = [g_1(x) \cdot f_1(x)]'|_{x=1} = 5,$

$h_{12}(1) = [g_1(x) \cdot f_2(x)]'|_{x=1} = 0,$

$h_{13}(1) = [g_1(x) \cdot f_3(x)]'|_{x=1} = 5.$

Thus, $A^1(1) = 505$. Similarly, $A^1(2) = 250, A^1(3) = 542,$ and $A^1(4) = 340$. Then, $A = \{505,250,542,340\}$. Assume that nodes $i = 1$ and $j = 4$ want to compute their pairwise secret key. Thus, node 1 sends $G(1)$ to node 4 and node 4 sends $G(4)$ to node 1. Finally, by Theorem (1), $K_{14} = K_{41} = 3$. The other shared secret keys between the other nodes are achieved in similar method.

B. Key agreement

After deployment, each node broadcasts a message including the node's identifier and the indices of the key spaces on the network. Assume that nodes i and j are neighbors. If they find at least one common array, they can establish a pairwise key by using Theorem (1): they first exchange their array in G and then compute a common key $K_{ij} = \left[h_i(x) \cdot g_j(x)\right]'\Big|_{x=1} = \left[h_j(x) \cdot g_i(x)\right]'\Big|_{x=1} = K_{ji}$. If they cannot find a common key between them, they can find a secure path. For example, let $i = v_0, v_1, \ldots, v_t, j = v_{t+1}$ be a path in the network in which every pair consecutive nodes v_{r-1} and v_r $(1 \le r \le t + 1)$ on the path has a secure link during the setup phase using the common keys in their key rings. Let L be the number of such paths that are disjoint and do not have any common link. Node i then generates L random keys $k_1, \ldots, k_L$ and sends each one to node j along a different path to node j. When node j has received all L keys, node i and node j compute the new link key as $k_{ij} = hash(k1||\ldots||kL)$.

Here we are interested in obtaining the computational overhead of our scheme. In the key agreement phase, the computational overhead of the multiplication of two λ-degree polynomials is $O(\lambda^2)$. By choosing the *Fast Fourier Transform (FFT)* method [11], the computational overhead is

$O(\lambda log\lambda)$. According to Equation (2), by using the primitive element for the construction of the public set G in our scheme, the computational overhead is equal to $O(\lambda)$.

V. EVALUATION OF THE PROPOSED SCHEME

There are various metrics to evaluate the performance and security properties in a WSN. A network designer must find suitable tradeoffs between the various metrics in a network. Some of these metrics are summarized as follows.

–*Connectivity:* Probability that two nodes share at least one common key.

–*Resilience against node capture:* Resilience against node capture is the fraction of total links which compromise by capturing x nodes which are not including the compromised links.

–*Storage memory:* Limitation of the number of keys which can be stored in each sensor node.

–*Computational overhead:* The amount of computation required to establish a key.

In this section, we compare our scheme and the MKSS in terms of connectivity, resilience, storage memory, and computational overhead. To do this, we employ two types of combinatorial design: *generalized quadrangles (GQ)* [6] and *modified hybrid symmetric* [12].

We implemented the MKSS and our proposed scheme based on the MHS and the GQ designs. We evaluated the behavior of the aforementioned schemes through the simpler Matlab simulations.

To simulate the behavior of the MKSS, we use the parameters τ, ω, and λ, where $2 \leq \tau < \omega$ and λ is the security parameter in the Blom's scheme [3]. We construct ω key spaces using Blom's scheme and assign τ randomly selected key spaces to each sensor node.

According to definition (8) and Remark (1), we find $v = (s+1)(st+1)$ points to construct a $GQ(s,t)$ design and use two pairs $(s,t) = (q,q)$ and $(s,t) = (q^2, q^3)$ to simulate the behavior of the proposed combinatorial property of the GQ design for parameters τ and ω.

For a sensor network with size N, we simulated the behavior of the proposed scheme based on the MHS design where we construct two similar key pools with d different keys between these. We use symmetric BIBD with parameter $(q^2+q+1, q+1, 1)$ [12] to generate b blocks of size $q+1$ and assign these b blocks that are generated from the first key pool to b nodes, where $b < N$. Finally, $N-b$ blocks generated from the second key pool are randomly assigned to the remaining $N-b$ nodes.

For any given choice of $\tau, \omega, \lambda, s, t, d, q$, and network size N, we performed our simulations in the following subsections.

A. Connectivity

Let $GQ(s,t)$ be a generalized quadrangle of order (s,t). According to Definition (8), there are $b = (t+1)(st+1)$ lines where each line intersects with $t(s+1)$ other lines. Thus, in a GQ design, the probability that any pair of nodes share at least a common key is

$$P_{GQ} = \frac{t(s+1)}{(t+1)(st+1)}. \quad (9)$$

According to Remark 1, we can use $GQ(q,q), GQ(q,q^2)$, and $GQ(q^2,q^3)$ designs, where $q \geq 2$ is an integer.

For a sensor network with N nodes, Dargahi et al. [12] have modified the hybrid design [6] which uses symmetric BIBD with parameters $(q^2+q+1, q+1, 1)$ to generate b blocks (key chains) of size $q+1$ such that $q^2+q+1 < N$ for the largest prime number q. In [12], the authors generate two key pools and introduce a parameter d which denotes the number of different keys between them. According to the analysis in [12], the probability P_{MHS} that any pair of blocks shares one or more objects in the MHS design is

$$P_{MHS} = \frac{b(b-1)+(N-b)(N+b-2d-1)}{N(N-1)}. \quad (10)$$

In the MKSS, the authors select the parameters τ and ω, where $2 \leq \tau < \omega$, to determine the performance and security in their scheme. They use the actual probability P_{actual} that any two neighboring nodes sharing at least one key space. Therefore,

$$P_{actual} = 1 - \frac{\left((\omega-\tau)!\right)^2}{(\omega-2\tau)!(\omega)!}. \quad (11)$$

In what follows, we investigate these probabilities to compare the connectivity between the existing schemes.

Fig. 1 compares the connectivity between the MKSS and the proposed scheme based on combinatorial properties of various combinatorial frameworks (e.g., the MHS and the GQ designs) for parameters ω and τ. We assume that $\omega \leq N$. In the proposed scheme based on the combinatorial property of the MHS design for parameters ω and τ, we select $d = 1$ and $d = 7$. The parameters in the MKSS are $\tau = 6,8$ and $\tau < \omega \leq 100$. For $\tau = 6$ and $\tau = 8$, the probability of key share from Equation (10) for $d = 1$ is greater than from Equation (11) when $\omega > 20$. Similarly, for $d = 7$ in the proposed scheme based on the MHS, $P_{MHS} > P_{actual}$ when $\omega > 40$ for $\tau = 6$ and $\tau = 8$. The value of d has a considerable role in the MHS design because small values of d lead to higher connectivity. When τ is very approximate to ω, the probability of key share for the MKSS increases. As a result, it shows that the parameter d in the MHS design has more effects than the parameters ω and τ in the MKSS for improving the connectivity metric. Note that the selection of a combinatorial design for parameters ω and τ in our scheme plays an important role in evaluating the proposed scheme. The connectivity metric of the proposed scheme based on GQ design is lower than the other schemes.

B. Resilience

In terms of resilience, we consider the probability that a link is compromised when an attacker captures x randomly nodes and thus x key chains. In this subsection, the resiliencies of the existing schemes are compared together.

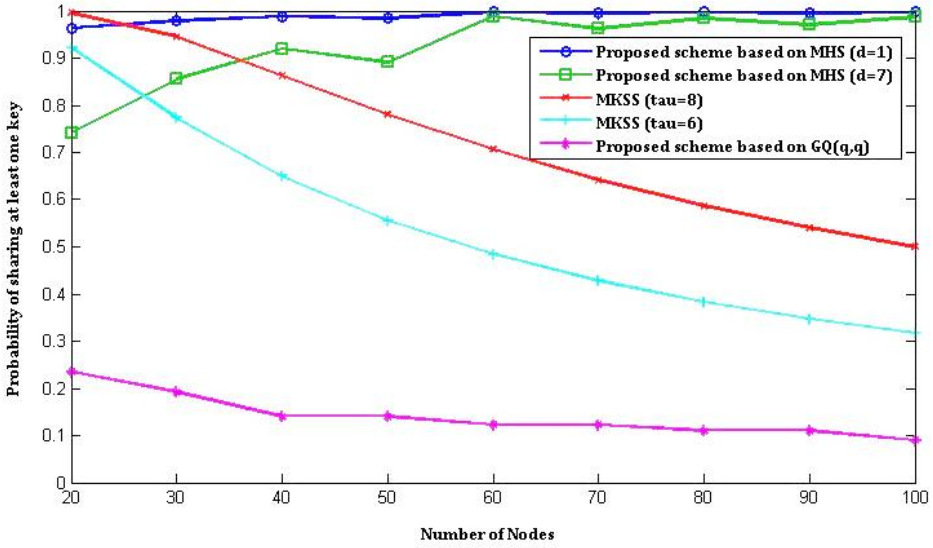

Fig. 1 Probability of sharing at least one key (connectivity) when two nodes each randomly chooses τ spaces from ω spaces in the MKSS and the proposed scheme based on combinatorial properties of various combinatorial frameworks for parameters ω and τ.

According to the analysis in [6], a key can be in $t+1$ of b blocks and if a link is included in key chains of both nodes of the link, then that link secured with a key j. Thus, the probability that a link is compromised when an attacker captures x key chains in $GQ(s,t)$ with b blocks can be defined as

$$P\{L|C_x\} \approx 1 - \frac{\binom{b-t-1}{x}}{\binom{b}{x}}, \tag{12}$$

where L and C_x denote the events that a link is compromised and x nodes (x key chains) are compromised, respectively. For fair comparison, according to Remark (1), we select the parameters $s = q^2$ and $t = q^3$.

In [12], each key appears in $q+1$ key chains and let C_x be the event that the adversary captures x nodes and thus x key chains. Therefore, the probability that link L is compromised when an attacker captures x key chains is computed for the MHS scheme as

$$P\{L|C_x\} \leq 1 - \frac{\binom{2q^2}{x}+2\binom{q^2}{x}}{\binom{2q^2+2q+2}{x}}. \tag{13}$$

To evaluate the resilience of the MKSS, the authors consider how the capture of x nodes by an adversary affects the fraction of communication among uncaptured nodes. To compute this fraction, they calculate the probability of a link being broken given x nodes are compromised as

$$P_{actual} = \sum_{j=1}^{\lambda+1} \binom{x}{j} \left(\frac{\tau}{\omega}\right)^j \left(1 - \frac{\tau}{\omega}\right)^{x-j}. \tag{14}$$

In this part, we compare the resilience between the MKSS and the proposed scheme based on combinatorial properties of various combinatorial frameworks (e.g., the MHS and the GQ designs) for parameters ω and τ. Assume that the MKSS given parameters ($\omega = 7,\ \tau = 2,\ m = 200,\ \lambda = 99,$

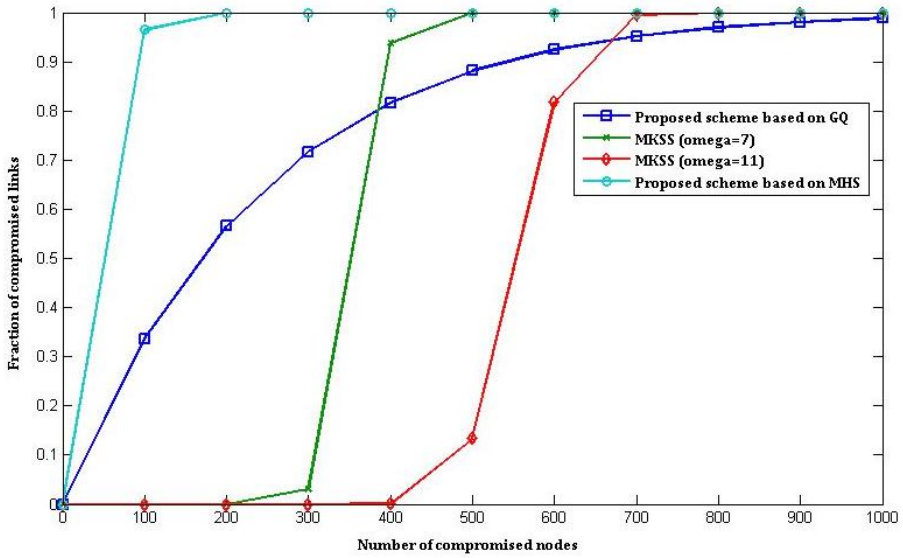

Fig. 2 Fraction of compromised links versus number of compromised nodes in the MKSS and the proposed scheme based on various combinatorial designs.

$P_{actual} = 0.5$). *Fig.* 2 shows that the fraction of compromised links in the proposed scheme based on the GQ design always has better than the MKSS's when the number of compromised nodes is higher than about 300 (over 30% compromised links). When the number of compromised nodes is higher than about 650 (over 65% compromised links), our scheme has better resilience than the MKSS given parameters ($\omega = 11,\ \tau = 2,\ m = 200,\ \lambda = 99,\ P_{actual} = 0.33$). Note that if suitable values are chosen for (ω, τ), then the proposed scheme based on the GQ design has only slightly significant resilience. The fraction of compromised links in the proposed scheme based on the MHS design scheme performs less than the other approaches in the network.

C. Storage memory

Another important factor in a WSN is storage memory at sensor nodes. The overall storage memory at sensor nodes in our scheme based on the MHS design becomes

$$m = (\lambda + 1)\mathrm{k}, \tag{16}$$

where k is key ring size. For the proposed scheme based on the GQ design, k is replaced by s. The overall storage memory at sensor nodes in the MKSS becomes

$$m = (\lambda + 1)\tau, \tag{17}$$

where τ is the number of random key spaces. Note that the parameter k for our scheme is fixed while the parameter τ in the MKSS changes between 2 to $\omega - 1$. In both schemes, set $\lambda << N$. *Fig.* 3 shows that for $\tau_{min} = 2$, the overall storage memory in the MKSS is better than our scheme. For instance, $(N, \lambda, \omega, m_{\tau_{min}}) = (200,23,20,48)$ and $(N, \lambda, \omega, m_{\tau_{min}}) = (700,76,70,154)$ while $(N, \lambda, k, m) = (200,23,14,336)$ and $(N, \lambda, k, m) = (700,76,26,2002)$ in our scheme. Note that

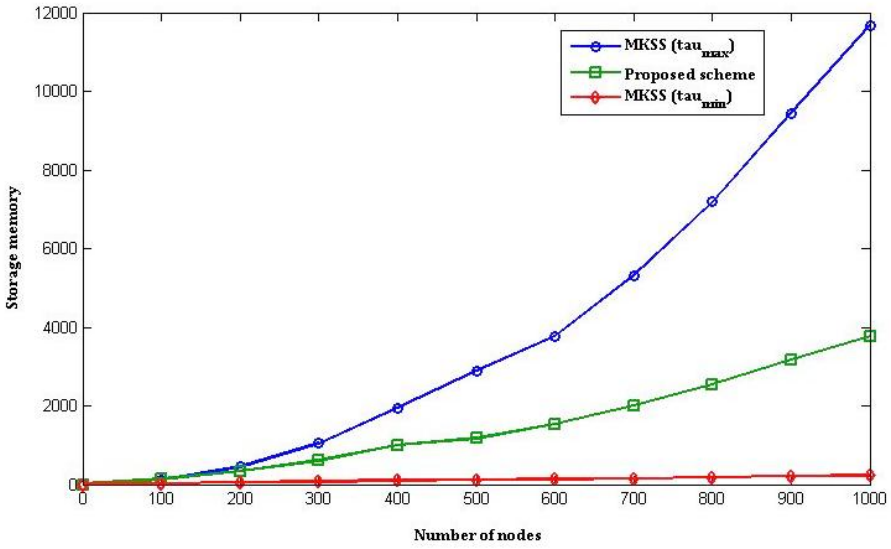

Fig. 3 Storage memory in the MKSS and our scheme.

providing suitable connectivity in the MKSS, the parameter τ must be close to $\tau_{max} = \omega - 1$. In the latter case, for $\tau_{max} > k$, *Fig.* 3 shows that the storage memory for our scheme is better than the MKSS's. For example, $(N, \lambda, \omega, m_{\tau_{max}}) = (400,49,40,1950)$ and $(N, \lambda, \omega, m_{\tau_{max}}) = (900,105,90,9434)$ while $(N, \lambda, k, m) = (400,49,20,1000)$ and $(N, \lambda, k, m) = (900,105,30,3180)$ in our scheme. In general, our scheme provides better storage memory than the MKSS's when $k < \tau \leq \tau_{max}$.

D. Computational overhead

Du et al. [13] construct an algorithm based on the multiplication of two matrices for establishing a key between two fixed nodes. The computational overhead, in this case, is $O(\lambda^3)$. Using *Strassen's algorithm* for matrix multiplication [11], the computational overhead can be $O(\lambda^{log\,7}) = O(\lambda^{2.81})$. To reduce the computational overhead, Du et al. choose the *Vandermonde matrix* as the public matrix which can be generated by a primitive element. Consequently, the computational overhead of their scheme is $O(\lambda)$. Our scheme provides better the computational overhead than the general case of the MKSS's [13] because it is $O(\lambda log \lambda)$. Similar to the MKSS, if we use the primitive element, then the computational overhead of our scheme and the MKSS are equal to $O(\lambda)$.

VI. DISCUSSION

In this section, we summarize the main results from our scheme.

—Our scheme proposes a modification of the MKSS which uses combinatorial properties for some parameters instead of a random selection of them. In the MKSS, the authors randomly select the parameters τ and ω, where $2 \leq \tau < \omega$. In this work, we use an arrangement of τ and ω to satisfy

combinatorial properties. We consider two combinatorial designs: the GQ design and the MHS design. Using these designs, we analyze performance, security, storage memory, and computational overhead in our scheme and compare our result with the MKSS.

—The MHS design has the highest connectivity when the parameter d is small. In the MKSS, fix the parameter τ. Observe that when τ is very close to ω (e.g. $\tau = \omega - 1$), the connectivity of the MKSS increases. To provide the better connectivity, the MHS design (with a small value of d) or the MKSS (with a large value of τ) would be preferred to the GQ design or the MHS design (with a large value of d) or the MKSS (with a small value of τ).

—The resilience of the MKSS exhibits interesting tradeoff with the resilience of the GQ design. For example, consider the MKSS with parameters ($\omega = 7,\ \tau = 2,\ m = 200,\ \lambda = 99,\ P_{actual} = 0.5$). The GQ design has better resilience than the MKSS's when over 30% links are compromised. If the MKSS is selected with parameters ($\omega = 11,\ \tau = 2,\ m = 200,\ \lambda = 99,\ P_{actual} = 0.33$), then the GQ design has only slightly better resilience. The MHS design has the lowest resilience.

—There is a trade-off between connectivity and storage memory. The storage memory in our scheme linearly increases, while this metric in the MKSS is related to $2 \leq \tau \leq \omega - 1$. To achieve higher connectivity in the MKSS, the parameter τ must be close to $\tau_{max} = \omega - 1$. Consequently, storage memory in the MKSS increases. Hence, for $\tau_{max} \geq \tau > k$, our scheme is better storage memory than the MKSS's.

—In the key agreement phase, the computational overhead of our scheme has better than the general case of the MKSS. Furthermore, if we use the primitive element similar to the MKSS, then the computational overhead of these schemes are equal.

VII. CONCLUSION

In this paper, we have proposed a deterministic key pre-distribution scheme for a wireless sensor network based on combinatorial structures such as designs. To achieve suitable tradeoffs between various metrics of interest in the wireless sensor network, we analyzed the general framework of the resulting schemes. In addition, we obtained a new key agreement phase by using derivative operation on univariate polynomials to aim a desired computational overhead. This deterministic process has a lower computational overhead than the general case of the MKSS's. Using the primitive element, the computational overhead in our scheme and the MKSS are equal. Finally, our analysis and experimental results show that our approach has lower storage memory than the MKSS's.

REFERENCES

[1] 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*, pp. 1-13, 2017.

[2] M. Anzani, H. Haj Seyyed Javadi, and A. Moeini, "A deterministic key pre-distribution method for wireless sensor networks based on hypercube multivariate scheme," *Iranian Journal of Science and Technology, Transactions A: Science*, DOI:10.1007/s40995-016-0054-3, 2016.

[3] R. Blom, "An optimal class of symmetric key generation systems," *in Advances in Cryptology: Proceedings of Eurocrypt 84*, Lecture Notes in Computer Science, vol. 209, Springer-Verlag, Berlin, pp. 335-338, 1984.

[4] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," *in Advances in Cryptology: CRYPTO 92*, LNCS, vol. 740, pp. 471-486, 1993.

[5] S.A. Camtepe, and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," *Rensselaer Polytechnic Institute*, Computer Science Department, Technical Report-TR-05-07, 2005.

[6] 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, 2007.

[7] H. Chan, A. Perring, and D. Song, "Random key pre-distribution schemes for sensor networks," *in Proceeding of IEEE Symposium on Security and Privacy*, pp. 197-213, 2003.

[8] C-Y. Chen, and H-C. Chao, "A survey of key distribution in wireless sensor networks," *Security and Communication Networks*, Wiley Online Library, DOI: 10.1002/sec.354, 2011.

[9] C.J. Colbourn, and J.H. Dinitz (Eds.), *Handbook of combinatorial designs* (2nd ed.). Boca Raton: CRC Press, 2007.

[10] M. Conti, R.D. Pietro, A. Gabrielli, L.V. Mancini, and A. Mei, "The smallville effect: social ties make mobile networks more secure against node capture attack," *in Proceedings of the 8th ACM International Workshop on Mobility Management and Wireless Access*, ACM, Bodrum, Turkey, pp. 99-106, 2010.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd edition, MIT Press and McGraw-Hill, 2009.

[12] T. Dargahi, H.H.S. Javadi, and M. Hosseinzadeh, "Application-specific hybrid symmetric key pre-distribution for WSNs," *Security and Communication Networks*, vol. 8, pp. 1561-1574, 2015.

[13] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, pp. 228-258, 2005.

[14] M. Ehdaie, N. Alexiou, M. Ahmadian Attari, M.R. Aref, and P. Papadimitratos, "Key splitting: making random key distribution schemes resistant against node capture," *Security and Communication Networks*, vol. 8, pp. 431-445, 2015.

[15] M. Ehdaie, N. Alexiou, and P. Papadimitratos, "Random Key Pre-Distribution Techniques against Sybil Attacks," *Journal of Communication Engineering*, vol. 5, no.1, pp. 1-13, 2016.

[16] L. Eschenauer, and V.D. Gligor, "A key management scheme for distributed sensor networks," *in Proceeding of the 9th ACM Conference on Computer and Communications Security*, pp. 41-47, 2002.

[17] A.M Frieze, and M. Karoński, *Introduction to Random Graphs*, Cambridge University Press, 2015.

[18] H.H.S. 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, 2015.

[19] 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, 2014.

[20] T. Kavitha, and D. Sridharan, "Hybrid design of scalable key distribution for wireless sensor networks," *IACSIT International Journal of Engineering and Technology*, vol. 2, no. 2, pp. 136-141, 2010.

[21] J. Kur, V. Matyas, and P. Svenda, "Two improvements of random key pre-distribution for wireless sensor networks," *in Proceedings of the International Conference on Security and Privacy in Communication Networks*, Padua, Italy, 2012.

[22] WS. Li, CW. Tsai, M. Chen, WS. Hsieh, and CS. Yang, "Threshold behavior of multi-path random key pre-distribution for sparse wireless sensor networks," *Mathematical and Computer Modeling*, vol. 57, no. 11, pp. 2776-2787, 2013.

[23] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Secure*, vol. 8, no. 1, pp. 41-77, 2005.

[24] I. Memon, "A secure and efficient communication scheme with authenticated key establishment protocol for road networks," *Wireless Personal Communications*, vol. 85, no. 3, pp. 1167–1191, 2015.

[25] S. Mitra, S. Mukhopadhyay, and R. Dutta, "Unconditionally-secure key pre-distribution for triangular grid based wireless sensor network," *Journal of Applied Mathematics and Computing*, vol. 44, no. 1-2, pp. 229-249, 2014.

[26] W.K. Nicholson, *Introduction to Abstract Algebra*, 4th edition, John Wiley & Sons, 2012.

[27] M.B. Paterson, and D.R. Stinson, "A unified approach to combinatorial key pre-distribution schemes for sensor networks," *Designs, Codes and Cryptography*, pp. 1-35, 2012.

[28] S.E. Payne, and J.A. Thas, *Finite Generalized Quadrangles*, MA: Pitman Advanced Publishing Program, Boston, 1984.

[29] S. Qian, "A novel key pre-distribution for wireless sensor networks," *Physics Procedia*, vol. 25, pp. 2183–2189, 2012.

[30] 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.

[31] S. Ruj, and B. Roy, "Key pre-distribution schemes using partially balanced designs in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 4742, pp. 431-445, 2007.

[32] Jr. M.A. Simpli'cio, P.S. Barreto, C.B. Margi, and T.C. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Computer Networks*, vol. 54, no. 15, pp. 2591-2612, 2010.

[33] C.W. Tsai, W.S. Li, W.S. Hsieh, C.S. Yang, and M.C. Chiang, "Analysis of Multi-path Random Key Pre-distribution for Wireless Sensor Networks," *in International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pp. 497-502, 2011.

[34] J. Zhang, and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63-75, 2010.

[35] H. Zhao, J. Hu, J. Qin, V. Varadharajan, and H. Wan, "Hashed random key pre-distribution scheme for large heterogeneous sensor networks," *in Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, Liverpool, pp. 706-713, 2012.