Scaling Laws of Key Pre-distribution Protocols in Wireless Sensor Networks

Wenjun Gu, Sriram Chellappan and Xiaole Bai

Abstract—Many key pre-distribution (KP) protocols have been proposed and well accepted in randomly deployed wireless sensor networks (WSNs). Being distributed and localized, they are perceived to be scalable as node density and network dimension increase. While it is true in terms of communication/computation overhead, their scalability in terms of security performance is unclear. In this paper, we conduct a detailed study on this issue. In particular, we define a new metric called Resilient Connectivity (RC) to quantify security performance in WSNs. We then conduct a detailed analytical investigation on how KP protocols scale with respect to node density and network dimension in terms of RC in randomly deployed WSNs. Based on our theoretical analysis, we state two scaling laws of KP protocols. Our first scaling law states that KP protocols are not scalable in terms of RC with respect to node density. Our second scaling law states that KP protocols are not scalable in terms of RC with respect to network dimension. In order to deal with the un-scalability of the above two scaling laws, we further propose logical and physical group deployment respectively. We validate our findings further using extensive numerical analysis and simulations.

Index Terms—Sensor Networks, Information Security, Key Management, Resilience, Scalability

I. INTRODUCTION

Many applications for Wireless Sensor Networks (WSNs) are envisaged in military, mission-critical and hostile environments. In such applications securing sensor communications from attackers is critical. The standard approach is to establish secure pairwise keys between communicating sensors. However, in many WSNs, deployment cannot be accurately determined or controlled, sensors have energy/storage constraints, are easier to be captured etc. These features make key management challenging in WSNs. They also limit the applicability of traditional schemes like centralized key distribution center (large messaging overhead), installing a single master key to all nodes (poor resilience) etc. for WSNs. The disadvantage of using public key cryptography based schemes for WSNs is the significant computational overhead.

Currently, the well accepted approach for key management in WSNs is based on the idea of key pre-distribution. In the simplest version, each sensor is pre-distributed with $k$ distinct keys randomly chosen from a large pool of $K$ keys, and the nodes are deployed randomly in the network. After deployment, neighboring nodes use the pre-distributed keys to establish a pairwise key in between either directly or using other nodes as proxies. The basic redundancy in initial key pre-distribution ($k$ keys per sensor) enables nodes to overcome deployment randomness, making it easier to discover secure neighbors and proxies. A host of key management protocol variants have been proposed based on key pre-distribution (called KP protocols) [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] etc., each one improving upon one or more features like connectivity, resilience, overhead etc. We point out that while each KP protocol variant is different from the other in terms of certain parameters and features (discussed in detail later), the core idea of pre-distributing keys to sensors and pairwise key establishment among the sensors is the same in all protocol variants.

Motivation: An important requirement of key management protocols in WSNs is scalability. Many security aware WSN applications are envisaged today where thousands of nodes are deployed [19]. There are two key properties that determine network size in WSNs: node density (average number of neighbors per node) and network dimension (geographical size of the network). Being purely distributed and localized, the communication and computational overhead increase in KP protocols is mild when node density increases. With higher node density, it is also believed that security performance improves, as nodes can now find more secure neighbors and proxies. In fact, many existing KP protocols assume a very high node density ($\geq 20$) neighbors per node) with the notion that it enhances performance [3], [4], [5]. From the perspective of network dimension also, KP protocols are generally believed to be scalable.

In the context of secure communications however, the WSN is under attacks. In such situation, while an increase in number of nodes helps the network side, it also enables malicious attackers to capture more nodes (to disclose more keys) and monitor more links in the network. Consequently, there is a tug of war between the network and attackers in terms of how extra nodes (and stored keys) are leveraged by each other. Because of this war between two conflicting entities, the scalability of KP protocols in terms of security performance has not yet been comprehensively determined. In this paper, we address this issue.

Our Contributions: We have the following four contributions in this paper.

- We define a new metric called Resilient Connectivity (RC) to quantify security performance for all KP protocols. Formally, $RC$ is the probability that two neighboring nodes can establish a secure pairwise key between...
them under attacks. This metric naturally considers both connectivity and resilience, two standard metrics used to evaluate security performance of KP protocols in previous literatures. To the best of our knowledge, ours is the first paper to propose a unified metric to evaluate the security performance of KP protocols.

- We conduct a comprehensive survey on state-of-the-art KP protocols, and make a detailed classification of all these protocols. In particular, we first identify and abstract the parameters that capture all the features that impact RC in any KP protocol. Then we classify all KP protocols based on different particular instances of these parameters, and derive the expressions for all the parameters in all the KP protocols.

- We rigorously derive a general expression for RC as the basis of our scalability study. Specifically, based on the derivations of the parameters that impact RC in all KP parameters, we are able to obtain a general form expression of RC as a function of all these parameters. Such general expression allows us to study KP protocol scalability efficiently, and draw conclusions that are general for all protocols. Our analysis has other merit in that this is the first time a general analysis of link resilience is given.

- We conduct a detailed analytical investigation on how KP protocols scale with respect to node density and network dimension in terms of RC in randomly deployed WSNs. Based on our theoretical analysis, we state two scaling laws for security performance of KP protocols. Our first scaling law states that KP protocols are not scalable in terms of RC with respect to node density. Our second scaling law states that KP protocols are not scalable in terms of RC with respect to network dimension. We conduct extensive numerical analysis and simulations to further validate our results. In fact, our data show that for reasonable network, protocol and attack parameters, RC starts to monotonically decrease from node densities around 20, and tends to zero quickly after that. Our data also demonstrate that RC rapidly tends to zero even for small values of network dimension (around 500m).

- Finally, we propose two types of group deployment to deal with the un-scalability of the above two scaling laws. In particular, we propose logical group deployment to deal with the un-scalability brought by node density, in which sensors are deployed in multiple rounds over the whole network. Since sensors in different rounds are pre-distributed with keys from disjoint key pools, the un-scalability issue brought by high node density is resolved. On the other hand, we propose physical group deployment to deal with the un-scalability brought by network dimension, in which sensors are deployed in groups over different areas of the network. Since sensors in different groups are pre-distributed with keys from partially overlapped or disjoint key pools, the un-scalability issue brought by network dimension is resolved.

We believe that our findings are fundamental and identify inherent limitations in key management in randomly deployed WSNs. We show that care should be taken during resource provisioning for secure WSNs. While focusing on protocol scalability in terms of overhead is important, we show it is equally (if not more) critical to also consider scalability in terms of security performance. Our work has quantitative significance too. When deployers may have a priori knowledge on attack intensities based on historical experience, our closed form expressions in this paper can be complemented with existing tools to derive optimum node densities and network dimensions for best performance.

The rest of our paper is organized as follows. In Section II, we present background on KP protocols, their variants, attack models and performance metrics. In Section III, we present analysis on the security performance of KP protocols. In Sections IV and V, we study how the KP protocols scale with respect to node density and network dimension respectively, including both theoretical analysis and simulation data. In Section VI, we propose two types of group deployment to deal with the un-scalability brought by the two scaling laws. We present related work and discussions in Section VII and conclude our paper in Section VIII.

II. KEY PRE-DISTRIBUTION PROTOCOLS

In this section, we provide a background on Key Pre-distribution (KP) based protocols, attack models and performance metrics for secure communications in WSNs. We also classify the KP protocols based on several features. We point out that the issue of securing sensor communications has been an important topic of research in WSNs. However, as discussed in Section I, key management is challenging in WSNs. Using Key Distribution Center (KDC) based scheme [20] to dynamically assign keys is typically centralized and not scalable for large scale multi-hop WSNs due to significant communication overhead. Since placement of sensors (and their neighborhood information) in a field cannot always be determined or controlled, pre-determining neighbors and assigning pairwise keys between them is not always feasible. Installing a single master key (or single key structure [21], [22]) for all sensors is too vulnerable under node captures, and pre-distributing a unique pairwise key for any two sensors incurs too much storage overhead. Public key cryptography based schemes while solving the above challenges incur too much computational overhead. Recently however, they are receiving attention [1], [2] due to advances in sensor node hardware.

A. Basic KP Protocol

1) Protocol Description: Fundamentally, all the above schemes suffer from issues related to scalability, resilience, storage overhead and energy consumption. In order to address these challenges, the seminal approach of key pre-distribution was first proposed in [3]. The core idea is to provision a certain
degree of redundancy in key sharing among nodes before deployment. After deployment, neighboring nodes leverage this redundancy to establish pairwise keys between them.

There are two stages in this protocol. At the key setup stage, each node is pre-distributed with \( k \) distinct keys randomly chosen from a large pool of \( K \) keys, and nodes are deployed randomly in the network. We point out that the pre-distributed keys are typically not deleted after protocol execution [3] [4] [5] [6]. They will be used for pairwise key establishment during later node additions due to faults, failures etc. Fig. 1 shows a deployment instance of 10 nodes, where \( k = 3 \) and \( K = 9 \). Nodes inside the circle are within the communication range of node \( a \). The pre-distributed keys for these nodes are also shown in Fig. 1. A list of basic parameters in the KP protocol and their notations are presented in Table 1.

At the pairwise key establishment stage, neighboring nodes try to establish a pairwise key in between using pre-distributed keys. First, each node obtains neighborhood key sharing information in its information area. The information area for a node is the area within which the node is aware of information on other nodes and their pre-distributed keys. We denote this parameter as \( A \). For instance, the information area for node \( a \) in Fig. 1 is the area in its one hop communication range. If two neighbors already share a pre-distributed key (e.g., nodes \( a \) and \( b \) that share key \( k_3 \)), they can establish a pairwise key directly. To do so, node \( a \) generates a random pairwise key and sends it to node \( b \) encrypted with key \( k_3 \). However, two physical neighbors may not always share a pre-distributed key due to randomness in key pre-distribution and deployment. In such cases, the nodes will use proxies to construct key paths for pairwise key establishment. A random key share is transmitted on each key path, and is encrypted/decrypted hop by hop. The pairwise key is a combination (e.g., bitwise XOR) of all the key shares. For example, nodes \( a \) and \( f \) can use node \( b \) as a proxy to construct a key path \( a \rightarrow b \rightarrow f \) (since nodes \( b \) and \( f \) share keys \( k_4 \) and \( k_7 \) and are physical neighbors). Note that nodes \( a \) and \( c \) cannot establish a pairwise key between them. This is because they do not share any pre-distributed key and cannot find any proxy in the above case. Finally, pairwise keys are used to encrypt future communications between neighboring nodes.

2) Attack Models: The standard attack model used in WSNs is one where the attacker attempts to decipher sensor communications [3] [4] [5] [6] etc. As such, the attacker will launch two types of attacks. In node capture attack, the attacker physically captures a certain percent of nodes, and disclose their pre-distributed and pairwise keys. The probability of a node to be captured is denoted as \( P_c \). In link monitor attack, the attacker monitors information on all network links immediately after deployment. Clearly, all communications to and from captured nodes are deciphered by the attacker. Furthermore, by combining the disclosed pre-distributed keys and messages recorded, the attacker can infer some pairwise keys between uncaptured nodes. For instance in Fig. 1, by capturing node \( g \), the attacker obtains key \( k_2 \), and automatically disclose the pairwise key between nodes \( a \) and \( e \) (without capturing either node), since the communication for establishing the pairwise key between nodes \( a \) and \( e \) is encrypted by \( k_2 \).

When multiple key paths are used to establish a pairwise key, the pairwise key is not disclosed unless all the key paths are compromised. A key path is compromised if one node on the path is captured or one link on the path is compromised. A link between two adjacent nodes on the key path is compromised if all shared pre-distributed keys between those two nodes are disclosed. An uncompromised key path is called as a secure key path, and an undisclosed pairwise key is called as a secure pairwise key. As we can expect, using multiple key paths to establish a pairwise key results in much higher resilience than using only one key path under attack. This is simply because the chance of attacker compromising all key paths decreases sharply with the number of key paths used. Although the basic scheme in [3] uses only one key path for pairwise key establishment, in the remaining of the paper, we assume all schemes use multiple key paths for pairwise key establishment and conduct our analysis based on this. This assumption is beneficial only to the network side and does not affect our conclusion in this paper. For similar reason, we also assume all shared keys on a link are used for pairwise key establishment instead of one of the shared keys as used in the basic scheme in [3]. We emphasize that the above attack model is the de-facto one used in many key management works.

3) Performance Metrics: To evaluate performance of KP protocols, two standard metrics are used: Connectivity and Resilience. Connectivity is the probability that two physical neighbors can establish a pairwise key in between. While the above definition refers to local connectivity and is standard, one could also define global connectivity as the probability that the entire network is securely connected, or as the percent of nodes in the largest connected component of the secure network. Since either definition of global connectivity is related with local connectivity [24], we only focus on local connectivity (henceforth called connectivity) in this paper. The other metric is resilience, which is the conditional probability that the pairwise key between two physically neighboring nodes is not disclosed to the attacker given that such pairwise key exists between those two nodes. In other words, in computing resilience, we only consider those links that have pairwise keys established. The effect of links that cannot establish pairwise keys is considered in the metric of connectivity above. The overall goal of any key management protocol is to achieve high connectivity and resilience.

B. KP Protocol Variants

In the above, we described the basic KP protocol in [3]. A host of KP protocol variants have been proposed to enhance the basic protocol across several features. However, the core idea of two stages, namely, key pre-distribution to sensors followed by pairwise key establishment among sensors is the same for all these protocols. In this section, we will describe these KP protocol variants based on the enhancement of the features in these two stages. A detailed classification KP 2

\(^2\)While some works like [23] assume a safe period (no node captures) after deployment, this assumption may not be always realistic in practice, and this attack model is not widely adopted.
protocols and the features (and corresponding specifics) that have been extended in those protocols are presented in Table II.

1) Enhancement in Key Setup Stage: The first feature is the nature of the pre-distributed keys. In the basic protocol [3], random keys are pre-distributed. In the KP protocol variant in [4], unique pairwise keys are distributed into pairs of sensors chosen randomly. Resilience is enhanced at the cost of poor connectivity in large scale networks under memory constraints. Works in [5] and [6] extend traditional crypto ideas in [21] and [22] respectively to distribute key structures (polynomials or matrix/vectors) instead of keys into sensors to enhance the resilience under low attack intensity.

The second feature is the method in pre-distributing the keys. In the basic protocol [3], keys are distributed randomly. In [15], keys are distributed according to some well known optimization designs, which helps increase chances of key sharing between nodes. In [6], quorum based methods are introduced to guarantee the existence of a key path between any two nodes. In [7], nodes are deployed into grids, and keys distributed in non-adjacent grids are disjoint, while keys pre-distributed in adjacent grids have a certain degree of overlap. This helps enhance the chance that two nodes in adjacent grids share keys.

The third feature is knowledge of deployment location. The basic KP protocol [3] do not assume nodes’ deployment positions are known as a priori. In works like [7], certain deployment knowledge is assumed to be known as a priori such that keys can be distributed based on the location information to enhance the chance of key sharing between neighboring nodes.

2) Enhancement in Pairwise Key Establishment Stage: The first feature we discuss in the pairwise key establishment stage is the information area of each node. In the basic protocol [3], each node is aware of the node/key information in its communication range. Thus the information area is within one hop. In [4], this feature is extended in that nodes are allowed to obtain node/key information in multiple hops to alleviate key path construction. In [6], nodes are even allowed to construct a key path using a proxy anywhere in the network. In effect, the information area in protocols becomes the entire network.

The second feature is the link usability on a key path. In most works, a link between two nodes is usable in key path construction as long as there is at least one shared key between those two nodes. However, in [4], $q$–composite concept was introduced, which allows two neighboring nodes to use the link between them only if they share at least $q$ keys. The resilience under low attack intensity (small value of $P_e$) is enhanced at the cost of lower resilience under higher attack intensity.

The last feature is the number of hops allowed on a key path. In the basic protocol [3], a key path can have arbitrary number of hops. However, in [4] [6] [10] etc., there are certain bounds on the maximum number of hops on a key path. Clearly this feature will affect the number of key paths constructed, and consequently affect the resilience of the pairwise keys established. More hops imply better chances of key establishment at the cost of increased communication/computation overhead and vice versa.

III. DERIVATION OF RESILIENT CONNECTIVITY

In this section, we first introduce our security metric called Resilient Connectivity (RC), followed by its derivation. Before the derivation of RC, we identify and abstract all the parameters that impact RC in all KP protocols listed in Table III. We then derive a general expression for RC incorporating these parameters to analyze KP protocols scalability. This generalizes our findings to all KP protocols.

A. Preliminaries

The traditional metrics to evaluate the performance of KP protocols are connectivity and resilience. These two metrics are disjoint in the sense that connectivity itself measures only the probability that physical neighbors can establish pairwise keys, irrespective of how secure these keys are from being disclosed by the attacker. On the other hand, resilience itself measures only how secure the established pairwise keys between neighbors are from being disclosed by the attacker, irrespective of the probability of physical neighbors actually establishing pairwise keys between them.

In order to quantify security performance, we combine the above two metrics and define a new one called Resilient Connectivity (RC). Formally, $RC = Connectivity \times Resilience$. There exists a strong physical meaning for RC, which is the probability that two physically neighboring sensors can communicate securely (with a secure pairwise key) under attacks. RC naturally encompasses both connectivity and resilience, and is our metric to evaluate the KP protocols scalability in terms of security performance.

We discussed various KP protocols in Section II. As discussed in Section II, since the core idea of all KP protocols follow two stages (key setup stage and pairwise key establishment stage), all their enhancements can be captured using certain parameters. We now introduce these parameters and derive their expressions. They will be used in our analysis later when we derive the expression of RC. Generalization of these parameters during RC derivation naturally generalizes our analysis to all KP protocols.

- Key Setup Parameter ($P[E_{sk}^i], P_{dis}$): As described earlier, there are various natures in pre-distributed keys, such as, key structure distribution in [5] [6], unique pairwise key distribution in [4], optimization design based distribution in [15], and location aided key distribution in [7] [11]. They can all be captured by two parameters, $P[E_{sk}^i]$ and $P_{dis}$, which denote the probability that two neighboring sensors share exact $i$ keys (or key structures) and the probability a single key (or key structure) is disclosed to the attacker respectively. The former captures the positive side of key redundancy in that it reflects the chance of direct key sharing and the chance of nearby proxies being helpful. The latter captures the negative side of key redundancy in that it reflects the chance that keys are disclosed by the attacker. In other words, two protocols with the same $P[E_{sk}^i]$ and $P_{dis}$ will have the same security performance, irrespective of the nature of keys (or key structures) being pre-distributed.
• **Pairwise Key Establishment Parameters** \((A, H, q)\): The basic operation in pairwise key establishment is constructing key paths using the links with shared keys. Three parameters that naturally affect the security performance are: the amount of information obtained by each node for key paths construction, the longest key paths allowed to be constructed, and the minimum requirement for a link to be usable in key path construction. These three parameters can be captured by three parameters respectively: the size of information area \((A)\), the maximum number of hops on one key path \((H)\), and the minimum number of shared keys for a link to be usable \((q)\). Large value of \(A\) or \(H\) makes more key paths available at the cost of communication/computation overhead. Large value of \(q\) achieves better resilience at low attack intensity at the cost of poorer resilience at high attack intensity [4].

In Table III, we show the expressions of the above five parameters for various \(KP\) protocols. In Table III, *same grid*, *edge adj. grids* and *corner adj. grids* denote the case when two neighboring sensors are in the same grid, in two edge adjacent grids, and in two corner adjacent grids respectively. In Table III, \(l\) denotes the size of each grid, and \(F(N_1, N_2, p) = \sum_{i=0}^{N_1} \binom{N_i}{i} p^i (1-p)^{N_1-i}\). Due to space limitation, interested readers are referred to [25] for the detailed derivations. To summarize at this point, we have identified and abstracted all parameters that impact \(RC\) in all \(KP\) protocols using the above five parameters. These parameters will be used in our analysis of \(RC\) in the next section. Since \(RC\) is our performance metric for scalability analysis, our results can be generalized for any \(KP\) protocol by substituting appropriate expressions for these parameters for different protocols during the analysis.

### B. Derivation of \(RC\)

We now discuss the derivation of \(RC\) for any general form of the above parameters. Certain other parameters, which are impacted by the parameters presented in Table III, will be used to derive \(RC\) and their notations are presented in Table IV. In our analysis, the attack model is the one discussed in Section II-A2. Table V gives the sequence of formulas in deriving \(RC\). We present here a basic overview of the derivation process.

In Table V (1), \(P[E^A_{i,j}]\) is the probability that an arbitrary uncaptured node (say node \(a\)) cannot construct a secure key path to its uncaptured physical neighbor (say node \(b\)) within \(a\)'s information area \((A)\). \(P[E^A_{+b}\mid E^A_{-a}]\) is the probability that uncaptured node \(a\) cannot construct a secure key path to uncaptured node \(a\) within \(b\)'s information area given that node \(a\) cannot construct a secure key path to node \(b\) within \(a\)'s information area \((A)\). Consequently, \((1 - P[E^A_{i,j}] P[E^A_{+b}\mid E^A_{-a}])\) is the probability that two arbitrary uncaptured neighboring nodes \(a\) and \(b\) are able to construct a secure key path in between. This value, times the probability that nodes \(a\) and \(b\) are themselves not captured (i.e., \((1 - P_c)^2\)) is \(RC\).

We now demonstrate the relationship between \(RC\) derivation in Table V and the parameters given in Table III. In (2) in Table V, we need to derive the probability for any node to establish a secure key path to its neighbor. Clearly, this depends on the information area size \((A)\) and the maximum number of proxies allowed on a key path \((H)\). In a similar manner, \(A\) and \(H\) are also used in (3). In (4), we compute the probability that two uncaptured neighboring nodes have a direct secure key path (one hop) between them \((P[E^A_{i,j}])\). To do so, we need to compute the \(P[E^A_{i,j}]\) and the probability that all the \(i\) shared keys are disclosed \((P[E^A_{dis}])\) (which depends on \(q\) and \(P_{dis}\) given in (7)). When we compute the probability that two uncaptured neighboring nodes have an indirect secure key path via the help of proxies in (5) and (6), we need \(P[E^A_{i,j}]\), which as discussed above needs \(P[E^A_{i,k}]\) and \(P_{dis}\). Note that since we use these parameters in their general forms, our \(RC\) derivation is general for all \(KP\) protocols. For more detailed discussions on the derivation of formulas in Table V, please refer to [25].

**Remarks:** Note that \(RC = Connectivity \times Resilience\). While there have been prior works on analyzing connectivity [3], [5], [26], no rigorous analysis on resilience has been conducted, except on the expected percent of disclosed pre-distributed keys. The difficulty is due to significant complexities in considering the nodes/keys overlaps among multiple key paths. In this paper, we have rigorously derived \(RC\), and Connectivity can be derived in the same way as \(RC\) by just substituting \(P_c = 0\). If solely Resilience is interested, it can naturally be analyzed via Resilience = \(RC/Connectivity\). To the best of our knowledge, ours is the first work that enables Resilience of \(KP\) protocols to be analyzed.

### IV. Scaling Law One: Scalability with Respect to Node Density

#### A. The Scaling Law

We now present our first finding on the scalability of \(KP\) protocols with respect to node density in terms of \(RC\). Based on the derivation of \(RC\) in Section III-B, we can treat \(RC\) as a function of node density \(D\) (denoted as \(RC(D)\)), given other parameters fixed. We now have the following theorem:

**Theorem 1:** For any \(KP\) protocol \((E^{sk}, P_{dis}, A, H, q)\), network parameters \((S, r)\), attack intensity \((P_c > 0)\), \(D\) and \(RC\) are not scalable in terms of \(RC\). Based on various protocols fixed. We now have the following theorem:

**Theorem 1:** For any \(KP\) protocol \((E^{sk}, P_{dis}, A, H, q)\), network parameters \((S, r)\), attack intensity \((P_c > 0)\), \(D\) and \(RC\) are not scalable in terms of \(RC\). Based on various protocols fixed. We now have the following theorem:

\[\exists D_1, D_2 : D_1, D_2 \in (0, +\infty), D_1 > D_2 : RC(D_1) < RC(D_2); \lim_{D \to +\infty} RC(D) = 0 \text{ (Proof in [25])}\]

The first part of Theorem 1 states that for any non-zero node capture probability \(P_c\), performance of \(KP\) protocols does not always increase with node density \(D\). There exists densities \(D_1\) and \(D_2\), where \(RC\) at a smaller node density is higher than \(RC\) at larger node density for any protocol and network parameters. The second part of the theorem further states that \(RC \to 0\) when \(D \to \infty\). It implies there is a finite value of node density \(D\) to achieve optimal performance for any \(KP\) protocol. To conclude, \(KP\) protocols are not scalable with respect to node density in terms of security. Based on the theorem above, we have the first scaling law for \(KP\) protocols.

**Scaling Law 1:** \(KP\) protocols are not scalable in terms of \(RC\) with respect to node density.

The fundamental explanation for the unscalability of \(KP\) protocols stems from *redundancy* in key pre-distribution inherent in all \(KP\) protocols, and the presence of attacks. Each sensor is provisioned with multiple keys, and each key is usually shared by multiple sensors. Clearly, redundancy helps the
network side overcome deployment randomness to discover more secure neighbors and proxies. However, this redundancy can be a double-edged sword. Attackers can also leverage redundancy to disclose more keys and communications. This redundancy is further amplified when node density increases. When the node density \((D)\) increases, the number of nodes \((N)\) and the number of captured nodes \((N_c)\) increase. We can then show that \(P_{dis} \rightarrow 1\) (from the expressions for \(P_{dis}\) in Table III) when number of captured nodes increases. When \(P_{dis} \rightarrow 1\), we can see from equation (7) in Table V that \(P[E_i^{th}] \rightarrow 1\). Thus from equations (2) to (6), we can see that \(RC \rightarrow 0\). This conclusion holds for all \(KP\) schemes since all the expressions of \(P_{dis}\) in Table III approach 1 when one of the three parameters \((D, N, N_c)\) approaches infinity. As pointed above, this is due to redundancy in key pre-distribution, which when amplified, causes degradation in \(RC\).

B. Numerical Results

In the following, we conduct extensive numerical studies on the sensitivity of resilient connectivity \((RC)\) to node density \((D)\) for different protocol variants under varying node capture probabilities \((P_c)\), number of keys \((k, K)\), maximum number of hops allowed in a key path \((H)\), key structure degree \((\lambda)\) and probability of key sharing \((P[E_i^{th}])\). Furthermore, we also demonstrate the soundness of our analysis by comparing its fidelity with simulation data. Unless otherwise stated, the following are default values: \(D = 15, S = 1000m \times 1000m, r = 10m, P_c = 0.005, k = 100, K = 30000, \lambda = 0, A = \pi r^2, H = \infty, q = 1\). By default, keys are pre-distributed randomly [3] without deployment knowledge.

The first observation we make from Figs. 2 to 5 is that \(RC\) does not monotonically increase with \(D\). Secondly, in all figures there is a particular point in density, beyond which \(RC\) monotonically decreases. We denote this \(D\) as density threshold \(D_{th}\). In fact, \(D_{th}\) indicates the critical point at which the attacker defeats the network in the tug of war between them (discussed above) in terms of redundancy exploitation. As we discuss below, the result of this war \((D_{th})\) is different for different protocols and parameters.

In Fig. 2, we study \(RC\) vs. \(D\) under different \(P_c\). When \(P_c\) is large, \(RC\) decreases from lower values in density (smaller \(D_{th}\)). This is because a large \(P_c\) means a powerful attacker. Increasing density means the attacker can capture more nodes and disclose more keys. Consequently \(D_{th}\) is low (near zero) when \(P_c\) is large. However, when \(P_c\) decreases it implies a moderate attacker. Increasing density (up to a point) will better facilitate the network side, and \(D_{th}\) thus increases. For example, when \(P_c = 0.005\), \(RC\) increases up to \(D_{th} = 15\) before decreasing. When \(P_c\) decreases further, \(D_{th}\) increases. To demonstrate the soundness of our analysis, we report data comparing numerical and simulation data for the case of \(RC\) vs. \(P_c\) in Fig. 3 (other parameters are default). As we can see, the numerical data match very well with simulation data. We note that simulation data are lower than analysis data due to network boundary effect.

In Figs. 4 and 5, we study \(RC\) vs. \(D\) under different \(K\) and \(H\). When \(K\) and \(H\) are small, \(D_{th}\) is small. This is once again because attack impacts are stronger than network benefits leading to more pairwise keys disclosed under smaller \(K\) (key pool size) and smaller \(H\) (number of hops) even at low densities. Increasing density further will better facilitate the attacker. When \(K\) and \(H\) are large, the attacker effectiveness decreases, which increases \(D_{th}\). We point out that there is a relationship between \(K\) and \(k\) from the perspective of key disclosure. A small \(k\) means fewer keys are disclosed per node capture and vice versa. This effect is opposite to that of \(K\).

C. Discussions

We wish to emphasize here an important observation from the above figures. Note that \(RC\) monotonically increases up to \(D_{th}\), after which it monotonically decreases in all figures. Towards this extent, we state the following conjecture:

Conjecture: For any \(KP\) protocol \((E_i^{th}, P_{dis}, A, H, q)\), any network parameters \((S, r)\) and any attack intensity \((P_c > 0)\),

\[
\begin{align*}
(1) &\text{ there is one and only } D_{th} \in (0, +\infty) \text{ where } RC \text{ is maximum;} \\
(2) &\forall D_1, D_2 : D_1 > D_2 < D_{th} : RC(D_1) < RC(D_2); \\
(3) &\forall D_1, D_2 : D_1 > D_2 > D_{th} : RC(D_1) < RC(D_2).
\end{align*}
\]

A rigorous proof of this conjecture is still an open issue. Here we provide an informal argument. An increase in node density will be leveraged by both the network and attacker. From the perspective of \(RC\), it translates to improved connectivity or decreased resilience respectively. The overall impact to \(RC\) is contingent on which factor dominates this tug of war. Initially, increase in node density improves connectivity significantly, which increases \(RC\). Considering there is an upper bound on connectivity in the network (at most \(one\)), there is a point from which resilience degradation always dominates with increase in density, resulting in the density threshold \((D_{th})\) from which \(RC\) monotonically decreases. However as our data show, the value of \(D_{th}\) itself is sensitive to the protocol, attack and network parameters. Given all the parameters, we are able to determine the optimal node density to achieve maximum \(RC\) based on our analysis in Section III-B.

V. SCALING LAW TWO: SCALABILITY WITH RESPECT TO NETWORK DIMENSION

A. The Scaling Law

Based on our earlier derivation of \(RC\) in Section III-B, we see that \(RC\) is dependent on network dimension \(S\). In the following, we denote \(RC(S)\) as resilient connectivity for a network with dimension \(S\), with other parameters fixed. We now have the following theorem:
Theorem 2: For any KP protocol \((E^k, P_{dis}, A, H, q)\), network parameters \((D, r)\), attack intensity \((P_c > 0)\),
(1) \(\forall S_1, S_2 : S_1 > S_2, RC(S_1) < RC(S_2)\); 
(2) \(\lim_{S \to +\infty} RC(S) = 0\) (Proof in [25]).

Theorem 2 states that for any non-zero node capture probability \(P_c\), performance monotonically decreases (to 0) as network dimension increases for any protocol and network parameters. This demonstrates the unscalability of KP protocols in terms of security performance with respect to network dimension. Based on the theorem above, we have the second scaling law for KP protocols.

Scaling Law 2: KP protocols are not scalable in terms of RC with respect to network dimension.

When network dimension increases, the number of nodes increases. This increases the redundancy in key sharing among nodes leveraged by the attacker, which is the fundamental reason for the unscalability of KP protocols with respect to network dimension. Specifically, when the network dimension \((S)\) increases, the number of nodes \((N)\) and the number of captured nodes \((N_c)\) increase. Similar to the discussions in Section IV, \(P_{dis} \to 1\) when the number of captured nodes increases. This further results in the fact that \(P[\text{dis}] \to 1\) and \(RC \to 0\). This conclusion holds for all KP schemes since all the expressions of \(P_{dis}\) in Table III approach 1 when one of the three parameters \((S, N, N_c)\) approaches infinity.

B. Numerical Results

In the following, we conduct a numerical study on the sensitivity of resilient connectivity \((RC)\) to network dimension \((S = L \times L)\) under different density \(D\). Other parameters are set as default. In Fig. 6, we observe that RC monotonically decreases as \(L\) increases for all \(D\). We also see that density threshold \(D_{th}\) (discussed earlier) decreases as \(L\) increases. This is because when network dimension is larger, more nodes are captured, resulting in more powerful attack impacts (even at low densities). Consequently \(RC\) decreases from an early \(D_{th}\) as \(L\) increases and vice versa.

Due to space limitation, we do not show the sensitivity of RC to \(L\) under other network parameters (e.g., \(P_c, K, k\) and \(H\)). Basically, the impact of the above network parameters on RC here is similar to that we discussed in Section IV.

VI. GROUP DEPLOYMENT

In this section, we propose two types of group deployment, that are, logical group deployment and physical group deployment, to deal with the un-scalability of KP protocols with respect to node density and network dimension respectively.

A. Logical Group Deployment

As we discussed in Section IV, high node density could result in security degradation in KP protocols. This is because given network dimension, the average number of captured nodes increases with the node density. As more nodes become captured, the attacker is able to compromise a larger percentage of pre-distributed keys and compromise secure communications in the whole network to a larger extent.

Intuitively, it seems to be a dilemma to achieve high security performance in sensor networks requiring high node density. However, we find that the above two factors do not necessarily contradict with each other. High node density hurts secure communications not because there are too many nodes in a unit area, but because there are too many nodes in a unit area whose pre-distributed keys come from a single key pool. If we can maintain the number of nodes in a unit area, while decreasing the number of nodes sharing a single key pool in a unit area, we are able to achieve both high security and high performance simultaneously.

In this paper, we propose logical group deployment to achieve high node density without sacrificing secure communications. In particular, we deploy sensor nodes in multiple rounds. In each round, certain number of sensors are deployed to the whole network, and these sensors are pre-distributed with keys from the same key pool. On the other hand, sensors deployed in different rounds are pre-distributed with keys from disjoint key pools. In other words, sensors are deployed in multiple logical groups, any two of which share no pre-distributed keys. By deploying nodes in this way, we can achieve arbitrarily high node density (with multiple rounds), while at the same time achieve high security as security is decided by the node density in a single round.

Our logical group deployment resolves the dilemma between high node density and high security at the cost of two nodes in different rounds not being able to communicate with each other. This is because sensors in different rounds use disjoint key pools, and cannot establish a pair-wise key in between. However, this will not be an issue as long as the node density determined by each round is high enough to achieve node connectivity within each round with high probability. In many applications, a sensor does not need to communicate with all its neighbors. It suffices if each node can communicate with a few neighbors to achieve connectivity and redundancy. However, in the rare cases where nodes in different groups are required to communicate with each other, we can extend our logical group deployment as follows. We let any two key pools in two different groups to have a small percentage of overlap (e.g., 20%) so that sensors in different groups can still establish pair-wise keys in between. The overlap is small so that security will not be compromised much.

In Fig. 7, we study the sensitivity of RC to node density \((D)\) under different \(P_c\) for KP protocols for traditional one time deployment and our logical group deployment. The node density decided by all sensor nodes is \(D\), and other parameters are set as default. Under logical group deployment, we divide sensors in two rounds with the same size, and the two key pools in two rounds are disjoint. We can see that when node
density is high, $RC$ is better in logical group deployment compared to that of traditional one time deployment, especially when node capture probability is high. However, when node density is low, $RC$ in logical group deployment may even be worse than that in traditional one time deployment, especially when node capture probability is low. This is because under low node capture probability, the threshold density is high. $RC$ increases with node density, in which case logical group deployment should not be used. To sum up, our logical group deployment helps to enhance security performance when node density is high. This also justifies that the adoption of our logical group deployment under high node density will not hurt connectivity much as node density in each round is still high enough.

**B. Physical Group Deployment**

As we discussed in Section V, large network dimension could result in security degradation in $KP$ protocols. This is because given node density, the average number of captured nodes increases with the network dimension. The more nodes become captured, the larger percentage of pre-distributed keys the attacker can compromise and to a larger extent the attacker can compromise the secure communications in the network. Decreasing network dimension certainly helps, however, many sensor networks are envisaged to be deployed in large area, such as the battle field or the border between two adjacent countries.

Intuitively, it seems to be difficult to achieve high security performance in sensor networks with large network dimension. However, the above two factors do not necessarily contradict with each other either. Large network dimension hurts secure communications not simply because there are too many nodes in the whole network, but because there are too many nodes in the network whose pre-distributed keys come from a single key pool. If we can maintain the total number of nodes in the network, while decreasing the number of nodes sharing a single key pool in the network, we are able to achieve both high security and large network dimension simultaneously.

In this paper, we propose physical group deployment to achieve large network dimension without sacrificing secure communications. In particular, we deploy sensor nodes in multiple groups. In each group, certain number of sensors are deployed to a specific area of the network, and these sensors are pre-distributed with keys from the same key pool. On the other hand, sensors in different groups are deployed in different areas of the network, and are pre-distributed with keys from partially overlapped or disjoint key pools. A simple example is first dividing the network into multiple disjoint grids, and then deploying one group of sensors in each grid. By deploying nodes in this way, we can achieve arbitrarily large network dimension (with multiple groups), while at the same time achieve high security as security is mainly decided by the dimension of a single group.

In our physical group deployment, we assume the knowledge as to which sensor belongs to which group is known as a priori, which is a common assumption in $KP$ protocols [11] [28]. Sensors in adjacent groups share different key pools with limited overlap to facilitate neighbor group communications, while key pools of non-adjacent groups share no overlap. To derive $RC$ under group deployment for general $KP$ protocols, we point out that there are two types of relationships between two adjacent groups, i.e., edge adjacent or corner adjacent. The derivation of $RC$ under group deployment thus has to consider the key pool overlaps between adjacent edges, which is not the case in traditional one time random deployment discussed in Section III. The derivations of $RC$ under group deployment are not presented here due to space limitation.

In Fig. 8, we study the sensitivity of $RC$ to network dimension ($L$) under different $P_c$ for $KP$ protocols for traditional one time deployment and our physical group deployment. The overall network dimension is $L \times L$. Under physical group deployment, we divide the network into four groups, each has dimension $L_{grid} = L/2$. By default, we set the percent of key overlaps among edge adjacent and corner adjacent grids as $\alpha = 0.20$ and $\beta = 0.05$ respectively (as in [11]). We see that $RC$ is consistently better in physical group deployment compared to that of traditional one time deployment. Thus, physical group deployment always helps enhance security, which is a little different from the case in logical group deployment we discussed above.

**VII. RELATED WORK AND DISCUSSIONS**

Our discussions above focused on a broad spectrum of $KP$ protocols in terms of their security scalability. We now discuss other important related works in key pre-distribution tradeoff, sensor network scaling effects, and public key cryptography. Besides, we will discuss $KP$ protocols scalability under attack models less powerful than what was used in this paper so far.

In random key pre-distribution, an inherent tradeoff exists among connectivity, resilience and storage overhead [4]. Fundamentally, when too few keys are pre-distributed, nodes cannot establish pairwise keys with many neighbors so that connectivity is decreased. Pre-distributing too many keys on the other hand not only incurs more storage overhead, but also can disclose more keys to an attacker even if only few nodes are captured, thus compromising resilience. This tradeoff does not affect our results, since we focus on protocol scalability with respect to node density and network dimension.

There have been some works showing the negative effects of nodes redundancy in other aspects of WSN performance. In [29], it is shown that per node throughput asymptotically reaches zero as number of nodes increases due to contention issues. In [30], impacts of node number on collisions is investigated, and mechanisms are suggested to reduce collisions. We point out that our work is orthogonal to the above, in that we are focusing on the downsides of network scale in terms of security performance in WSNs, which to the best of our knowledge has not been addressed before.

Recently, the idea of public key cryptography has been proposed as an alternative to symmetric key schemes in wireless sensor networks [1], [2]. However, such schemes are still impractical in sensor networks, especially since sensors are severely energy constrained. In large scale networks where communication is based on multi-hops, performing encryption/
decryption of asymmetric keys can rapidly drain out sensor energies. Nevertheless, we believe that public key cryptography for energy constrained sensors is an interesting topic that will receive more attention in the near future and beyond.

In the attack model that we discussed in Section II-A2 and used for analysis in Section III-B, the average number of captured nodes increased linearly with total number of nodes in the network (when node capture probability $P_c$ is fixed), and the attacker was able to monitor all the links in the network. In practice, it may happen that attackers are less potent. Based on our analysis in Section III-B, we find that as long as the number of captured nodes increases (even sublinearly) with total number of nodes in the network and unbounded, both our scaling laws still hold in the monitored area. This is because as long as the attacker impacts increase with network scale, the redundancy in key pre-distribution will result in the attacker winning the tug of war with the network, although the node density and network dimension beyond which the protocols become unscalable increase under less powerful attacks.

VIII. Final Remarks

In this paper, we have conducted a detailed investigation on scalability of key pre-distribution (KP) protocols in randomly deployed WSNs. Contrary to common perceptions, we find that KP protocols are not scalable in terms of security performance with respect to node density and network dimension. The fundamental reason is due to redundancy in key pre-distribution among nodes, which, while helping the network overcome deployment randomness, can also be leveraged by attackers to disclose more keys, hence compromising security performance. While group deployment can improve protocols scalability to a certain extent, the basic scalability problem still persists here.

We believe that our work in this paper is fundamental, and identifies inherent limitations in key management in randomly deployed WSNs from the perspective of redundancy in resource (keys and nodes) provisioning. The significances of our work also extend to other network systems that utilize redundancy. In secure overlay forwarding systems [31], while redundancy in system connectivity enables clients to find more paths to the server, attackers can leverage high connectivity to disclose the server rapidly (and attack it). For file sharing systems [32], while popular content replication enhances load sharing among servers and faster service, it can be exploited by attackers to disrupt system quality by corrupting such popular files. We believe that our work here can be directly extended to such systems to understand their tradeoffs and provision resources carefully.

REFERENCES


Fig. 1. An initial deployment of sensors pre-distributed with keys.

### TABLE I

**PARAMETERS IN KP PROTOCOLS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>Network dimension</td>
</tr>
<tr>
<td>$S$</td>
<td>The area of network, $S = L \times L$</td>
</tr>
<tr>
<td>$D$</td>
<td>Node density (average number of nodes in communication disk)</td>
</tr>
<tr>
<td>$N$</td>
<td>The total number of nodes deployed, $N = DS/\pi r^2$</td>
</tr>
<tr>
<td>$r$</td>
<td>Communication range</td>
</tr>
<tr>
<td>$k$</td>
<td>The number of keys (key structures) distributed in a sensor</td>
</tr>
<tr>
<td>$K$</td>
<td>The total number of keys (key structures) in the pool</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>The degree of key structure</td>
</tr>
<tr>
<td>$A_c$</td>
<td>The size of information area for a node</td>
</tr>
<tr>
<td>$N_c$</td>
<td>The number of nodes in a given sensor’s information area</td>
</tr>
<tr>
<td>$H$</td>
<td>Maximum number of hops allowed on one key path</td>
</tr>
<tr>
<td>$q$</td>
<td>Minimum number of shared keys needed for a link to be usable</td>
</tr>
<tr>
<td>$P_c$</td>
<td>Probability for each node to be captured by the attacker</td>
</tr>
</tbody>
</table>

### TABLE IV

**NOTATIONS FOR RC DERIVATION**

<table>
<thead>
<tr>
<th>RC</th>
<th>Resilient Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E^{k}_{i}$</td>
<td>The event that two nodes share $i$ keys (key structures)</td>
</tr>
<tr>
<td>$P_{dis}$</td>
<td>The probability that a key (key structure) is disclosed</td>
</tr>
<tr>
<td>$A$</td>
<td>The size of information area for a node</td>
</tr>
<tr>
<td>$A_o$</td>
<td>The expected overlapping area created by two information areas of two neighboring nodes, $A_o = 0.5865A$ [4] when $A$ is a disk</td>
</tr>
<tr>
<td>$H$</td>
<td>The maximum number of hops allowed on one key path</td>
</tr>
<tr>
<td>$q$</td>
<td>The minimum number of shared keys for a usable link</td>
</tr>
<tr>
<td>$E^{A}_{a\rightarrow b}$</td>
<td>The event that, given two uncaptured neighboring nodes $a$ and $b$, node $a$ can construct one secure key path to node $b$ with all proxies in $a$’s information area $A$. $E^{A}_{a\rightarrow b}$ denotes its negative</td>
</tr>
<tr>
<td>$E^{A}_{b\rightarrow a}$</td>
<td>The event that, given two uncaptured neighboring nodes $a$ and $b$, node $b$ can construct one secure key path to node $a$ with all proxies in $b$’s information area $A$. $E^{A}_{b\rightarrow a}$ denotes its negative</td>
</tr>
<tr>
<td>$E^{A}_{a\leftrightarrow b}$</td>
<td>The event that, given two uncaptured neighboring nodes $a$ and $b$, node $a$ can construct one secure key path to node $b$ with minimum hops $i$ with all proxies in $a$’s information area $A$</td>
</tr>
<tr>
<td>$E^{i}_{dis}$</td>
<td>The event that all $i$ shared keys (key structures) between two nodes are disclosed to the attacker</td>
</tr>
<tr>
<td>$P[E]$</td>
<td>Probability of occurrence of event $E$</td>
</tr>
</tbody>
</table>

### APPENDIX


Fig. 2. $RC$ vs. $D$ under different $P_c$

Fig. 3. Comparison of analysis and simulation data

Fig. 4. $RC$ vs. $D$ under different $K$

Fig. 5. $RC$ vs. $D$ under different $H$

Fig. 6. $RC$ vs. $L$ under different $D$

Fig. 7. $RC$ vs. $D$ under logical group deployment with different $P_c$

Fig. 8. $RC$ vs. $L$ under physical group deployment with different $P_c$
### TABLE II
**Classification of KP Protocols**

<table>
<thead>
<tr>
<th>Protocol Stages</th>
<th>Protocol Features</th>
<th>Protocols</th>
<th>Specifics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>key setup</strong></td>
<td>random keys</td>
<td>basic protocol [3], q – composite [4], direct/cooperative [8], probabilistic [9], deployment knowledge [11], PIKE [12], RKEP [13], configuration/intersection [15]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unique pairwise keys</td>
<td>random pairwise key [4], closest pairwise key [7], GKE [16]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>random key structures</td>
<td>multi – space [5], random subset [6], grid based [6], location based [7], location aware [10], hexagonal grid [14], multivariate [17] [18]</td>
<td></td>
</tr>
<tr>
<td><strong>key distribution method</strong></td>
<td>random</td>
<td>basic protocol [3], q – composite [4], random pairwise key [4], multi – space [5], random subset [6], closest pairwise key [7], direct/cooperative [8], probabilistic [9], RKEP [13], multivariate [17] [18]</td>
<td>configuration/intersection [15]</td>
</tr>
<tr>
<td></td>
<td>optimization design</td>
<td>configuration/intersection [15]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>grid based</td>
<td>grid based [6], PIKE [12]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>aware</td>
<td>location based [7], location aware [10], deployment knowledge [11], hexagonal grid [14], GKE [16]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deployment knowledge</td>
<td>closest pairwise key/location based [7], location aware [10], deployment knowledge [11], hexagonal grid [14], GKE [16]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unaware</td>
<td>basic protocol [3], q – composite/random pairwise key [4], multi – space [5], random subset/grid based [6], direct/cooperative [8], probabilistic [9], PIKE [12], RKEP [13], configuration/intersection [15], multivariate [17] [18]</td>
<td></td>
</tr>
<tr>
<td><strong>information area</strong></td>
<td>within one hop</td>
<td>basic protocol [3], q – composite [4], random subset [6], closest pairwise key/location based [7], direct/cooperative [8], hexagonal grid [14], configuration [15], multivariate [17] [18]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>within multiple hops</td>
<td>random pairwise key [4], multi – space [5], probabilistic [9], location aware [10], deployment knowledge [11]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>entire network</td>
<td>grid based [6], PIKE [12], RKEP [13], configuration/intersection [15], GKE [16]</td>
<td></td>
</tr>
<tr>
<td><strong>pairwise key establishment</strong></td>
<td>minimum number of shared keys on a usable link</td>
<td>1</td>
<td>all other protocols</td>
</tr>
<tr>
<td></td>
<td>&gt; 1</td>
<td>q – composite [4]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>maximum number of hops on one key path</td>
<td>1</td>
<td>random pairwise key [4], direct [8], hexagonal grid [14], configuration [15]</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>grid based [6], closest pairwise key/location based [7], cooperative [8], PIKE [12], RKEP [13], configuration/intersection [15]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>location aware [10], GKE [16]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>∞</td>
<td>basic protocol [3], q – composite [4], multi – space [5], random subset [6], probabilistic [9], deployment knowledge [11], multivariate [17] [18]</td>
<td></td>
</tr>
</tbody>
</table>
TABLE III
PARAMETERS OF KP PROTOCOLS THAT AFFECT RC

<table>
<thead>
<tr>
<th>Protocols</th>
<th>$P[E^*_p]^2$</th>
<th>$P_{div}$</th>
<th>$A$</th>
<th>$H$</th>
<th>$q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic protocol [3]</td>
<td>$(H)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\infty$</td>
<td>$1$</td>
</tr>
<tr>
<td>q – composite [4]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\geq 1$</td>
<td></td>
</tr>
<tr>
<td>random pairwise key [4]</td>
<td>${1-\frac{k}{\pi}} i=0 {i=1 \ i&gt;1$</td>
<td>$1-(N_{P/D})/(N_{P})$</td>
<td>$\geq \pi^2$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td>multi – space [5]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\geq \pi^2$</td>
<td>$\infty$</td>
<td>$1$</td>
</tr>
<tr>
<td>random subset [6]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\infty$</td>
<td>$1$</td>
</tr>
<tr>
<td>grid based [8]</td>
<td>${1-\frac{2(N-1)}{k^2}} i=0 {i=1 \ i&gt;1$</td>
<td>$F(2\sqrt{N-1}+\lambda,1,\frac{k}{\pi})$</td>
<td>$S$</td>
<td>$2$</td>
<td>$1$</td>
</tr>
<tr>
<td>closest pairwise key [7]</td>
<td>${1-\frac{k}{\pi}} i=1 {i=0 \ i&gt;1$</td>
<td>$1-(N_{P}^2)/(N_{P})$</td>
<td>$\pi^2$</td>
<td>$2$</td>
<td>$1$</td>
</tr>
<tr>
<td>location based [7]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$2$</td>
<td>$1$</td>
</tr>
<tr>
<td>direct/cooperative [8]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$1,2,1$</td>
<td></td>
</tr>
<tr>
<td>probabilistic [9]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\geq \pi^2$</td>
<td>$1$</td>
</tr>
<tr>
<td>location aware [10]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\infty$</td>
<td>$1$</td>
</tr>
<tr>
<td>deployment knowledge [11]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\infty$</td>
<td>$1$</td>
</tr>
<tr>
<td>PIKE [12]</td>
<td>${1-\frac{2(N-1)}{k^2}} i=0 {i=1 \ i&gt;1$</td>
<td>$1-(N_{P}^2)/(N_{P})$</td>
<td>$S$</td>
<td>$2$</td>
<td>$1$</td>
</tr>
<tr>
<td>RKEP [13]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$S$</td>
<td>$2$</td>
<td>$1$</td>
</tr>
<tr>
<td>hexagonal grid [14]</td>
<td>depend on grid shape, refer to [14] for details</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$1,1$</td>
<td></td>
</tr>
<tr>
<td>configuration [15]</td>
<td>${0 \ i \neq 1$</td>
<td>$1-(N_{P}^2)/(N_{P})$</td>
<td>$\pi^2$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td>intersection [15]</td>
<td>${\frac{2(N-1)}{k^2}} i=1 {i \neq 1$</td>
<td>$1-(N_{P}^2)/(N_{P})$</td>
<td>$\pi^2$</td>
<td>$S$</td>
<td>$2$</td>
</tr>
<tr>
<td>GKE [16]</td>
<td>${\frac{k}{\pi}} i=1, same grid {i \neq 1$</td>
<td>$1-(N_{P}^2)/(N_{P})$</td>
<td>$\pi^2$</td>
<td>$S$</td>
<td>$3$</td>
</tr>
<tr>
<td>multivariate [17] [18]</td>
<td>$(K)_{2(k+1)}(k_F)^2/(k^2)$</td>
<td>$F(p_{div},\lambda+1,\frac{k}{\pi})$</td>
<td>$\pi^2$</td>
<td>$\infty$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

TABLE V
RC DERIVATION FORMULAS

$$RC = (1 - P_c)^2(1 - P[E^*_p]^2 P[E^*_p]/P[E^*_p])$$ (1)

where
$$P[E^*_p] = 1 - P[E^*_A] = 1 - \sum_{i=1}^{N} P[E^*_A]$$ (2)

$$P[E^*_p]/P[E^*_p] = 1 - \sum_{i=1}^{N} P[E^*_A] - \sum_{i=1}^{N} P[E^*_A]$$ (3)

$$P[E^*_p] = 1 - \sum_{i=1}^{N} (P[E^*_p]/P[E^*_p])$$ (4)

$$P[E^*_p] = (1 - P[E^*_A]) \sum_{i=0}^{N-2} F(N-2, N_e, \frac{N}{2}) \frac{N}{2} \sum_{i=1}^{N} (F(N_u, n_i, P_l) P_2(n_i))$$ (5)

$$P[E^*_p] = (1 - P[E^*_A]) \sum_{i=2}^{N-2} F(N-2, N_e, \frac{N}{2}) \frac{N}{2} \sum_{i=1}^{N} (F(N_u, n_i, P_l) (1 - A_0 P[E^*_A]^2))$$ (6)

$$P[E^*_p] = \sum_{i=0}^{K} F(K, m, A, n)^{m}_{k} i < q$$ (7)

$$F(N_1, N_2, P) = (\frac{N_1}{P} N_2 (1 - P))^N_1$$ (8)

$$H(i, j, N_e, n_i, n_j) = \sum_{i=1}^{N_{max}} P_2(n_i+1) H(i-j, N_e, n_i, \cdots, n_{j+1})$$ for $1 \leq j \leq i < 2, H(1, N_e, n_1, \cdots, n_{1}) = 1 - (1 - A_0 P[E^*_A]/A)^{n-1}$ (9)

$$P_1 = [P[E^*_A] (1 - P)]$$ (10)

$$P_2(n_i) = 1 - (1 - A_0 P[E^*_A]/A)^{n_i}$$ (11)

$$P_3(n_i) = F(N_e - \sum_{j=1}^{i-1} n_j, n_i, (1 - A_0 P[E^*_A]/A)^{n-1} - (1 - A_0 P[E^*_A]/A)^{n_i})$$ (12)

$$P_4(n_i) = F(N_e - \sum_{j=1}^{i} n_j, n_i, (1 - A_0 P[E^*_A]/A)^{n-1} - (1 - A_0 P[E^*_A]/A)^{n_i})$$ (13)