www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2017
http://www.sensorsportal.com

Hash Chains Sensornet: A Key Predistribution Scheme for Distributed Sensor Networks Using Nets and Hash Chains[1]

Deepak Kumar DALAI and * Pinaki SARKAR
School of Mathematical Sciences, National Institute of Science Education and Research,
Bhubaneswar 752 050, India
* Tel.: +919433531020
E–mail: pinakisark@gmail.com

Received: 24 April 2017 /Accepted: 24 May 2017 /Published: 31 May 2017

Abstract: Key management is an essential functionality for a security protocol; particularly for implementations to low cost devices of a distributed sensor networks (DSN)–a prototype of Internet of Things (IoT). Constraints in resources of constituent devices of a low cost IoT (example: sensors of DSN) restricts implementations of computationally heavy public key cryptosystems. This leads to adaptation of the novel key predistribution technique in symmetric key platform to efficiently tackle the key management problem for these resource starved networks. Initial key predistribution schemes (KPS) use random graphs; while later ones exploit combinatorial approaches that assure predictable design properties. Combinatorial designs like a *(v, b, r, k)–configuration* that forms a *μ–CID* are effective schemes to design a KPS. A net in a vector space is a set of cosets of certain kind of subspaces called partial spread. A *μ(v, b, r, k)–CID* can be formed from a net. In this paper, we propose a KPS for DSN, named as Sensornet, using a net. We observe that any deterministic KPS suffer from "smart attack" and hence devise a generic method to eliminate such attacks. Resilience of a KPS can improve by clever application of a Hash Chains technique introduced by *Bechkit et al*. We improve our *Sensornet* to obtain a new *Hash Chains Sensornet (HC(Sensornet))* by the applications of these two generic methods. Effectiveness of *Sensornet* and *HC(Sensornet)* in term of crucial metrics in comparison to prominent schemes has been theoretically established.

Keywords: Distributed sensor networks, Key management, Combinatorial designs, Attacks, Hash function.

1. Introduction

Distributed (Wireless) Sensor Networks (DSN) are revolutionary information gathering systems owing to their easy deployment and flexible topology. They are decentralized with numerous low cost identical resource starved wireless devices, called sensors or nodes that deal with sensory data.

They are considered to be a nice prototype of Internet of Things (IoT) which is a sophisticated concept that aims to connect our world beyond imagination. This has boosted the study of such DSN in recent times.

Prominent scientific applications of IoT are smart homes, smart cities, smart grids, smart water networks, agriculture, health-care, etc. Of particular interest are applications of DSN to networks where

[1] This paper is thoroughly revised and substantially extended version of our "best paper" awarded conference publication at Sensornet 2017. Title of our conference version is: "Sensornet: A Key Predistribution Scheme for Distributed Sensors using Nets" [1]. Additional sections in this work in comparison to earlier version are Sections 2, 6, 7 and 8 and all their subsections.

security is a premium. For instance, security may be essential for certain sensitive scientific and military networks that are meant for (i) self-healing minefields, (ii) military surveillance, (i) force protection arenas, and so on. Primary tasks of devices of an IoT in any such application are to collect information from their surroundings, process and forward them to other devices. Depending on specific applications, they may be further required to (i) track and/or classify an object, (ii) determine parametric value(s) of a given location, etc. These sensitive tasks for such critical applications create necessity of secure message exchange among the low cost IoT devices.

1.1. Type of Cryptosystem: KPS

Constraints in resources of constituent tiny devices of a low cost IoT (like sensors of DSN) make us opt for lightweight symmetric key cryptosystems (SKC) over expensive public key cryptosystems (PKC) while designing security protocols for such networks. SKC require both senders and receivers to possess the same encryption/decryption key before message exchange. Standard online key exchange techniques that involve PKC are avoided due to high cost factor.

Few trivial key distribution techniques are as below. First approach is to assign a single key for the entire network. This method is vulnerable to "single point *fail*ure" (compromise of one sensor reveals this single system key). Second is to think of assigning pairwise distinct SKC keys for every pair of devices. This later strategy overloads the memory of each sensor; $\mathcal{N}-1$ keys are required to be stored per sensor for a network of size $\mathcal{N}$. This is impractical for large networks (i.e., large value of $\mathcal{N}$ ($\approx 10^4$, say, or greater). Treating a node (or a few) as Trusted Authority (TA) is risky. This also makes the network prone to "single point *fail*ure" because capture of sensors acting as TA leads to vulnerable systems. Thereby schemes like LEAP [2] are avoided while designing secure key management schemes for DSN.

These facts emphasize the importance of proper employment of an adequate key management scheme. This situation was wittily overcome in 2002 by Eschenauer and Gligor [3] by introducing the concept of *key predistribution* that involves applications of SKC to sensor networks. Any KPS primarily executes:

- Key distribution: Prior to deployment, keys are preloaded into sensors to form their *keyrings or key chains* from the collection of all network keys, called *key pool*. Each system key is marked with a unique identifier *(key id)*. Certain schemes [4] consider *(node id)* as a unique function of all the key ids. These key or, node id are used during key establishment.
- Key establishment: The preloaded keys are established by a two steps process, as below:

(i) *Shared key discovery phase* establishes the shared key(s) among the participant nodes. This may be achieved by broadcasting the key ids of all keys contained in the nodes (or node id). On receiving each other's key ids, the sensors tally them to trace their shared key id(s), hence common shared key(s).

(ii) *Path key establishment phase* establishes a path key between a pair of nodes that do not share key. This process involves intermediate nodes. Refer to common intersection designs *(μ–CID)* in Section 4.

Depending on whether the above processes are probabilistic or deterministic, such schemes are classified into two types: (a) *random* and (b) *deterministic*. Sections 3.1 and 3.2 present a brief overview of individual type of schemes.

1.2. Summary of our Contributions

Observing the significant advantages of deterministic KPS during key management for low cost distributed networks, we set out to propose one such scheme. Our proposal uses net of partial spreads (or, nets) in a finite vector space that have been well studied combinatorically and as such, we name the scheme as *Sensornet*. Later we extend our protocol to a resiliency enhanced version, *Hash Chains Sensornet* (*HC(Sensornet)*). The process eliminates dangerous "smart attacks" (defined below in Section 2).

1.3. Paper Organization

Prior to the proposal of our protocols, we introduce in Section 2, all threat models that we consider in our work. Section 3 conducts a brief literature survey on KPS and then presents some preliminary theory related to combinatorial set systems that are required to construct such KPS. Construction of net of partial spreads or nets is reviewed in Section 4 and our scheme *Sensornet* is presented in Section 5. This is followed by proposal of a generic approach that eradicates "smart attacks" in Section 6. A generic Hash Chains based approach of Bechkit et al. [5, 6] that enhances resilience of any KPS is reviewed in Section 7. Our basic *Sensornet* scheme is then extended to *Hash Chains Sensornet* (*HC(Sensornet)*) by applications of these generic approaches. We analyze *Sensornet* and *HC(Sensornet)* in terms of performance metrics in Section 9 and thereby establish our scheme's efficiency in comparison to prominent proposals. We infer that our protocols adhere to the desirable criteria set out in Section 3.4. We summarize our work in Section 10 and state related future research directions in Section 11.

2. Threat Models

Passive eavesdroppers have little effect on KPS systems. So, our system's resilience is analyzed against two active adversarial attacks. They are *random node compromise* and *smart attacks*.

'Random node (compromise) attacks', as the name suggests, is the random compromise of nodes by an

adversary. This leads to partial disclosure of key pool ($\mathcal{K}$) of existing devices; thereby restricting the use of links that were secured by these keys.

"Smart Attack" [6] is essentially selective or "smart" compromise of (other) nodes that share the same key(s) as a pair of communicating nodes. This attack occurs because the key establishment process requires exchange of *key ids* in (unencrypted) plain text. This reveals the *key sharing graph* that can be beneficial to an adversary to selectively (or smartly) target specific nodes (see Pietro et al. [7] for details).

Most KPS solution exchange unencrypted set of their key ids or a unique function of this set, aka node ids during key establishment. Thus the sets of key ids (or the node ids) of these nodes becomes a public information. So, an attacker can easily compute the shared key ids by 'equating' them (like any node) and successfully launch a "smart attack" [7]. Few prominent examples where this happens are [5, 6, 8-13]. However, there are works [14, 15] that encrypt these secondary node ids or set of key ids before transmissions and thereby seal this attack.

Due to its predictable nature, selective capture of (uncompromised) nodes may have more devastating effects than the random model. This situation is rectified by exchange of encrypted sets of key ids; thereby eliminating "smart attack" (see Section 6).

3. A Brief Survey of KPS

This section presents a state–of–the–art survey of prominent KPS. We split survey into three stage: (i) random KPS (RPKS), (ii) deterministic KPS (DKPS), and (iii) advantages of later type over former. Thereby, we justify proposal of our new deterministic KPS adhering to design criteria set out in Section 3.4.

3.1. Random Key Predistribution Schemes

First generation KPS rely on random graph theory pioneered by Erdos and Renyi [16] to preload SKC keys into the sensors. Therefore, keyrings are formed randomly. This leads to probabilistic key sharing and establishment. Later is achieved by either broadcast of key ids or challenge and response. Refer to [3, Section 2.1]. Earlier, Blom proposed the first key distribution scheme [17] in public key settings meant for resourceful ad hoc networks. Blom's schemes uses pairs of public private matrices for key distribution. It cannot be applied to resource constraint sensor networks due to its heavy memory requirement to store huge vectors. Several researchers use variants of Blom's schemes to propose both random and deterministic KPS for DSN.

3.2. Deterministic Key Predistribution

Deterministic KPS were simultaneously proposed by [9, 10, 13] in 2004. The work [13] combines subset based schemes with existing key distribution schemes such as [17] to obtain multiple key spaces. The scheme of Lee and Stinson [10] uses quadratic equation solving and can be viewed as a scalable extension of their later proposal [11] that uses Transversal Design *($TD(k,p)$)*. This later work further summarizes the necessary conditions for a combinatorial design to yield a deterministic KPS. The work [9] exploits combinatorial designs like symmetric Balanced Incomplete Block Designs (BIBD), generalized quadrangles and projective planes [11, 12, 18]. Certain KPS [4] exploit special Algebraic structures like Reed Solomon code based KPS. However, these protocols permit alternate combinatorial descriptions that have been well studied in [14, 15, 18].

3.3. Advantages of DKPS over RKPS

Deterministic schemes have advantages over their random counterparts. For instance, a desired property of a randomized scheme may occur only with a certain probability whereas they can be proven to hold in a deterministic scheme [11, 12, 18]. This led to proposals of numerous deterministic KPS using various combinatorial tricks. Further the predictable nature of these combinatorial structures has been efficiently exploited to address design weaknesses of certain prominent KPS. For instance, the schemes [14, 15] addresses the connectivity and resilience aspect of [4, 11] by a deterministic design specific approach.

Contrary to these observations, Ruj and Pal [19] state that random graph models are well suited for 'scalability' and 'resilience'. Thereby, they justify their proposals of random graph based preferential attachment models with degree bounds. They design networks using their model. Their designs suffer from highly skewed load distribution, poor connectivity and resiliency; and so, are inappropriate for applications to (distributed) IoT.

In fact, sensitive IoT applications require protocols to yield equal distribution of tasks among peers. Moreover, to reduce hops and hence potential risks from node capture, it is more important to have connected networks that cannot be guaranteed by random schemes. So we opt for deterministic protocols for security applications in low cost IoT networks that assure predictable (high) connectivity; despite them having restricted scaling operations. This is a major area of study for most (deterministic) KPS proposals, including ours (recalled in Section 11).

Structure of the combinatorial objects used to design deterministic KPS cannot directly model networks of any specified size N. Usually, such structures result in designs having a specific pattern in the number of resultant blocks, viz. a prime power etc. Since N can be any number, a standard strategy is to consider the least prime power that is greater than the network size (i.e., $p^z \geq \mathcal{N}$). Then $\mathcal{N}$ subset are randomly selected to form the key rings of the network nodes. Bose et al. [8] speculate that random removal of blocks may have a disadvantageous effect on the underlying design's properties and hence become an issue of concern. Fortunately, this claim of Bose et al. [8] has been successfully challenged by Henry et al. [20]. Through practical experiments, they establish that random removal of key rings of a combinatorial KPS has negligible effect with overwhelming probability. This work reestablishes the importance of combinatorial schemes. Further these excess block may be useful in (restricted) scaling of a network.

3.4. Desirable Design Criteria

Devices of a low cost IoT (example: sensors of a DSN) are highly prone to damage and/or physical capture. This is a crucial consideration during the design of an (energy) efficient KPS. Prime objectives of any KPS is to ensure that the resulting network:

1. Has less number of keys per node, i.e., sizes of individual keyrings are less;
2. Have large *node support*, i.e., support large number of network nodes;
3. Has good (ideally full secure) *connectivity*. Secure connectivity (or, simply *connectivity*) is the ratio of number of (secure) links in eventual network to all possible links. A pair of nodes are said to be connected by a (secure) link if there exists at least one secret key between them;
4. is *resilient* against various types of adversarial attacks. A prevailing method adopted in most existing works [9-15,18] is to show that a standard resiliency coefficient *fail(t)* is minimized. We follow suit. The quantifier *fail(t)* measures the ratio of links broken after compromise of *t* sensors to the total number of links in the remaining network. Formally, *fail(t)* = u_t / b_t, where b_t is the number of links broken when *t* nodes are compromised and u_t is the total number of links in the remaining network of uncompromised nodes.

Ideally a KPS should have small keyrings, large network support with appreciable resiliency, scalability and (secure) connectivity. However, prominent researches prove the impossibility of construction of a *perfect KPS* that meets all these criteria [11, 18]. This motivates proposal of designs that are robust for specific purpose. In the same spirit, we propose our schemes, *Sensornet* (in Section 5) and its resiliency enhanced version, *Hash Chains Sensornet (HC(Sensornet))* in Section 8 that are derived from net of partial spreads (or nets). Our schemes have a good balance of these combinatorial properties and so, are useful to design a deterministic KPS.

3.5. Deteriorated Resilience of KPS

Any KPS assigns multiple sensors to a given key. For a deterministic scheme, this value is the (regular) degree *r* of the design (refer to Section 4 below). Therefore, compromise of a node exposes partial key rings of uncompromised ones that affect their secure communication. This makes the resultant system vulnerable to various node capture attacks; thereby affecting the system's *resilience*. Several researches develop tricks to reduce the effect of this attack. We discuss a prominent effort in a Section 7 while extending our *Sensornet* to *Hash Chains Sensornet*.

4. Preliminaries

This section introduces definitions and notations that are required to describe our scheme, *Sensornet*.

4.1. Combinatorial Set Systems and KPS

Construction of generic deterministic KPS by the use of a combinatorial designs is presented in the paper [11]. A unified treatment of prominent combinatorial designs in terms of partially balanced *t – design* is present in [18]. We present below the basic design theoretic concepts:

Let $\mathcal{X}$ be a finite set. Elements of $\mathcal{X}$ are called varieties. Each subset of $\mathcal{X}$is termed as a block. Consider $\mathcal{A}$ to be a collection of blocks of $\mathcal{X}$. Then $(\mathcal{X},\mathcal{A})$ is said to be a *set system or, a design*. $(\mathcal{X},\mathcal{A})$ is regular of degree *r* if each point is contained in *r* blocks.

$(\mathcal{X},\mathcal{A})$ is uniform (of rank *k*) if all blocks have the same size, say *k*. A design $(\mathcal{X},\mathcal{A})$ is said to form a *(v,b,r,k)–design* if: $|\mathcal{X}| = v$, $|\mathcal{A}| = b$; it is regular of degree *r* and of uniform of rank *k*. A *(v,b,r,k)–design* forms a *(v,b,r,k)–configuration* if any arbitrary pair of blocks intersect in *at most* one point. Moreover, if any pairs of varieties occur in exactly one block, then a *(v,b,r,k)–design* forms a *(v,b,r,k)–BIBD* (Balanced Incomplete Block Designs). They can be used to construct various KPS [11] by mapping:

a) *v* varieties of $\mathcal{X}$to the set of keys in the scheme (i.e., $|\mathcal{X}|$ = *v* =*size of key pool*),
b) *b* to the number of network nodes in the system (*:=network size*),
c) *k* to the number of keys per node (*:=size of key rings*), and
d) *r* to the number of nodes that share a given key (*:=degree of the resultant KPS*).

The target is to construct KPS with identical burden on each sensor. This leads to opting for design with uniform rank (*k*) and regular degree (*r*), so that every *key ring* is of equal size (*k*) and same number of nodes (*r*) share each key for the resultant network.

Block graph $G_{\mathcal{A}}$ of the set design $(\mathcal{X},\mathcal{A})$ is defined with the vertex set $\mathcal{A}$ and edge set $E_{\mathcal{A}} = \{(A, B) : A, B \in \mathcal{A} \text{ and } A \cap B \neq \emptyset\}$. If the set design is regular of degree *r* and uniform of rank *k*, then the block graph $G_{\mathcal{A}}$ is *k(r 1)–regular*. A *(v,b,r,k)−configuration* $(\mathcal{X},\mathcal{A})$ is said to form a *μ−common intersection design* (*μ−CID*) in case for every pairwise empty intersection of blocks, there exists *μ* other blocks that share common keys with these blocks. It is important to construct designs that maximize the value of *μ*.

4.2. Net of Partial Spread or Nets

Let IF$_p$ be the finite field on p elements where p is a prime. Denote by $V_n(= IF_p^n)$ to be the vector space of dimension n over the field IF$_p$ with zero vector **0**. Since the finite field IF_p^n is a vector space over IF_p and is isomorphic to IF_{p^n} [21], we interchange the notation as per its suitability. This isomorphism mapping can be considered as a mapping from a basis set of IF_p^n to a basis set of IF_{p^n}. We consider $n=2m$ to be an even integer in this work.

A partial spread Σ of order s in V_n is a set of pairwise supplementary *m–dimensional* subspaces $E_1, E_2, \ldots, E_s$ of V_n i.e., $E_i \cap E_j = \{0\}$, $1 \leq i < j \leq s$. A partial spread Σ forms a spread if $\bigcup_{i=1}^{s} E_i = V_n$. It is known that a spread of V_n exists since m divides n [22], then $|\Sigma| = p^m + 1$. Therefore, from a given spread Σ each of the choices of s members of Σ provides a partial spread of V_n. Note that a partial spread might not be a subset of a spread (refer to Eisfeld and Storme [23]). An interested reader can refer to the books [24, 25].

Let E be a subspace of the vector space V_n. A coset of E in V_n is of the form $\alpha + E = \{\alpha + v\colon v \in V_n\}$ for an $\alpha \in V_n$. The set of cosets makes a disjoint partition of V_n. The element α is called a coset representative of the coset α+E. Since E is an additive group, any element from the coset α+E can be a coset representative. Given a partial spread $\Sigma = \{E_1, E_2, \ldots, E_s\}$ in V_n, let E_i be a set of coset representatives of subspace E_i for $1 \leq i \leq s$. Then the set $A = \{\alpha + E_i : \alpha \in E_i, 1 \leq i \leq s\}$ i.e., set of all cosets of subspaces E_i, $1 \leq i \leq s$ forms a *net* in V_n. An interested reader is referred to the book by Johnson et al. [25] for a detailed study of nets.

4.3. Examples of Partial Spreads

There are numerous constructions of spreads and partial spreads that can be found in the literature [25]. Now we present a few spreads S in $IF^n{}_p$, where p is a prime. For a given s, any $\Sigma \subseteq S$ such that $|\Sigma| = s$ forms a partial spread of order s. By Theorem 1 (see below in Section 5), this partial spread yields a KPS.

Spread I: This is a classic example of a spread from the additive group of the finite field IF_p^m. Since $n=2m$, IF_p^m is a subspace of IF_p^n with respect to a basis. Let $\{\alpha_i : 1 \leq i \leq p^m+1\}$ be a set of coset representative of the cosets of the subgroup IF_p^m in the multiplicative group $IF^n{}_p$. Then the set S$_{\mathrm{I}}$ = $\{S_i = \alpha_i IF_p^m, 1 \leq I \leq p^m + 1\}$ is a spread in IF_p^m.

Spread II: This example of spread is represented in bivariate form (see [26]). For each $\alpha \in IF_p^m$, define a subspace U_α of $IF^m{}_p \times IF^m{}_p$ by $U_\alpha = \{(\alpha u, u) |\ u \in IF_p^m\}$ and for sake of the consistency $U_\infty = \{(u,0) | u \in IF_p^m\}$. The set $S_{II} = \{U\alpha : \alpha \in IF_p^m\} \cup U_\infty$ constitute a spread in $IF_p^m \times IF_p^m \simeq IF_p^n$.

Spread III: This example of spread is generated from pre-quasifield, which is defined as following. A system $Q=(V,+,\circ)$, with finite $|V|$, is a pre-quasifield if the following axioms holds:

1. $(V, +)$ is an abelian group, with identity **0**.
2. $(V^*, \circ)$ is a quasigroup where $V^* = V \backslash \{\boldsymbol{0}\}$. That is, for any $a \in V^*$, the left multiplication operator $a \circ x$ and the right multiplication operator $x \circ a$ are both bijective from V^* to V^*.
3. $\forall x, y, z \in V$, $(x + y) \circ z = x \circ z + y \circ z$.
4. $x \circ \boldsymbol{0} = \boldsymbol{0}$, $\forall x \in V$.

Now assuming $(IF_p^m, +, \circ)$ is a pre-quasifield, set $E_a = \{(x, a \circ x) : x \in IF_p^m\}$ for any $a \in IF_p^m$ and $E_\infty = \{(0,x) : x \in IF_p^m\}$. Then it can be checked that $S_{III} = \{E_a : a \in IF_p^m \cup \{\infty\}\}$ is a spread in$IF_p^m \times IF_p^m$ [25]. Many pre-quasifields are available in literature. Refer to [27] for three types of pre-quasifields on set IF_2^m and [28] for a pre-quasifield on set IF_p^m.

Example 1 (of NETS): Here, we present a simple KPS from the spread of type **S$_{\mathbf{I}}$**. Take $V_n = IF_9 = IF_3[x]/(x^2+1)$. Consider the subspace $IF_3 = \{0,1,2\}$ and $\{1, x, x+1, x+2\}$ as a set of coset representatives of IF$_3^*$ in IF$_9$. Then $\boldsymbol{S_I} = \{\{0,1,2\}, \{0,x,2x\}, \{0,x+1,2x+2\}, \{0, x+2, 2x+1\}\}$ is a spread in IF$_9$. Consider a partial spread $\Sigma = \{E_1 = \{0,1,2\}, E_2 = \{0, x, 2x\}\}$ where $\overline{E_1} = E_2$ and $\overline{E_2} = E_1$. So, by Theorem 1, the set $\mathcal{X}$= IF$_9$ and the net $\mathcal{A}$={{0,1,2},{x,x+1,x+2},{2x,2x+1, 2x+2}, {0,x,2x},{1,x+1,2x+1},{2,x+2,2x+2}} forms a KPS **($\mathcal{X}$,$\mathcal{A}$)**. The block graph of **($\mathcal{X}$,$\mathcal{A}$)** is the K$_{3,3}$.

5. Sensornet

Sensornet is a class of KPS for distributed sensor networks. The design of *Sensornet* is based on partial spread or nets and is a consequence of the forthcoming set design and Theorem 1.

Given a partial spread S = $\{E_1, E_2, \ldots, E_s\}$ in V_n, let $\overline{E_\imath}$ be a supplementary subspace of E_i in V_n (their direct sum $E_i \oplus \overline{E_\imath} = V_n$ and $E_i \cap \overline{E_\imath} = \{\mathbf{0}\}$). One can check that $\overline{E_\imath}$ is a set of coset representatives of E_i for$1 \leq i \leq s$. Note that the subspaces E_i's in a partial spread are pairwise supplementary. So, any E_j, $j \neq i$ can be chosen as $\overline{E_\imath}$. Consider a set system **($\mathcal{X}$,$\mathcal{A}$)** with $\mathcal{X}$=V_n and the set of blocks, $\mathcal{A}$= {$\alpha + E_i$: $\alpha \in E_i$, $1 \leq i \leq s$}, which is a net in V_n. Then we have the results below:

Theorem 1. Given any partial spread Σ, the set design **($\mathcal{X}$,$\mathcal{A}$)** is $\mu(p^n, sp^m, s, p^m)$–*CID* for $\mu = (s-1)p^m$.

Proof. Here $v = |X| = p^n$. Consider two blocks $\alpha + Ei$ and $\beta + Ej$, then we have the following cases:

1. If $i = j$, then
   a. $\alpha + E_i = \beta + E_j$ *if* $\alpha = \beta$ or,
   b. $(\alpha + Ei) \cap (\beta + Ej) = \emptyset$ if $\alpha \neq \beta$.
2. If $i \neq j$, then we shall show that:
   $|(\alpha + Ei) \cap (\beta + Ej)| = 1$.

E_i and E_j are supplementary to each other. So, the element $\alpha - \beta \in V_n$ can be uniquely expressed as $-u + v$ where $u \in E_i$ and $v \in E_j$. That is, $\alpha - \beta = -u + v$ => $\alpha + u = \beta + v$ is a unique element in $(\alpha + Ei) \cap (\beta + Ej)$.

So, the number of blocks i.e., the number of cosets is $b = sp^m$ and each block contains $k = p^m$ elements. Given a subspace E_i, $i \in \{1, 2, \ldots, s\}$, each element $u \in V_n$ belongs to exactly one coset of Ei. So, each $u \in V_n$

belongs to exactly ***s*** many blocks in A. The set design $(\mathcal{X},\mathcal{A})$ is regular with r = s. Every two distinct blocks intersect each other by at most one element which implies that $(\mathcal{X},\mathcal{A})$ is a (p^n,sp^m,s,p^m)– configuration.

We observe that two blocks $\alpha+E_i$ and $\beta+E_j$ do not intersect if $i = j$ and $\alpha \neq \beta$, i.e., both are distinct cosets of same subspace E_i. For the case of non–intersecting blocks $\alpha+E_i$ and $\beta+E_i$, $\alpha \neq \beta$, both blocks intersect all other blocks of the form $\gamma+E_j$ where $j \neq i$. Since there are$\mu=(s-1)p^m$ such blocks $\gamma+E_j$ in A, $(\mathcal{X},\mathcal{A})$) is a $(s-1)p^m(p^n,sp^m,s,p^m)$–*CID*.

Here, the set of block $\mathcal{A}$ of the scheme $(\mathcal{X}, \mathcal{A})$ forms a net in a vector space. It can be checked then the block graph of $(\mathcal{X},\mathcal{A})$ is a strongly regular graph with parameters $(n=sp^m,\ r=(s-1)p^m,\ \lambda=(s-2)p^m,\ \mu=(s-1)p^m)$. Moreover, the block graph is a complete s–partite graph. In the study of finite geometry, the varieties together with the blocks (i.e., cosets) form the points and lines of an affine plane. Since two non–parallel lines (i.e., $\alpha+E_i$ and $\beta+E_j$ for i ≠j) intersects at one point, this set of cosets is called a net. We name the scheme as *Sensornet*.

6. Eradication of Smart Attacks

Sensornet, like all combinatorial KPS is susceptible to "smart attacks" (formalized in Pietro et al. [7]). We briefed this attack in Section 2. This section devises a generic method to reduce it to random node compromise attacks.

Our approach requires locally or group-wise random of nodes with one (ordinary) node in each group acting as its lead. Each group member is assigned exact one extra key for secure communication with its lead. While these leads stores an additional $g + l - 1$ keys for its g children and $l-1$ co-leads. Therefore $\mathcal{N}(= b) = gl$ and so, $g = l =\sqrt{\mathcal{N}}$is an optimized value. Association of keys and primary addresses (IP/MAC) can be stored in a table and preloaded in these group leads, so that *key establishment is not required for these extra keys*. These extra keys are thus be available to secure communication between these leads and their children. We use these extra keys for secure exchange of node ids during initial or subsequent (seldom) key establishment phases during a network's life time. Recall that these node ids are unique function of the set key ids of the preloaded keys for a given node and so are *secondary* ids. They are consequences of the combinatorial design construction meant for a deterministic KPS.

We propose a modified key establishment protocol as below. Sensors securely transmit their own node ids to their leads by the extra key. The leads circulates these node ids among themselves (in communication range). Node ids being linear functions are easy to "equate" and hence their collective computation burden is not abrupt. Moreover leads can apply a distributed algorithm to reduce the mutual burden on these leads. A table of shard key-primary address (IP/MAC) is formed during run time for each node. This tabular representation is returned to individual sensors and destroyed instantaneously. Node ids of children and co-leads are also flushed. So these leads retain data required only for their own conversations. Their children also gets only those information that concerns themselves.

Being orders less in number (=$\sqrt{b}$), we assume non compromise of the leads during the (short lived) key establishment phase(s). In fact, all existing works assume absolute trust on all system devices that includes their non-compromise during this phase. We are less *restrictive* and allow (random) "children compromise attacks" during key establishment. Refer to Section 2. Our construction does not reveal key sharing graph for "children compromise attacks" during key establishment as they do not ever possess other nodes' ids. Therefore, it is reasonable to assume concealment of the key sharing graph during key establishment. Post key establishment capture of any children does not reveal the key sharing graph due to same reason. Moreover, destruction of relevant information from the leads means that their capture at a late stage does reveal these node ids. Therefore, compromise of any sensor (lead or children) during (later) run time of the system does not reveal node ids, though all information about keys of that sensors are exposed. Thus cycles of keys that are not contained in exposed nodes are not disclosed. In effect "smart attack" cannot be launched.

The remedial construction may seem to convert a distributed system into a hierarchical one. Well, the hierarchy is required only during key establishment. We do not advocate use of these extra keys of the leads and/or their children for message exchange later on. (This is because "single point attack" on the leads may reveal the message (exchange) of their children.) Once a key establishment phase is over, the leads have no role in their children's conversations; these leads are not gateways for future conversations of their children. Therefore, our seemingly hierarchical construction retains its distributed flavor for the entire life time of the network barring the short lived key establishment phases(s). This is contrary to an inherent hierarchy in most key management protocols like [14, 15].[2]

7. Lightweight Resilience Improvement

Degree of any KPS lead to a security deterioration due to node capture attacks (defined to Section 2). A

[2] We do not advocate use of the extra keys in leads and their children for message exchange after key establishment phases. Our system remains distributed later; as opposed to an inherent hierarchy in key management protocols [14, 15].

network's resilience against such attacks is of vital importance. Many work aim to improve this aspect.

One such approach, due to Bechkit et al. [5, 6], presents a cute application of (recursive) hash function in a generic fashion. Their Hash Chains scheme $HC(x)$ successfully improves resilience of any KPS x without affecting other parameters and is briefed below:

- node ids of nodes vary from 0 to $b - 1$ where b is the number of blocks of the underlying combinatorial design. Observe that $b \approx N$.
- given a key K, let us inductively define $H^i(K) := H(H^{i-1}(K))$. That is, $H^i(K)$ denotes the i times use of the hash function H on key K for $i \in Z_+$.
- due to resource constraints, let the maximum number of times that we can repeat this (recursive) hash function computation in any sensor be N-1, ($1 \leq N \leq b \approx N$).
- node ids are used to discriminate initial preloaded KPS keys as described below:
  (i) instead of original keys, K, they preloaded a node with id i with the key $H^{(i \bmod N)}(K)$, for each key K in the i-th node $(0 \leq i < N)$;
  (ii) thus, two nodes with id i and j that shared the same key K in original KPS x end up with $H^{(i \bmod N)}(K)$ and $H^{(j \bmod N)}(K)$;
  (iii) if $(j - i \bmod N) > 0$ then node i calculates $H^{(j-1 \bmod N)}(k^i)$. *Preimage resistant property* of the (cryptographic) hash function H *implies* node j cannot find $H^{(i \bmod N)}(K)$.
- key establishment of these nodes uses set of key ids or node ids and is same as the original KPS.
- nodes i, j establishes their shared key, $SK = H^l(K)$, where $l = \max(i \bmod N; j \bmod N)$. This SK can be computed at either end, in case they possess either key $H^{(i \bmod N)}(K)$ or $H^{(j \bmod N)}(K)$. Node a computes
$$H^{l-a}(H_1{}^a(K)), \text{ for } a = i, j:$$
- capture of i-th node exposes all its keys $H^{(i \bmod N)}(K)$ to the adversary, who:
  (i) cannot establish links with the nodes that possess the keys $H^{(j \bmod N)}(K)$, for any key K in the i-th node and $j>i$ (mod N);
  (ii) can establish link with nodes that possess a key $H^{(j \bmod N)}(K)$ for $j < i$ (mod N).
  (iii) resilience dip is 30% for node capture.

So, in this Hash Chains based schemes $HC(x)$, connectivity, storage overhead and communication overhead remains same as the original scheme x. Application of our generic "smart attack" removal method (devised in Section 6) reduces any "smart attack" to random node compromise attack. Therefore the combination of the two approaches improves the original system's resilience by 50% against any node compromise attack (see Theorem 3). This combined technique is suitably adapted to enhance resilience of our *Sensornet* scheme and yield a new protocol: *Hash Chains Sensornet* in Section 8.[3]

8. *Hash Chains Sensornet (HC(Sensornet))*

Prototyped application of Bechkit et al.'s idea to our *Sensornet* scheme produces a resilience enhanced scheme that we term as *Hash Chains Sensornet*. Therefore considering x = *Sensornet*, the shared key between the node i, j with $(j \bmod N) > (i \bmod N)$ is computed as described below.

Let K is the distributed keys between the i^{th} and j^{th} nodes for $j > i \bmod N$. Then the nodes i, j compute shared secret key as

$$SK = H^j(K):$$

i-th node computes this shared secret key as

$$SK = H^l(H^i(K)$$

where $l = (j-i \bmod N) > 0$ So, in this Hash Chains based schemes $HC(x)$, connectivity (range), storage overhead and communication overhead remains same as the original scheme x.

Key establishment process is similar to *Sensornet* with modification. We store the basis vectors in leads, say E_i (α= 0 is natural choice). So its children $\alpha+E_i$ contains only α*'s*. These α*'s* are exchanged securely using the extra keys. Therefore the leads E_i's gets the entire node ids $(\alpha; \beta_i^1, \beta_i^2, \beta_i^3, \ldots, \beta_i^m)$ of each of their children $\alpha+E_i$, $\alpha=1,2\ldots m$. So in our *HC(Sensornet)* these leads can only perform the three steps given in Section 9.1 (nodes do not have their own id). Of course our *Sensornet* could have been combined with this unique "smart attack" removal technique. So, key establishment of both these combined schemes require less ($O(\log_p N)$) data transfer; computation complexity is same as original *Sensornet* scheme that was presented in Section 5.

9. Analysis of Our Protocols

In this section we compute the values of some important metrics involved in the protocols we have proposed, i.e., *Sensornet* and *HC(Sensornet)*.

9.1. Time (T_k) and Space Complexities (M_k) for Key Establishment

For the key establishment between two nodes, the nodes need to discover a common key stored between them. For this purpose, the nodes need to broadcast some data, which is required to trace the common key between two nodes. Since the sensor nodes have low memory and computation power, data and time requirement for key establishment are two very important factors to design a KPS. In this subsection we discuss the process of key establishment between two nodes and associated time and data requirement of the process. In case of path key establishment (refer to Section 1.1), both the concerned nodes have to find a common neighbor with whom they discover their shared key and establish connection via this neighbor.

[3] We denote a keys by K (CAPITAL LETTER) and its id by k (small letter) throughout this work. A full domain hash function is a suitable candidate for our work (like [5, 6]).

Denote by T_k and M_k to be the time and memory complexity functions for the key establishment.

The blocks of *Sensornet* forms a net, i.e., they are affine spaces. So, nodes can be identified by their basis vectors and key establishment is done using the node id. When nodes $\alpha+E_i$ and $\beta+E_j$ need key establishment between them, they follow the process below:

Step 1: The nodes $\alpha+E_i$ and $b + E_j$ compare the last m (basis) vectors in their node id. If they are same then follow Step 3 else follow Step 2.

Step 2: Here, we have $E_i \neq E_j$ i.e., so they share a common key. Let this key be $\alpha+u = \beta+v$, where $u \in E_i$ and $v \in E_j$. Now we need to find u and v in terms of the basis vectors of E_i and E_j respectively. Here, $\alpha–\beta$=v–u $\in V_n$. Since E_i and E_j are supplementary subspaces in V_n, $\alpha–\beta$ can be uniquely expressed as a linear combination of the basis vectors of E_i and E_j. This leads to the fact that this common key is in both E_i and E_j. The time complexity in this step is the time complexity to express $\alpha–\beta$ in terms of the basis vectors in a basis i.e., $O(n^3)$. Same is true for *HC(Sensornet).*

Step 3: In this case, $E_i = E_j$ i.e., they do not share a common key. So, they have to establish connection through another node with whom they share a key, individually. That is, they have to find a node $\gamma+E_k$ where $k \neq i$. The probability of finding such a node using a random pick up is s–1/s which is very high. Since both $\alpha+E_i$ and $\beta+E_j$ share a key with $\gamma+E_k$, each one does the same process described in Step 2 with $\gamma+E_k$ to discover their common key. After that $\alpha+E_i$ and $\beta+E_j$ can establish connection through $\gamma+E_k$. So, the time complexity in this case is $O(n^3)$.

So, each node spends $M_k = (m+1)*n*(\log_2 p) = O(n^2)$ bits of data for broadcasting of their identity and the time complexity to discover the common key(s) is $T_k=O(n^3)$. Note that in *Sensornet* and so, in *HC(Sensornet),* nodes broadcast only node id, that are only $O(n^2)$ bits instead of $O(rp^{n/2})$ many (all) key ids as broadcast by other prominent schemes.

9.2. Key-node Ratio (σ)

The key-node ratio is defined as $\sigma = k/b$. This ratio provides idea about the storage requirement of the scheme at each node with respect to the total number of nodes. With this metric we can compare the storage requirement of the schemes from different designs. It is desirable for this ratio σ to be as small as possible as lesser amount of memory required for key storage at each node. In both our schemes, *Sensornet* and *HC(Sensornet)*, value of key-node ratio is

$$\sigma = \frac{p^m}{sp^m} = \frac{1}{s} = \frac{1}{\Sigma}.$$

If the size of partial spread is larger, then the storage requirement to store keys in *Sensornet* is lesser.

9.3. Resiliency *(fail(t))*

Schemes should be equipped to perform against adversarial attacks. To this end, the standard resiliency metric *fail(t)* need to be minimized. This is prevalent method adopted by most existing works [4, 9,-12, 14, 15]. The quantifier *fail(t)* measures the probability that a random link between two sensor nodes is broken due to the compromise of *t* other random nodes. Formally, *fail(t)* = b_t/u_t where b_t is the number of links broken when *t* nodes are compromised and u_t is the total number of links among uncompromised nodes of remaining network. Theorem 2 is due to Lee and Stinson ([11, Section VIII]) provides the formula to compute *fail(t)* for any *(v, b, r, k, 1)–configuration*.

Theorem 2. Theorem 2. *For any (v, b, r, k, 1)–configuration, the value of the metric fail(t) on random compromise of t nodes is given by:*

$$fail(t) = 1 - \left(\frac{b-r}{b-2}\right)^t \tag{1}$$

Corollaries 1 is an immediate outcome of substituting the values of b and r in Equation 1, for the scheme *Sensornet*.

Corollary 1. The value of the resilience *fail(t)* for the set design ***(𝒳,𝒜)*** of the scheme *Sensornet*, which is a *(p^n,sp^m,s,p^m)−configuration* is

$$fail(t) = 1 - \left(\frac{sp^m-s}{sp^m-2}\right)^t \tag{2}$$

Clearly, the metric *fail(1)* = $O(p^{-m})$ i.e., if a node *N* is compromised, then the probability that a link (which is not incident with *N*) is $O(p^m)$. Here, the size of the partial spread has no significant effect on *fail*(1). For example, with $p = 2$, $n = 10$ (i.e., network size= $2^{10} > 1000$), the value of *fail*(1) = 0:03.

Theorem 3. Value of our resilience metric fail(t) for our HC(Sensornet) design is given by:

$$fail(t) = \frac{1}{2}\left(1 - \left(\frac{sp^m-s}{sp^m-2}\right)^t\right) \tag{3}$$

In particular, fail(1) =

$$fail(1) = \frac{1}{2}\left(\frac{s-2}{sp^m-2}\right) \approx \frac{1}{2\times p^m} \tag{4}$$

Proof. The result follows the observation made in point 6(*c*) during the brief of Bekhkit at al. along with Theorem 2. Of course our *Sensornet* and *HC(Sensornet)* schemes are assumed to be blessed with the generic “smart attack” removal approach. (Lee and Stinson, 2005) like most KPS assume only random node compromise attack in their analysis. Therefore, we rectify this half analyzed situation.

9.4. Connectivity (*ρ*)

We say two blocks in a set system are connected by d links (or, are at a distance *e*) if the shortest path between them in the block graph includes ‘e’ edges. Hence, we define the metric connectivity (or, connection probability) ρ_e of the network to be the probability that two nodes (placed in physical neighborhood) are connected by *e* links for a positive integer '*e'*.

Observe that the value of d for a μ CID with $\mu > 1$ is either 1 (if they share a key) or 2 (if they do not share a key). The formula for ρ_1 and ρ_2 are provided in (Lee and Stinson, 2005, Section VI), which are being formally restated in the following theorem. Let τ denote the number of nodes in the intersection of the physical neighborhood of two given nodes.

Theorem 4. *The value of the matric connectivities of a $\mu(v,b,r,k)-CID$ are: (i)* ρ_1=k × $\left(\frac{r-1}{b-1}\right)$; and

$$(ii)\ \rho_2 = (1-\rho_1) \times \left(1 - \left(\frac{b-\mu-2}{b-2}\right)^{\tau}\right).$$

The following corollary is an immediate outcome for our scheme by substituting the values of *b, r, k* and λ in Theorem 4.

Corollary 2. *The value of the connectivity metrics for the set system (X,A), which is a $(s-1)p^m-(pn, spm, s, pm)-CID$ are* $\rho_1 \approx 1-\frac{1}{s}$ and $\rho_2 = \frac{s^{\tau}-1}{s^{\tau+1}}$.

The metric $\rho_1 \approx 1-\frac{1}{s}$, i.e., connectivity increases if the size of spread increases. Here, the size of base field (i.e., the value of the prime *p*) has no significant effect on (direct) connectivity. As an example, if $n = 10$, $p = 2$ (i.e., there are $2^{10} \approx 1000$ many nodes) and s=25 then the value of $\rho_1 = 1-2^{-5}$.

9.5. Comparative Study

This section presents a comparative study of *our schemes (Sensornet* and *HC(Sensornet))* with existing works with respect to connectivity, resilience and network scaling. Performance of our schemes with respect to other prominent metric like storage, etc. has been discussed in previous section.

Schemes with high connectivity *(i.e., ρ_1)* and resiliency as small as possible are preferred. Unfortunately, both these metrics are inversely related to each other. So, it is a fundamental problem of trading off connectivity verses resiliency. The works [18, 29] considers the ratio $\rho = \frac{\rho_1}{fail(1)}$ for comparison of several combinatorial designs. Therefore, the larger ρ value confirms higher connectivity and lower resiliency. It is desirable that the ratio ρ be as large as possible for the basic combinatorial designs. If necessary, resilience improvement tricks like Bechkit et al. [5, 6] can applied like we did to construct *HC(Sensornet)* from our basic scheme, *Sensornet.*

There have been several proposals for deterministic key predistribution schemes for wireless sensor networks based on various types of combinatorial structures such as designs and codes. The paper [18] proposes a general framework by unifying those structures into a new design, termed as "*partially balanced t-designs (PBtD)*". Although, our scheme *Sensornet* falls into *$2-(v,\ k,\ \lambda_0=b,\ \lambda_1=r)-PBtD$* as a configuration, the generalization does not consider *$\mu-CID$s*. Hence, being a *$\mu-CID$*, *Sensornet* does not classify as PBtD by their description [18]. There are few comparison tables of different schemes are provided in [18]. In the following, we take data of *TD(t,k,Q)* with intersection threshold $\eta = 1$ from the paper [18] along with other designs to compare with the scheme *Sensornet.*

Let consider the number of nodes in all the compared scheme is N. Now we shall compare the asymptotic behavior of metrics ρ_1, *fail(1)* and the ratio ρ. The comparison is displayed in Table 1.

Table 1. Comparison of asymptotic behavior of different schemes. Refer to Remark 1 for discussions about the parameter $\boldsymbol{\rho}$.

Schemes	**Number of nodes ($\mathcal{N}$)**	$\boldsymbol{\rho_1}$	***fail(1)***	$\boldsymbol{\rho = \frac{\rho_1}{fail(1)}}$
(Sensornet) (devised in Section 5)	$\mathcal{N} = sp^m$	$1-\frac{1}{s}$	$p^{-m} = \mathcal{N}^{-1/2}$	$\left(1-\frac{1}{s}\right)\mathcal{N}^{1/2}$
HC(Sensornet) (devised in Section 8)	$\mathcal{N} = sp^m$	$1-\frac{1}{s}$	$p^{-m} = \mathcal{N}^{-1/2}$	$\left(1-\frac{1}{s}\right)\mathcal{N}^{1/2}$
TD(2; k; q); k = cq; [18]	$\mathcal{N} = q^2$	C	$q^{-1} = \mathcal{N}^{-1/2}$	$c\mathcal{N}^{1/2}$
TD(3; *k; q*); *k = cq; c* < 1	$\mathcal{N} = q^3$	$\frac{c(2-c)}{2}$	$\frac{2(1-c)}{2-c}\mathcal{N}^{-1/3}$	$\frac{c(2-c)^2}{4(1-c)}\mathcal{N}^{1/3}$
TD(3; k; q); k = q	$\mathcal{N} = q^3$	$\frac{1}{2}$	$5\mathcal{N}^{-2/3}$	$\frac{1}{10}\mathcal{N}^{2/3}$
TD(4; k; q); k = cq; c > 1	$\mathcal{N}=q^4$	$\frac{c(c^2-3c+6)}{6}$	$\frac{3(c^2-2c+2)}{c^2-3c+6}\mathcal{N}^{-1/4}$	$\frac{c(c^2-3c+6)^2}{18(c^2-2c+2)}\mathcal{N}^{1/4}$
Symmetric BIBD [9]	$\mathcal{N}=q^2+q^2+1$	1	$\mathcal{N}^{-1/2}$	$\mathcal{N}^{1/2}$
RS code based [4]	$\mathcal{N}=q^2$	$\frac{q-1}{q+1}$	$\mathcal{N}^{-1/2}$	$\mathcal{N}^{1/2}$
MB designs [15] over TD(k,q) KPS [11, 12] or RS code KPS [4]	$\mathcal{N} = \frac{q^2}{2}$	1	$(2\mathcal{N})^{-1/2}$	$(2\mathcal{N})^{1/2}$

Remark 1. For parity with the existing works that we consider in in Table 1, stated *r* values are independent of "smart attack" removal technique devised in Section 6. In case, this novel technique is applied, the *l* LEAD nodes gets $g+l-1$ extra keys. So their memory is a bit strained. However Moore's law ascertains that memory expansion with time is easy as compared to other hardware. Therefore this excess memory overload *(O(k),k : original key rings sizes)* is reasonable to assume in front of existing computation and transreceiver overheads.

From this comparison table it is clear that the asymptotic behavior of the ratio r of *Sensornet* and *HC(Sensornet)* is similar or better than all other schemes except the scheme $TD(3;k;q);k=q$ and Merging Block (MB) design of [14, 15].

Former scheme needs computation of number theoretic problems during key agreement; while the later supports significantly less (merged) blocks. Moreover shared key discovery in *Sensornet* scheme requires $O((log_p\ \mathcal{N})^3)$ time complexity and the amount of broadcast data is $O((log_p\ \mathcal{N})^2)$. Broadcast data in both *Sensornet* and *HC(Sensornet)* is $O(log_p\sqrt{\mathcal{N}}$ since transfer of basis vectors not required. This is advantageous over many KPS that require more data broadcast and complex establishment mechanism.

9.5.2. Scalability Comparison

Sensornet and thus *HC(Sensornet)* can support large networks. This is because the choice *n* and respectively *m* and/or *s* are unbounded in theory. Thus networks designed by our schemes are scalable.

Scalability is a major challenge in most deterministic KPS. For instance, the schemes [4, 9-12] have restricted scaling. This owes to the fact that key establishment for these network require general solutions of polynomials. Therefore, the complexity of the key establishment process increases with increment in degree of these polynomials. Random schemes can scale arbitrarily [19]; but at the expense of desirable parameters like connectivity, resilience, storage (key-node ratio), etc. Therefore, we opt deterministic schemes while designing KPS [18]. Also refer to Section 3.3.

10. Conclusion

Realizing the need of deterministic KPS with desirable properties (set out in Section 3.4) to address the problem of key management in low cost networks, we propose one such scheme. Since the scheme is constructed using nets in a vector space, we named it as *Sensornet*. The scheme is later improved using Hash Chains trick of Bechkit et al. [5,6] to obtain *HC(Sensornet)*, a resilience improved version. Key establishment of both *Sensornet* and *HC(Sensornet)* is a great advantage over other schemes, but still exploits the network's key sharing graph. This is overcome by a generic approach devised in Section 6 that removes "smart attacks". Resilience of *Sensornet* can be improved by a Hash Chains approach (along with the above approach) to yield *HC(Sensornet)*.

11. Related Future Works

Although both or schemes, *Sensornet* and *HC(Sensornet)* suffer from lack of full connectivity, it is very close to full connectivity for large size of partial spread. Moreover, the generic computations in Section 9.4 establish that connectivity of *Sensornet* is good (either direct or 1–hop path). It is preferable to have full connectivity or at least a deterministic path in case of 1–hop connectivity. The sophisticated MB designs of [14, 15] establishes a deterministic 1– hop connectivity for the Reed Solomon code based KPS (Ruj and Roy [4]). These heavily design dependent works can certainly open the doors for future research by considering similar constructions over *Sensornet* in place of other combinatorial design based schemes.

Efficient deterministic protocols for security applications in low cost IoT networks have restricted scaling. We adopt them due to their predictable connectivity and resilience. Scalable deterministic security protocols with flat topology using SKC techniques is a major area of study. Our "smart attack" removal approach may give interesting leads.

References

[1]. Dalai, D. K. and Sarkar, P., Sensornet: A key predistribution scheme for distributed sensors using nets, in *Proceedings of the 6th International Conference on Sensor Networks SENSORNETS*, Porto, Portugal, Vol. 1, 2017, pp. 49-58.

[2]. Zhu, S., Setia, S., and Jajodia, S., LEAP - efficient security mechanisms for largescale distributed sensor networks, in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, (SenSys'03)*, Los Angeles, California, USA, November 5-7, 2003, pp. 308–309.

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

[4]. Ruj, S. and Roy, B. K., Key predistribution schemes using codes in wireless sensor networks, in Information Security and Cryptology, Inscrypt 2008, *Lecture Notes in Computer Science*, 5487, pp. 275–288.

[5]. Bechkit, W., Bouabdallah, A., and Challal, Y., Enhancing resilience of probabilistic key pre-distribution schemes for WSNs through hash chaining, in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS'10),* Chicago, Illinois, USA, October 4-8, 2010, pp. 642–644.

[6]. Bechkit, W., Challal, Y., and Bouabdallah, A., A new class of hash-chains based key pre-distribution schemes for WSN, *Computer Communications*, 36, 3, 2013,
pp. 243–255.

[7]. Pietro, R. D., Mancini, L. V., and Mei, A., Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks, *Wireless Networks,* 12, 6, 2006, pp. 709–721.

[8]. Bose, M., Dey, A., and Mukerjee, R., Key predistribution schemes for distributed sensor networks via block designs, *Design, Codes and Cryptography,* 67, 1, 2013, pp. 111–136.

[9]. Camtepe, S. A. and Yener, B., Combinatorial design of key distribution mechanisms for wireless sensor networks, *IEEE/ACM Transactions on Networking,* Vol. 15, Issue 2, April 2007, pp. 346 - 358.

[10]. Lee, J. and Stinson, D. R., Deterministic key predistribution schemes for distributed sensor networks, in *Selected Areas in Cryptography (SAC'04), Lecture Notes in Computer Science*, 3357, 2004, pp. 294–307.

[11]. Lee, J. and Stinson, D. R., A combinatorial approach to key predistribution for distributed sensor networks, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'05),* 2005, pp. 1200-1205.

[12]. Lee, J. and Stinson, D. R., On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs, *ACM Transactions on Information and System Security,* 11, 2, 2008, pp. 854–867.

[13]. Wei, R. and Wu, J., Product construction of key distribution schemes for sensor networks, in *Selected Areas in Cryptography (SAC) 2004, Lecture Notes in Computer Science,* 3357, 2004, pp. 280–293.

[14]. Bag, S., Dhar, A., and Sarkar, P., 100 % connectivity for location aware code based KPD in clustered WSN: Merging blocks, in *Proceedings of the Information Security Conference (ISC'12), Lecture Notes in Computer Science,* 7483, 2012, pp. 136–150.

[15]. Sarkar, P., Rai, B. K., and Dhar, A., Connecting, scaling and securing RS code and TD based KPDs in WSNs: deterministic merging, in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '13),* Bangalore, India, July 29 - August 01, 2013, pp. 301–304.

[16]. Erdos, P. and Renyi, A., On the evolution of random graphs, *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 1960.

[17]. Blom, R., An optimal class of symmetric key generation systems, in *Advances in Cryptology - Eurocrypt 1984, Lecture Notes in Computer Science,* 209, 1985, pp. 335–338.

[18]. Paterson, M. B. and Stinson, D. R., A unified approach to combinatorial key predistribution schemes for sensor networks, *Design, Codes and Cryptography,* 71, 3, 2014, pp. 433-457.

[19]. Ruj, S. and Pal, A., Preferential attachment model with degree bound and its application to key predistribution in WSN, in *Proceedings of the IEEE Conference on Advanced Information Networking and Applications (AINA'16),* 2016, pp. 677–683.

[20]. Henry, K. J., Paterson, M. B., and Stinson, D. R., Practical approaches to varying network size in combinatorial key predistribution schemes, in *Selected Areas in Cryptography (SAC) 2013, Lecture Notes in Computer Science,* 8282, 2014, pp. 89-117.

[21]. Lidl, R. and Niederreiter, H., Finite fields. Encyclopaedia of mathematics and its applications. *Cambridge University Press*, 1997.

[22]. Eisfeld, J. and Storme, L., (partial) t-spreads and minimal t-covers in finite projective spaces, in Lecture notes for the Socrates Intensive Course on Finite Geometry and its Applications, *University of Ghent*, 2000.

[23]. Lu, H. Y., Partial spreads and hyperbent functions in odd characteristic, Master's Thesis, *Simon Fraser University*, 2008.

[24]. Johnson, N. L., Combinatorics of Spreads and Parallelisms, *CRC Press*, 2010.

[25]. Johnson, N. L., Jha, V., and Biliotti, M., Handbook of Finite Translation Planes, volume 289 of Pure and Applied Mathematics, *Chapman & Hall/CRC*, 2007.

[26]. Bu, T., Partitions of a vector space, *Discrete Mathematics*, 31, 1980, pp. 79–83.

[27]. Wu, B., Ps bent functions constructed from finite prequasifield spreads, 2013. Available at http://arxiv.org/pdf/1308.3355.pdf

[28]. Çeşmelioğlu A., Meidl, W., and Pott, A., Bent functions, spreads, and o-polynomials, *SIAM Journal of Discrete Mathematics,* 29, 2, 2015, pp. 854–867.

[29]. Dong, J. W., Pei, D. Y., and Wang, X. L., A class of key predistribution schemes based on orthogonal arrays, *Journal of Computer Science and Technology*, 23, 2011, pp. 825–831.

Published by International Frequency Sensor Association (IFSA) Publishing, S. L., 2017
(http://www.sensorsportal.com).