Optimally Increasing Secure Connectivity in Multi-hop Wireless Ad Hoc Networks

Seyit Ahmet Camtepe, Sahin Albayrak
Technische Universitaet Berlin, Berlin, Germany
Email: {ahmet.camtepe, sahin.albayrak}@dai-labor.de
Bülent Yener
Rensselaer Polytechnic Institute, Troy, NY
Email: yener@cs.rpi.edu

***Abstract*—We consider the problem of how to maximize secure connectivity of multi-hop wireless ad hoc networks after deployment. Two approaches, based on graph augmentation problems with nonlinear edge costs, are formulated. The first one is based on establishing a secret key using only the links that are already secured by secret keys. This problem is in NP-hard and does not accept polynomial time approximation scheme PTAS since minimum cutsets to be augmented do not admit constant costs. The second one is based of increasing the power level between a pair of nodes that has a secret key to enable them physically connect. This problem can be formulated as the optimal key establishment problem with interference constraints with bi-objectives: (i) maximizing the concurrent key establishment flow, (ii) minimizing the cost. We show that both problems are NP-hard and MAX-SNP (i.e., it is NP-hard to approximate them within a factor of $1+\epsilon$ for $\epsilon > 0$) with a reduction to MAX3SAT problem. Thus, we design and implement a fully distributed algorithm for authenticated key establishment in wireless sensor networks where each sensor knows only its one-hop neighborhood. Our *witness* based approaches find witnesses in multi-hop neighborhood to authenticate the key establishment between two sensor nodes which do not share a key and which are not connected through a secure path.**

I. Introduction

Efficient key management schemes are essential to ensure authentication, integrity and confidentiality in multi-hop wireless ad hoc networks. An example of such networks are multi-hop wireless sensor networks operating in adversarial conditions. Many different key management schemes are proposed for wireless sensor networks. Some solutions assign each node a key-chain, a set of symmetric keys or keying materials (e.g., ID, master keys, hash functions, pseudo random functions, shared polynomials, key matrices and location information), to be shared with *some* of its neighbors after deployment with high probability. Others are based on trusted entities (e.g., base stations, trusted nodes and certificate authorities) to establish symmetric or asymmetric keys between sensor nodes. The unique key-chain assigned to each node creates a binding between the identity of a node and its set of keys; thus, provides authentication which is limited by the resilience of the underlying key distribution scheme. A detailed comparative survey on wide range of key management schemes for such networks can be found in [1], [2].

Keys and keying materials may be pre-distributed to nodes during key pre-distribution phase in a central location. There has been a significant focus in designing probabilistic, deterministic and hybrid key management schemes to ensure that neighboring sensor nodes can find a key to secure their link with high probability. In *random key pre-distribution scheme* [3] and its variants [4], [5], [6], [7], [8], each node receives a key-chain which is *randomly* drawn from a pool. In [9], [10], [4] random and expander graphs are used to generate key-chains of dedicated pairwise keys for the nodes. In [11], [12], [13], [14], [15], [16], [8], [17], [18], *deterministic* techniques from algebra and design theory are used to generate a key-chain for each node. In [19], [20], [21], [22], [23] deployment knowledge, in [24], [25] master keys and in [14] node IDs are used to improve the probability of finding common key. Finally, there are promising attempts to use RSA, elliptic curve and ID-based cryptography on sensor networks [26], [27], [28], [29], [30].

In wireless sensor networks, sensor nodes are usually randomly scattered over a large application area which might be inaccessible or infeasible to access after the deployment. Even with controlled placement of sensor nodes, due to environmental challenges and deployment errors, post-deployment network configuration might be unknown a priori. After the deployment, each node discovers its neighbors and tries to find a symmetric key to secure its links in shared-key discovery phase. Majority of key management schemes can not guarantee a symmetric key to secure each link. Therefore, in the key establishment phase, each pair of neighboring nodes, which do not have common keys, establish one or more keys. Key establishment between two nodes can be achieved by using pre-distributed keying materials and by exchanging messages directly over their insecure wireless link or over one or more secure paths on which each link is secured with a symmetric key.

There is no one-size-fits-all solution and there are significant trade-offs among these solutions in terms of their scalability, probability of finding a key, resilience and overhead (communication, storage and processing). A set of deterministic solutions guarantee a key between each pair of nodes regardless of the underlying physical network topology. But, these solutions can not scale, provide low resilience, introduce high overhead or may not support node addition and deletions. Focus of this work is on multi-hop wireless ad hoc networks with key pre-distribution schemes in which not every link is protected. Thus,

secure connectivity cannot be ensured or the network cannot be utilized fully.

Utilization of multi-hop wireless networks is investigated as wireless scheduling problem which assigns transmission power levels to the network nodes and tries to schedule all the links in an arbitrary network topology. Scheduling complexity of arbitrary topologies in wireless networks in the context of physical Signal-to-Interference-plus-Noise-Ratio (SINR) has been investigated in [31], [32], [33], [34] and shown to be NP-complete in various formulations. Secure capacity of a randomly deployed network is analyzed in [35] where each node receives a key-chain due to *random key pre-distribution scheme* [3]. In [36] a framework is proposed to improve existing key pre-distribution schemes by assuming that sensors are deployed in groups and group members are located close to each other after deployment. Hence, more research attention is required on analyzing the complexity of increasing the secure connectivity and capacity in multi-hop wireless ad hoc networks after deployment.

Our Contribution:

Our first contribution is theoretical as we formulate the different variants of the problem and analyze their complexity. Our practical contribution is to provide distributed heuristic algorithms.

In particular, we present two approaches: (i) establish new symmetric keys on the existing physical links (problem **P1**), and (ii) establish new physical links by increasing transmission power to connect the nodes that they do share a symmetric key (problem **P2**). Both of the problems are variants of graph augmentation problem which are in general NP-hard for fixed cost functions and accept polynomial time constant approximation schemes (PTAS) [37]. However, our problems are more complex.

Problem **P1** is a variant of optimal graph (edge) augmentation problem on *key* graph G_K (Figure 1). However, instead of a fixed cost assignment, it defines a nonlinear cost function on the links since the order of augmentation changes the cost assignment. In problem **P2**, new physical links can be created by increasing the power levels to reach a node with a shared secret key. Although this problem can also be formulated as an optimal graph augmentation problem on *physical* graph G_P (Figure 1), it has two main differences. First, increasing power levels induce interference on the nodes and may have an adverse effect on the overall network capacity. Thus, there are interference constraints on the nodes in **P2** to ensure an acceptable signal to interference plus noise ratio (SINR). Second, the cost of each link has two parameters: (i) energy cost for establishing this link, and (ii) amount of interference this links induces on the other nodes. We show that neither **P1** nor **P2** accepts PTAS.

Given the complexity of **P1** and **P2**, we present a distributed heuristic for increasing secure connectivity and study its performance.

Organization of the Paper: Rest of the paper is organized as follows: in Section II, we describe the network model and basic notations. We break problem of optimally increasing secure connectivity into three optimization problems. In Section III, we formulate the first problem **P1** as an instance of edge augmentation problem on the key graph. In Section IV, we formulate the second problem **P2** as a constrained optimization problem with interference constraints on the physical graph. In Section V, we provide a witness based distributed key establishment algorithm for increasing the secure connectivity. Finally, in Section VI we conclude.

II. Notations and Problem Definition

A. Network Model

We model a multi-hop wireless ad hoc network as a set of nodes $WN = \{n_1, n_2, \ldots, n_N\}$ distributed over an Euclidean plane. The Euclidean distance between two nodes n_s and n_r is represented by $d(n_s, n_r)$. In this work, we assume that each node n_s has discrete power levels $(1, 2, 3, \ldots, l^i_{max})$. Each node may have different maximum power level l_{max} due to its battery condition. By changing their power levels (P^l_s: node n_s transmitting at power level l), nodes can control the received signal strength $\frac{P^l_s}{d(n_s,n_r)^\alpha}$ (α is a constant that depends on the medium) on the intended recipient. Successful reception of the message depends on transmission power of the sender, interference and noise on the environments. We use Signal-to-Interference-plus-Noise-Ratio (SINR) model because graph-theoretic modeling of interference ignores the fact that interference coming from different transmitters accumulate and can not be limited to specific border. SINR model considers that a message is successfully received by a receiver if the ratio between received signal strength and noise plus interference from other nodes exceeds a threshold β (Equation 1) which is defined by the hardware.

$$\frac{\frac{P^l_s}{d(n_s,n_r)^\alpha}}{Noise + \sum_{n_k \in WN \setminus n_s} \frac{P^l_k}{d(n_k,n_r)^\alpha}} \geq \beta \tag{1}$$

Wireless networks are generally represented with undirected graphs where uniform transmission range and symmetric links are assumed. **Physical Graph** $G_P = (V, E_P)$ represents network where each node is represented with a vertex, and there is an edge between two vertices if the corresponding nodes are within each others transmission range. For the same vertex set V, **Key Graph** $G_K = (V, E_K)$ represents the key connectivity where there is an edge in between two vertices if the corresponding nodes share or can establish one or more symmetric key to secure their communication. In **Secure Graph** $G_S = (V, E_S)$, there is an edge in between two vertices if they have an edge both in G_P and G_K. In other words, $E_S = E_P \bigcap E_K$ as illustrated in Figure 1.

B. Notations

Nodes which are within each other's radio range are called **neighboring nodes**. A wireless link between two neighboring nodes is called a **physical link**. A physical link between two neighboring nodes that share a key is called a **secure link**. If the nodes don't share a key, then it is an **insecure link**. A **secure (trust) path** is a path on which each physical link is a secure link. A **key path** is a secure path which is used to exchange a shared-key (i.e. with a mechanism similar to Diffie-Hellman [38]). Table I lists the notation used throughout this paper.

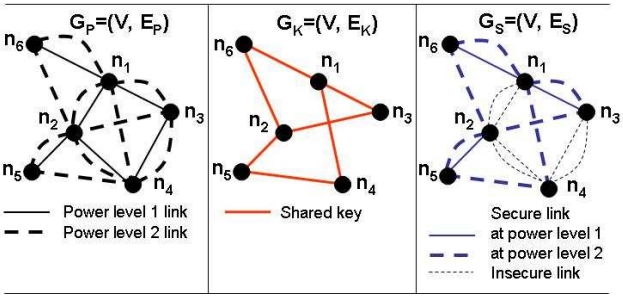

Fig. 1. Physical graph $G_P = (V, E_P)$, Key graph $G_K = (V, E_K)$ and Secure graph $G_S = (V, E_S)$ where $E_S = E_P \bigcap E_K$.

TABLE I
ABBREVIATIONS

WN	Network with nodes $\{n_1, n_2, \ldots, n_N\}$
N	Network size
$\mathcal{F}$	Set of flows (s,t)
T	Set of transmitters $n_{i,l}$
R	Set of receiver n_i
$T(i)$	Transmitters of node n_i
$R(j)$	Receiver of the transmitter $j \in T$
P_i^l	Transmission power of $n_{i,l}$ at power level l
$f_{i,j}^{s,t}$	Flow on edge (i,j) due to flow $f^{s,t}$, $f_{i,j}^{s,t} \in \{0,1\}$
$f^{s,t}$	Flow (s,t), $f^{s,t} \in \{0,1\}$
$n_{i,l}$	i^{th} node transmitting at l^{th} power level
l_{max}^i	Maximum power level for node n_i
$K_{i,j}$	Shared key between nodes n_i and n_j
KC_i	Key-chain of node n_i
E^R	Receive cost of a unit flow
E^T	Transmission cost of a unit flow
$G_P(V, E_P)$	Physical graph
$G_K(V, E_K)$	Key graph
$G_S(V, E_S)$	Secure graph
$G_A(V_A, E_A)$	Auxiliary graph

C. Problem Definition

Upon deployment of a multi-hop wireless ad hoc network, the induced secure graph may be under-utilized (i.e., although $G_S = G_K \cap G_P$ is connected many physical links are not secured by a shared-key, resulting in inefficient routing as shown in Figure 2-C.) or it may be even disconnected as depicted in Figure 2-E.

In this paper we consider the problem of how to optimally increase secure connectivity by either establishing new keys using the secure paths (we rule out executing Diffie-Hellman (DH) [38] or similar techniques over an *insecure* wireless link due to lack of authentication that makes man-in-the-middle attacks possible), or adding new physical links between nodes that share a key by increasing transmission power.

We consider three optimization problems in this paper as summarized in Table II. Problem **P1** is a variant of edge augmentation problem on the keying graph G_K (Figure 2-(A,B)). However, the cost function to be minimized is different from the ones studied in the literature (since the cost assignment of the edges changes as a function of augmentation order).

In problem **P2**, we assume adjustable power levels available to each node and consider the optimal power selection problem to create a physical link between a pair of nodes that share a pre-distributed key. Cost of a physical link depends on the energy consumption which in turn depends on the distance between the corresponding nodes. The optimization problem here is to determine which nodes should increase their power levels to provide secure connectivity at a minimum cost (Figures 2-E,F). Increasing power levels decreases the number of hops in a secure path as illustrated in Figures 2-(C,D). However, increasing transmission power generates more interference on surrounding nodes that also communicate. Enforcing a bound on *instantaneous interference* to ensure acceptable SINR for wireless communications, yields to a mixed integer non-linear optimization problem [39]. Thus **P2** aims to determine the optimal power level assignments so that maximal secure connectivity is obtained under interference constraints.

TABLE II
RESEARCH PROBLEMS CONSIDERED IN THIS PAPER.

Problem	Definition and Approach
P1: $G_K \rightarrow G_S$	Find order of shared key establishment for unsecured physical links. Find optimal secure paths to establish shared keys Approach: Graph augmentation on G_K
P2: $G_P \rightarrow G_S$	Find optimal set of new physical links to be established between the nodes with shared keys Approach: Graph augmentation on G_P constrained with interference
P3: Distributed	Distributed version of P1 Approach: Set coverage problem.

III. PROBLEM P1: $(G_K \rightarrow G_S)$ AUGMENTING THE KEY GRAPH G_K

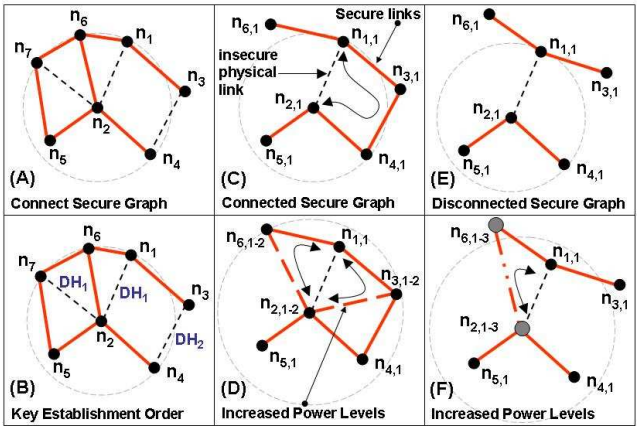

Fig. 2. (A) Under-utilized secure graph $G_S = (V, E_S)$. (B) Order of DH key establishment for minimized cost (e.g., establishing key for (n_1, n_2) first results in shorter secure path for (n_3, n_4). Thus, the order of a key establishment should be $\{(n_1, n_2), (n_2, n_7)\}$ and $\{(n_3, n_4)\}$). (C) Secure graph is connected. Nodes $n_{1,1}$ and $n_{2,1}$ have a physical link but don't share a key. They can communicate through a secure path of 3 hops to establish a key. (D) Nodes n_2, n_3 or n_6 can establish new secure links at power level 2 to provide shorter secure paths for nodes n_1 and n_2. Let $E(l_i)$ be total energy consumed when the node transmits in power level l_i, we simply assume that $E(l_i) + E(l_j) > E(l_{i+j})$. But, increased power level also means more interference is caused on surrounding nodes. (E) Secure graph is disconnected. Nodes n_1 and n_2 have a link but they do not share a key, and they can not find a secure path to establish key. (F) Nodes n_2 and n_6 share a key, and they can establish a new link at power level 3 to provide secure connectivity.

Problem **P1** assumes both key graph G_K and physical graph G_P are connected and it adds edges to G_K to increase key connectivity of *under-utilized multi-hop wireless ad hoc network* to obtain $\kappa - connected$ secure graph where $\kappa \geq 2$

In problem **P1** adding an edge between the nodes n_i and n_j in G_K means establishing keys between node n_i and n_j through a secure path by using Diffie-Hellman (DH) or similar key establishment algorithms. Recall that DH itself does not provide authentication, thus it should be applied through a secure path where each pair of neighboring nodes on the path shares a key.

Consider Figure 2-(A,B) as an example where secure graph is connected. Although each node pairs (n_1, n_2), (n_3, n_4) and (n_2, n_7) has a physical link, they do not share a key to secure their links. These node pairs have to communicate through secure paths yielding under-utilized multi-hop wireless ad hoc network. In this problem, our challenge is three-fold. First, a pair of nodes should be identified to establish a key between them. Second, a minimum cost (e.g., shortest hop count) secure path for each node pair should be found through which DH key establishment can be executed. Third, we should find the order in which DH key establishment should be executed for these pairs. In the secure graph of Figure 2-(A,B), establishing a key for (n_1, n_2) first results in a shorter secure path for the nodes (n_3, n_4).

Problem **P1** is a variant of graph augmentation problem on the keying graph G_K. Given a graph $G = (V, E)$ with n nodes and m edges where each edge (u, v) has an arbitrary non-negative weight $c_{(u,v)}$, let $G' = (V, E')$ be its subgraph where $E' \subseteq E$. The edge augmentation problem is to find minimum-weight set of edges from edge set $E \backslash E'$ whose addition makes G' $\kappa - edge - connected$. The node connectivity augmentation version is slightly different. Given a graph $G = (V, E)$ and a set of vertices $V' \subseteq V$, problem is to find a set of edges with minimum-weight whose addition provides connectivity between every pair of vertices in V'.

The augmentation problem is NP-Hard when $\kappa - edge$ or $\kappa - vertex$ disjoint paths are required between every pair of nodes in V' for $\kappa \geq 2$. However, for fixed cost assignment on the edges it has an approximation (PTAS) which achieves a factor of 2 for $\kappa = 2$ [37]. There is a rich literature of previous work for such tractable variant of **P1** that offers both deterministic [40], [41], [42] and randomized [43] approaches.

However, the cost function to be minimized in **P1** is different from classical graph augmentation since cost of each edge-to-be-inserted (call this a new-edge) to G_S may change as the new edges are added to G_K. For example, suppose the cost or weight of a new-edge (i, j) is the length of the shortest path between i and j in G_S, then this cost will change depending on the order of insertion. This dependency presents a non-linear cost function on the links and makes the order of augmentation important. Thus, optimality depends upon the ordering of the set of node pairs ($E_W \subseteq E_P \backslash E_K$) as illustrated in Figure 2-(A,B). This problem is not only NP-Hard but also it does not admit a PTAS since minimum cutsets to be augmented do not admit constant costs.

IV. Problem P2: ($G_P \rightarrow G_S$) Augmenting the Physical Graph G_P

In this problem, we consider adjusting power levels to create a (new) physical link between a pair of nodes that share a symmetric key. Since increasing transmission power levels generates more interference on surrounding nodes. We enforce a bound on interference to ensure acceptable SINR for wireless communications. As a result, problem **P2** has two parts: (i) identification of optimal number of edges to augment G_P, and (ii) interference constrained power selection for materializing these edges. We use an *auxiliary graph representation* similar to [39] for representing the power levels and formulating the interference constraints.

We note that problem **P2** can be formulated also as an instance of the *edge augmentation* problem. However, there are two complications: (i) interference constraints on the nodes, and (ii) a complex cost function on the edge set that must capture not only the energy cost but also the interference induced on the other nodes. Thus, **P2** is optimal augmentation of G_P subject to interference constraints with a nontrivial cost function.

We formulate *edge augmentation* problem with the interference constraints on nodes and transmission costs on edges as a flow problem using an auxiliary graph $G_A = (V_A, E_A)$ similar to [39].

A. Auxiliary Graph Representation

In this representation, for each node n_i, auxiliary G_A includes a receiver vertex n_i and l^i_{max} transmitter vertices $(n_{i,1}, n_{i,2}, \ldots, n_{i,l^i_{max}})$ corresponding to the each discrete power level. Receivers from all nodes form the receiver set $R = \{n_1, n_2, \ldots, n_i\}$, and transmitters form the transmitter set $T = \{ n_{1,1}, \ldots, n_{1,l^1_{max}}, n_{2,1}, \ldots, n_{2,l^2_{max}}, \ldots, n_{i,1}, \ldots, n_{i,l^i_{max}}\}$ where $V_A = R \bigcup T$. $T(i)$ represents all transmitters $\{n_{i,1}, n_{i,2}, \ldots, n_{i,l^i_{max}}\}$ of the receiver n_i, and $R(j)$ represents receiver n_j of the transmitter $n_{j,l}$. Edge set E_A includes edges (i, j) of types: (1) $i \in R \; and \; j \in T(i)$, and (2) $i \in T \; and \; j \in R$ where there is a shared-key between nodes n_i and n_j (i.e. $(i, j) \in E_K$). First rule states that there are edges from the receiver of each node to all of its transmitters (dashed edges in Figure 3). Second rule states that there is an edge from each transmitter to each receiver located within the transmission range required that both nodes share a key (solid edges in Figure 3). Second types of edges have the costs associated for the energy consumption due to message transmit and receive. All edges have infinite capacities but the network is capacitated due to interference. There is a limit on the amount of interference a receiver can handle meaning that not all transmitters can transmit at the same time. Figure 3 illustrates auxiliary graph corresponding to the example network of Figure 1.

Cost of a physical link (u, v) between nodes u and v is the amount of energy consumed to transfer one unit of flow. Energy consumption depends on the transmit power level. The transmit power level of (u, v) link is the lowest power level which provides transmission radius greater or equal to the Euclidian distance between the nodes.

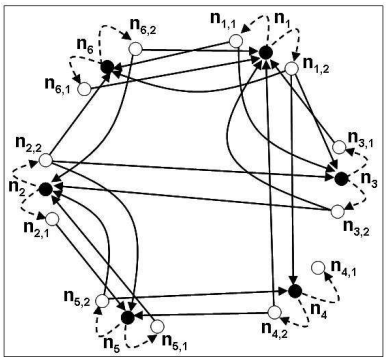

Fig. 3. Auxiliary graph $G_A = (V_A, E_A)$ corresponding to the secure graph $G_S = (V, E_S)$ of Figure 1. Black vertices are receivers $R = \{n_1, n_2, n_3, n_4, n_5, n_6\}$. Each node has two transmit power levels which are the white transmitter vertices $T = \{n_{1,1}, n_{1,2}, n_{2,1}, n_{2,2}, \ldots, n_{6,1}, n_{6,2}\}$. Each solid edge has a cost associated which might be the total energy used by the system to pass one unit of flow and/or the energy consumption due to interference created on the surrounding receivers. Dashed edges have no cost. All edges have infinite capacities but the network is capacitated due to interference because there is a limit on the amount of interference a receiver can handle due to SINR model.

We force a limit on the amount of interference plus noise that a node can tolerate as the *Reception Quality* constraint. This constraint requires that a message is received by a receiver if the ratio between received signal strength and noise plus interference due to surrounding transmitters do not exceed a threshold as specified in Equation 1.

Then, our optimization problem becomes finding minimum cost set of edges on the auxiliary graph subject to the interference constraint where cost of an edge is $E = E^T + E^R$ so that resulting secure graph is $\kappa - connected$.

The optimization problem **P2** has bi-objectives: (1) maximizing the number of concurrent flows -this is the augmentation part, and (2) minimizing the cost which is defined w.r.t. power consumption (since we handle the interference in constraints). Thus, we break the problem into two subproblems and formulate two *integer programs*. In *maximum key establishment flow* problem **P2.1**, we seek for the maximum amount of flow $\mathcal{F}_{Max} \subseteq \mathcal{F}$ that we can grant subject to interference constraints. In *minimum cost key establishment flow* problem **P2.2**, we seek for minimum cost flow assignment on the auxiliary graph edges while keeping $|\mathcal{F}_{Max}|$ and interference as the constraints.

It can be shown that both problems are $NP - Hard$ and $MAX - SNP - Hard$ based on reduction from MAX3SAT (see appendix for formal proofs) which means that they are intractable and it is NP-Hard to approximate them within a factor $1 + \epsilon$ for some fixed $\epsilon > 0$.

We formulate **P2.1** as a constrained optimization problem. The optimization problem aims to maximize the number of source-destination pairs $(s, t) \in \mathcal{F}$ be granted on the auxiliary graph concurrently subject to interference thresholds on each vertex.

B. Problem P2.1: Mathematical Programming Formulation

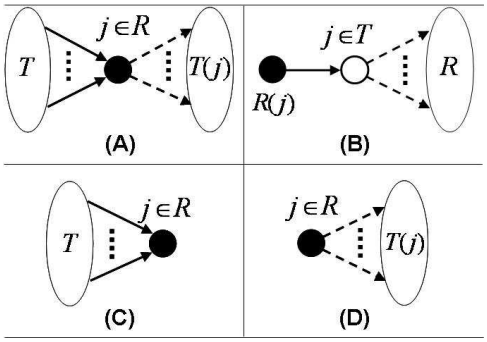

Fig. 4. (A) Receiver flow conservation for Equation 2, (B) Transmitter flow conservation for Equation 3, (C) Receiver utilization for Equation 5, and (D) Transmitter utilization for Equation 6.

*Definition 1 (MaxKeyEstabFlow Problem **P2.1**):* Given the auxiliary graph $G_A = (V_A, E_A)$ representation of a deployment, euclidian distances $d(n_i, n_j)$ between nodes for all node pairs (n_i, n_j), SINR constants β and α, power levels $(1, 2, 3, \ldots, l^i_{max})$ for all nodes n_i and set of flows $\mathcal{F}$ for the key establishment traffic, **P2.1** is the problem of maximizing the number $\mathcal{X}$ ($\mathcal{X} = |\mathcal{F}'|$ where $\mathcal{F}' \subseteq \mathcal{F}$) of source-destination pairs that can exchange key establishment messages concurrently on the auxiliary graph G_A subject to interference constraints. Solution to the problem is the subset $\mathcal{F}'$ of source-destination pairs, and flows of source-destination pairs $(s, t) \in \mathcal{F}'$ assigned to a subset of edges $E'_A \subseteq E_A$.

Problem is similar to *integer multiflow* optimization problem [44] because flows belonging to multiple source-destination pairs $(s, t) \in \mathcal{F}$ is assigned to edges of the auxiliary graph. Vertices of the edges having non-zero flow in the auxiliary graph will correspond to power level of the corresponding pairwise communication.

Let G_A be the auxiliary graph corresponding to a deployment with N nodes. Also, $\mathcal{F}$ is the set of node pairs (s, t) representing neighboring nodes which don't share a key, and which need to exchange key establishment messages. We assume that key establishment is done by exchanging two units of messages between s and t, thus the demand for (s, t) and (t, s) are both one. Then, problem is to find largest routable subset of $\mathcal{F}$ in G_A subject to: (i) flow conservation, (ii) flow symmetry, (iii) utilization, and (iv) reception quality.

Receiver flow conservation constraint requires that the difference between flows coming and leaving a receiver (as in Figure 4-A) due to a flow between (s, t) should be: (i) zero if the node is not the source or the destination, (ii) $f^{s,t} \in \{0, 1\}$ if the node is destination, and (iii) $(-f^{s,t}) \in \{-1, 0\}$ if the node is source. Thus, for each $j \in R$ and $\forall (s, t) \in \mathcal{F}$:

$$\sum_{i \in T} f^{s,t}_{i,j} - \sum_{i \in T(j)} f^{s,t}_{j,i} = x \;\; s.\,t. \;\; \begin{cases} x = f^{s,t}, & \text{j=t;} \\ x = -f^{s,t}, & \text{j=s;} \\ x = 0, & o/w. \end{cases} \quad (2)$$

Transmitter flow conservation constraint requires that all flows coming and leaving a transmitter (as in Figure 4-B) due to a flow between (s, t) should be equivalent. Thus, for each

$j \in T$ and $\forall (s,t) \in \mathcal{F}$:

$$\sum_{i \in R(j)} f_{i,j}^{s,t} - \sum_{i \in R} f_{j,i}^{s,t} = 0. \quad (3)$$

Flow symmetry constraint requires that when there is a flow on link $(n_{i,l}, n_j)$ $(1 \leq l \leq l_{max}^i)$ due to the flow between $(s,t) \in \mathcal{F}$, there should be a flow on link $(n_{j,l'}, n_i)$ $(1 \leq l' \leq l_{max}^j)$ due to the flow between $(t,s) \in \mathcal{F}$. In other words, key exchange request and response messages between two nodes use the same path in secure graph. This assumption helps in that whenever the transmitter $n_{i,l}$ or $n_{j,l'}$ can not be activated due to interference, the other one should not be. Thus, for each node pair n_i and n_j, and $\forall (s,t) \in \mathcal{F}$:

$$\sum_{l=1}^{l_{max}^i} f_{n_{i,l},n_j}^{s,t} - \sum_{l'=1}^{l_{max}^j} f_{n_{j,l'},n_i}^{t,s} = 0. \quad (4)$$

Receiver utilization constraint requires that receiver utilization (as in Figure 4-C) due to a flow should not exceed unity. Thus, for each $j \in R$ and $\forall (s,t) \in \mathcal{F}$:

$$\sum_{i \in T} f_{i,j}^{s,t} \in \{0,1\}. \quad (5)$$

Transmitter utilization constraint requires that transmitter utilization (as in Figure 4-D) due to a flow should not exceed unity. Thus, for each $j \in R$ and $\forall (s,t) \in \mathcal{F}$:

$$\sum_{i \in T(j)} f_{j,i}^{s,t} \in \{0,1\}. \quad (6)$$

Reception Quality constraint states that flow $f_{i,k}^{s,t}$ (flow on edge (i,j) due to flow $f^{s,t}$) exists if the ratio between received signal strength and noise plus interference, due to surrounding transmitters, do not exceed a threshold as specified in Equation 1. This threshold is applicable to a receiver if there exists a flow on this receiver. Thus, given δ^k and $f_{i,j}^{s,t}$ which are the indicator of flow on a transmitter k and on the receiver j respectively:

$$\forall k \in T,\ \delta^k = \begin{cases} 1, & \sum_{(s,t)\in\mathcal{F}} \sum_{m \in R} f_{k,m}^{s,t} > 0; \\ 0, & \text{o/w.} \end{cases}$$

For each $j \in R$:

$$\frac{\frac{P_i^l}{d(n_i,n_j)^\alpha}}{Noise + \sum_{k \in T \setminus \{i\}} \frac{P_k^l \times \delta^k}{d(n_k,n_j)^\alpha}} \geq \beta \times f_{i,j}^{s,t} \quad (7)$$

Our mathematical program becomes:

$$Maximize \quad \mathcal{X} = \sum_{(s,t)\in\mathcal{F}} f^{s,t}$$
$$Subject\ to\ (2), (3), (4), (5), (6), (7).$$

We prove that MaxKeyEstabFlow is NP-hard using a reduction from MAX3SAT problem, which is a truth assignment to the variables, to find maximum number of clauses that can be satisfied in a boolean formula in $3CNF$ form. We define reduction from MAX3SAT to MaxKeyEstabFlow in two steps. First, given a boolean formula in $3CNF$ form with n variables and m clauses, we create a WSN deployment in an Euclidian plane. We create sensor nodes C_i and D_i for i^{th} clause, and sensor nodes x_j and $\overline{x}_j$ for j^{th} variable where only the sensor nodes x_j and $\overline{x}_j$ create interference on each other. We define set of flows $\mathcal{F} = \{(C_i, D_i), (D_i, C_i) | 1 \leq i \leq m\}$. Second, using this WSN deployment we create an auxiliary graph representation as described in Section IV-A. Thus, objective of finding a truth assignment to the variables so that number of clauses that can be satisfied becomes finding maximum number of source-destination pairs in $\mathcal{F}$ which can be granted concurrently both on the WSN and on the auxiliary graph G_A subject to interference constraints.

Inapproximability results for **P2.1** comes from the interference created by the links and the interference threshold constraint. We show that for every $\epsilon > 0$, there is a gap preserving reduction from MAX3SAT to MaxKeyEstabFlow that has parameters (c, 1+ϵ, c$|\mathcal{F}|/2$, 1+ϵ) where $\mathcal{F}$ is the set of flows. We show that MAX3SAT (φ) = c $\Leftrightarrow$ MaxKeyEstabFlow ($\tau(\varphi)$) = c.m. (see appendix for formal proofs).

C. Problem P2.2 Mathematical Programming Formulation

*Definition 2 (MinCostKeyEstabFlow Problem **P2.2**):* Given the auxiliary graph $G_A = (V_A, E_A)$ representation of a deployment, euclidian distances $d(n_i, n_j)$ between nodes for all node pairs (n_i, n_j), SINR constants β and α, power levels $(1, 2, 3, \ldots, l_{max}^i)$ for all nodes n_i, set of flows $\mathcal{F}$ for the key establishment traffic and the maximum number $\mathcal{X}$ of concurrent key establishment flow, it is the problem of finding at least $\mathcal{X}$source-destination pairs which can exchange key establishment messages on the auxiliary graph G_A at a minimum cost subject to interference constraints. Solution to the problem is the subset $\mathcal{F}'$ of source-destination pairs, flows of source-destination pairs $(s,t) \in \mathcal{F}'$ assigned to a subset of edges $E'_A \subseteq E_A$ and overall cost.

Our objective is to grant at least $\mathcal{X}$ flows through the auxiliary graph G_A with a minimum cost. Result of the program is the flow assigned to each link on the auxiliary graph G_A. This result will also imply the power level assignment to each sensor node so to grant at least $\mathcal{X}$ flows between source-destination pairs. Our formulation has the same constraints as the maximization problem: (i) flow conservation, (ii) flow symmetry, (iii) utilization, and (iv) reception quality. In addition to these constraints, we want total flow to be at least $\mathcal{X}$:

Flow bound constraint requires total flow granted by the flow assignment should be at least $\mathcal{X}$. Thus:

$$\sum_{(s,t)\in\mathcal{F}} f^{s,t} \geq \mathcal{X}. \quad (8)$$

Our mathematical program becomes:

$$Minimize \quad \sum_{(s,t)\in\mathcal{F}} \sum_{i \in T,\ j \in R} f_{i,j}^{s,t}\ C_{i,j}$$
$$Subject\ to\ (2), (3), (4), (5), (6), (7), (8).$$

where $C_{i,j} = E^T + E^R$ is the energy cost of a unit flow on the edge (i,j) where $i \in T$ and $j \in R$. All other edges have zero costs.

We prove that MinCostKeyEstabFlow is NP-hard using a reduction from *Weighted MAX3SAT* problem where each clause has a weight, and the problem is to maximize the sum of the weights of satisfied clauses. *Weighted MAX3SAT* is both NP-hard and MAX-SNP [45] problem. We use similar approach as in MaxKeyEstabFlow to show that MinCostKeyEstabFlow problem is both NP-hard and MAX-SNP (see appendix for formal proofs).

V. Problem P3: Witness Based Authenticated Key Establishment

We consider multi-hop wireless ad hoc networks where pairs of nodes do not share keys to secure their links and can not find secure paths for authenticated key establishment. In Problem **P2**, we show that optimally increasing transmission powers of nodes to establish new secure links is both NP-hard and MAX-SNP. Moreover, establishing a key over an insecure wireless link with Diffie-Hellman or similar techniques can not be used due to lack of authentication which also makes the man-in-the-middle kind of attacks possible. Hence, authentication can be achieved if trusted neighbors act as the witness for authentication purpose. We investigate how to integrate witness-based mechanisms in a fully distributed algorithm. Key pre-distribution solutions in general require every pair of neighbors to exchange list of IDs of the keys in their key-chains during shared-key discovery phase. Thus, every node knows key-chains (i.e., only the key IDs) of its one-hop neighbors. Let KC_i^1 be the set of key IDs known to one-hop neighbors of node n_i in the secure graph (e.g., node n_i in Figure 5-A has KC_i^1=$\{(n_i, KC_i = \{key\ IDs\}), (n_j, KC_j = \{key\ IDs\}), (n_k, KC_k = \{key\ IDs\})\}$). Assume that nodes n_i and n_j want to establish a key. During key establishment phase, n_i and n_j can exchange key ID lists KC_i^1 and KC_j^1 (with blinded node IDs) through their insecure link. After this exchange one of the following cases may happen: (i) node n_i shares a key with n_j, (ii) n_i or one of n_i's neighbors shares a key with n_j or one of n_j's neighbors, and (iii) there is no common key so that next-hop neighbors (Figure 5-B) should be involved.

A. Witness Scheme I: One-hop Neighborhood

Figure 5-C and Table III illustrates the message flow for authenticated key establishment between the nodes n_i and n_j with their one-hop trusted neighbors n_k and n_l respectively. Assume that n_k and n_l share a key $K_{k,l}$ (i.e., $KC_i^1 \bigcap KC_j^1 = \{K_{k,l}\}$). With messages (1) and (2), nodes n_i and n_j exchange the list of key IDs of one-hop neighbors. Once a common key $K_{k,l}$ is identified, the owners n_k and n_l (i.e., key $K_{k,l}$ is in both nodes' key-chain) are requested to witness the key establishment between nodes n_i and n_j with messages (3) and (5). Nodes n_k and n_l do not reveal the key $K_{k,l}$, but they independently send its hash value $K'_{i,j}$ to nodes n_i and n_j with encrypted messages (6) and (4) respectively. Nodes n_i and n_j can use the common key $K'_{i,j}$ as the session key or use this key to authenticate a Diffie-Hellman or similar key exchange where a fresh session key $K_{i,j}$ is generated through messages (7) and (8). It is also possible that node n_i shares a key with one of n_j's neighbors (a.k.a., n_j shares a key with one of n_i's neighbors). In this case authenticated key exchange can be completed using messages (1), (2), (3), (4), (7) and (8) (a.k.a., messages (1), (2), (5), (6), (7) and (8)).

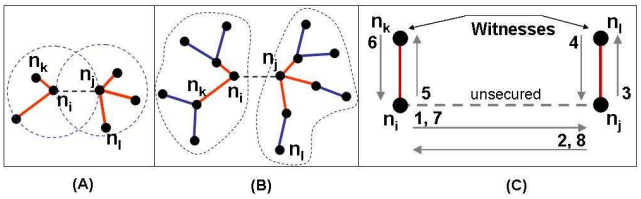

Fig. 5. A distributed approach for authenticated key establishment. Key establishment is authenticated by two witness nodes n_k and n_l located (A) in one-hop, or (B) in two-hop neighborhood of nodes n_i and n_j. (C) Authenticated key establishment between nodes n_i and n_j by using trusted neighbors n_k and n_l as witnesses.

TABLE III
Authenticated key establishment message flow due to the scenario described in Figure 5-C.

No	Source	Destination	Message
(1)	n_i	n_j	KC_i^1
(2)	n_j	n_i	KC_j^1
(3)	n_j	n_l	$ENC_{K_{j,l}}[ID(K_{k,l}), n_i, n_j]$
(4)	n_l	n_j	$ENC_{K_{j,l}}[K'_{i,j} = H(K_{k,l}, n_i, n_j)]$
(5)	n_i	n_k	$ENC_{K_{i,k}}[ID(K_{k,l}), n_i, n_j]$
(6)	n_k	n_i	$ENC_{K_{i,k}}[K'_{i,j} = H(K_{k,l}, n_i, n_j)]$
(7)	n_i	n_j	$ENC_{K'_{i,j}}[g^x\ mod\ p]$
(8)	n_j	n_i	$ENC_{K'_{i,j}}[g^y\ mod\ p]$

Next we briefly analyze probability of finding a witness within one-hop neighborhood using sample probabilistic key management schemes ([3], [4]), combinatorial key management scheme ([13]) and graph theoretic key management scheme ([10]). Table IV lists probability of key share p_S between a pair of nodes for sample key pre-distribution schemes. Parameters are: (k) and (s) size of the key-chain, (KP) key pool, (b) the network size N, (s) and (t) design parameters for Generalized Quadrangle (GQ) and Ramanujan Expander Graph based solutions.

Due to the results of Xue and Kumar in [46], each node should be connected to $\log N$ nearest neighbors for the wireless network to be asymptotically connected with probability one as $N \rightarrow +\infty$. This result is extended to the secure wireless networks in [10]. Thus, in our secure network, each node should be connected to $\log N$ nearest neighbors for the secure multi-hop wireless ad hoc network to be asymptotically connected. This means, nodes n_i and n_j each has average $\log N$ nodes in its one-hop neighborhood. Let p_S be the probability of key share between a pair of nodes n_i and n_j due to the underlying key management technique (sample probabilities p_S are listed in Table IV). Probability that n_i shares a key with one of n_j's neighbors (a.k.a., n_j shares a key with one of n_i's neighbors) is given by the Equation 9. Probability that one of n_i's neighbors shares a key with one

of n_j's neighbors is given by the Equation 10.

$$p(x) = 1 - (1 - p_S)^{(\log N)}. \quad (9)$$

$$p(x) = 1 - (1 - p_S)^{(\log N)^2}. \quad (10)$$

Key Pre-distribution	**Key Sharing Probability** p_S
Random [3], [4]	$P_{RAND} = 1 - \frac{(1-\frac{k}{\lvert KP\rvert})^{2(\lvert KP\rvert-k+1/2)}}{(1-\frac{2k}{\lvert KP\rvert})^{(\lvert KP\rvert-2k+1/2)}}$
Combinatorial [13]	$P_{GQ(s,t)} = \frac{t(s+1)}{b} = \frac{t(s+1)}{(t+1)(st+1)}$
Expander [10]	$P_{X^{s,t}} = \frac{s+1}{t+1}$

TABLE IV
PROBABILITY OF KEY SHARE p_S BETWEEN A PAIR OF NODES FOR SAMPLE KEY PRE-DISTRIBUTION SCHEMES.

B. Witness Scheme II: Next-hop Neighborhood

Figure 5-B and Table V illustrates the message flow for authenticated key establishment between the nodes n_i and n_j when they can not find a witness within their one-hop neighborhood. Assume that n_k and n_l share a key $K_{k,l}$. With messages (1) and (2), nodes n_i and n_j exchange the list of key IDs of their one-hop neighbors and can not find a shared key (i.e., $KC_i^1 \bigcap KC_j^1 = \emptyset$). With messages (3+) and (4+) either n_i or n_j forward their witness request to their neighbors along with the list of key IDs KC_j^1 and KC_i^1 respectively. These messages may be forwarded multiple hops until a witness is found. Assume that n_l is such a witness located in two-hop neighborhood of node n_j. Node n_l shares a key with node n_k which is located in one-hop neighborhood of node n_i. With message (5+), node n_l responds to node n_j through a secure path. Node n_j informs n_i, and n_i informs n_k with messages (6) and (7). Nodes n_k and n_l do not reveal the key $K_{k,l}$, but they independently send its hash value $K'_{i,j}$ to nodes n_i and n_j with encrypted messages (8) and (5) respectively. Nodes n_i and n_j can use the common key $K'_{i,j}$ as the session key or use this key to authenticate a Diffie-Hellman or similar key exchange where a fresh session key $K_{i,j}$ is generated through messages (9) and (10). Flood of messages (3+) and (4+) can be limited by using the knowledge coming from node deployment or key management scheme.

Next, we briefly analyze probability of finding a witness within x-hop neighborhood. In authenticated key establishment scheme, nodes n_i and n_j first check their one-hop neighbors and if there are no nodes which can witness their key establishments, either one of the nodes n_i and n_j checks their two-hop neighborhood. This process continues until a witness is found. We assume that key and physical graphs are connected. Similar to discussions in previous section, in our secure network, each node should be connected to $\log N$ nearest neighbors for the secure network to be asymptotically connected. For simplicity, we assume that N sensor nodes are uniformly deployed in a unit area. That means, each node with transmission radius r, covers an area of πr^2 and has $\log N$ one-hop neighbors. Two-hop neighborhood will cover a circle with radius $2r$ with approximately $4 \log N$ nodes, and x-hop neighborhood will have $x^2 \log N$ nodes. Probability that one of n_i's neighbors shares a key with one of n_j's x-hop neighbors is given by the Equation 11.

TABLE V
AUTHENTICATED KEY ESTABLISHMENT MESSAGE FLOW DUE TO THE SCENARIO DESCRIBED IN FIGURE 5-B.

No	Source	Destination	Message
(1)	n_i	n_j	KC_i^1
(2)	n_j	n_i	KC_j^1
(3+)	n_i	...	KC_j^1, n_i, n_j
(4+)	n_j	...	KC_i^1, n_i, n_j
(5+)	n_l	n_j	$ENC_{...}[ID(K_{k,l}) \mid\mid K'_{i,j} = H(K_{k,l}, n_i, n_j)]$
(6)	n_j	n_i	$ID(K_{k,l}), n_k$
(7)	n_i	n_k	$ENC_{K_{i,k}}[ID(K_{k,l}), n_i, n_j]$
(8)	n_k	n_i	$ENC_{K_{i,k}}[K'_{i,j} = H(K_{k,l}, n_i, n_j)]$
(9)	n_i	n_j	$ENC_{K'_{i,j}}[g^x \ mod \ p]$
(10)	n_j	n_i	$ENC_{K'_{i,j}}[g^y \ mod \ p]$

$$p(x) = 1 - (1 - p_S)^{x^2(\log N)^2}. \quad (11)$$

VI. CONCLUSION AND DISCUSSIONS

We consider the problem of how to maximize the number of secure links (ones that are protected by secret keys) in a multi-hop wireless ad hoc network in order to increase its secure connectivity after deployment.

Key management solutions for in multi-hop wireless ad hoc networks usually assign each node a set of symmetric keys or keying materials (e.g., ID, master keys, hash functions, pseudo random functions, shared polynomials, key matrices and location information), called key-chain, to be shared with *some* of its neighbors after deployment. Although, the unique key-chain assigned to each node before deployment creates a binding to its identity, key pre-distribution schemes are blind to after deployment properties of such ad hoc networks. As a result, many physical links may be left unprotected (i.e., without a shared key on them) which may result in a suboptimal secure routing, or even worse: secured links may not induce a *connected* network. What is needed is to optimally increase the secure connectivity after deployment.

We present several mathematical programming formulations, namely *maximum key establishment flow* and *minimum cost key establishment flow*, as variants of graph augmentation problems. We prove that finding optimum solutions and finding polynomial time approximations are both NP-hard. We place these problems in inapproximability Class I [37] which is the richest class of all. Thus, we present a distributed heuristic algorithm for increasing the secure connectivity of multi-hop wireless ad hoc networks after deployment and analyze its performance.

REFERENCES

[1] J. Zhang and V. Varadharajan, "Wireles ssensor network key management survey and taxonomy," *J. Network and Computer Applications*, vol. 33, 2010.

[2] S. A. Camtepe and B. Yener, *Wireless Sensor Network Security*. IOS Press, Cryptology and Information Security Series, Javier Lopez and Jianying Zhou (editors), ISBN 978-1-58603-813-7, 2008, ch. Key Management.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *ACM CCS*, 2002.

[4] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symp. Security and Privacy*, 2003.

[5] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," in *IEEE ICNP*, 2003.

[6] J. Hwang and Y. Kim, "Revisiting random key pre-distribution for sensor networks," in *ACM SASN*, 2004.

[7] T. Ito, H. Ohta, N. Matsuda, and T. Yoneda, "A key pre-distribution scheme for secure sensor networks using probability density function of node deployment," in *ACM SASN*, 2005.

[8] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *ACM Conf. Computer and Commun. Security*, 2003, pp. 231–240.

[9] H. Shafiei, A. Mehdizadeh, A. Khonsari, and M. Ould-Khaoua, "A combinatorial approach for key-distribution in wireless sensor networks," in *IEEE GLOBECOM*, 2008.

[10] S. A. Camtepe, B. Yener, and M. Yung, "Expander graph based key distribution mechanisms in wireless sensor networks," in *IEEE ICC*, 2006.

[11] E. S. Elmallah, M. G. Gouda, and S. S. Kulkarni, "Logarithmic keying," *ACM Trans. Auton. Adapt. Syst.*, vol. 3, no. 4, 2008.

[12] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, pp. 122–152, 2006.

[13] S. A. Camtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," *IEEE/ACM TON*, vol. 15, no. 2, 2007.

[14] J. Lee and D. R. Stinson, "Deterministic key pre-distribution schemes for distributed sensor networks," in *ACM SAC*, 2005.

[15] D. S. Sanchez and H. Baldus, "A deterministic pairwise key predistribution scheme for mobile sensor networks," in *IEEE Securecomm*, 2005.

[16] F. Delgosha and F. Fekri, "Key pre-distribution in wireless sensor networks using multivariate polynomials," in *IEEE SECOM*, 2005.

[17] Z. Yu and Y. Guan, "A robust group-based key management scheme for wireless sensor networks." in *IEEE WCNC*, 2005.

[18] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *ACM CCS*, 2003.

[19] A. K. Das, "Ecpks: An improved location-aware key management scheme in static sensor networks," *Int. J. of Network Security*, vol. 7, no. 3, 2008.

[20] C. Huang and D. Du, "New constructions on broadcast encryption and key pre-distribution schemes." in *IEEE INFOCOM*, 2005.

[21] D. Liu and P. Ning, "Location-based pairwise key establishment for static sensor networks," in *ACM SASN*, 2003.

[22] H. Chan and A. Perrig, "Pike: Peer intermediaries for key establishment," in *IEEE INFOCOM*, 2005.

[23] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *IEEE INFOCOM*, 2004, p. 597.

[24] B. Lai, S. Kim, and I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks," in *IEEE Workshop on Large Scale Real-Time and Embedded Systems*, 2002.

[25] B. Dutertre, S. Cheung, and J. Levy, "Lightweight key management in wireless sensor networks by leveraging initial trust," System Design Laboratory, Tech. Rep. SRI-SDL-04-02, 2004.

[26] C.-K. Chu, J. K. Liu, J. Zhou, F. Bao, and R. H. Deng, "Practical id-based encryption for wireless sensor network," in *ACM ASIACCS*, 2010.

[27] W. Hu, P. Corke, W. C. Shih, and L. Overs, "secfleck: A public key technology platform for wireless sensor networks," in *EWSN*, 2009.

[28] J. Zhang and V. Varadharajan, "Group-based wireless sensor network security scheme," in *ICWMC*, 2008.

[29] G. Yang, C. Rong, C. Veigner, J. Wang, and H. Cheng, "Identity-based key agreement and encryption for wireless sensor networks," in *IJCSNS Int. J. Computer Science and Network Security*, 2006.

[30] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *IEEE SECON*, 2004.

[31] P. Santi, R. Maheshwari, G. Resta, S. Das, and D. M. Blough, "Wireless link scheduling under a graded sinr interference model," in *ACM FOWANC*, 2009.

[32] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric sinr," in *ACM MobiHoc*, 2007.

[33] T. Moscibroda, R. Wattenhofer, and A. Zollinger, "Topology control meets sinr: the scheduling complexity of arbitrary topologies," in *ACM MobiHoc*, 2006.

[34] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *MobiCom*, 2006.

[35] V. Bhandari and N. H. Vaidya, "Secure capacity of multi-hop wireless networks with random key pre-distribution," in *IEEE MCN*, 2008.

[36] D. Liu, P. Ning, and W. Du, "Group-based key predistribution for wireless sensor networks," *ACM TOSN*, vol. 4, no. 2, 2008.

[37] D. S. Hochbaum, *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

[38] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, 1976.

[39] O. Savas, M. Alanyali, and B. Yener, "Joint route and power assignment in asynchronous multi-hop wireless networks," in *MedHocNet*, 2004.

[40] T. Watanabe, Y. Higashi, and A. Nakamura, "Graph augmentation problems for a specific set of vertices," in *Int. Symp. on Algorithms*, 1990.

[41] D. Naor, D. Gusfield, and C. Martel, "A fast algorithm for optimally increasing the edge connectivity," *SIAM Journal of Computing*, vol. 26, no. 4, 1997.

[42] H. Nagamochi and T. Ibaraki, "Augmenting edge-connectivity over the entire range in o(nm) time," *J. Algorithms*, vol. 30, no. 2, 1999.

[43] A. A. Benczúr and D. R. Karger, "Augmenting undirected edge connectivity in (n^2) time," in *ACM-SIAM SODA*, 1998.

[44] M.-C. Costa, L. Létocart, and F.Roupin, "Minimal multicut and maximal integer multiflow: a survey," *Elsevier J. of Operational Research*, vol. 162, 2005.

[45] C. H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," *J. Computer and System Sciences*, vol. 43, no. 3, 1991.

[46] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, 2004.

APPENDIX

Proof: (MaxKeyEstabFlow in NP-hard) We prove that MaxKeyEstabFlow is NP-Hard using a reduction from MAX3SAT problem which is a truth assignment to the variables $\{x_1, x_2, \ldots, x_n\}$ to find maximum number of clauses that can be satisfied in a boolean formula φ in $3CNF$ form with clauses $\{C_1, C_2, \ldots, C_m\}$. We define reduction τ from MAX3SAT to MaxKeyEstabFlow in two steps. In the first step we show how to reduce a MAX3SAT problem instance into WSN problem instance, and in the second step reduce it into an auxiliary graph representation.

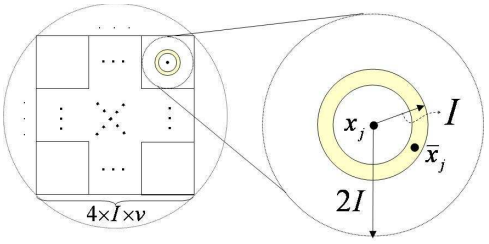

Fig. 6. Placement of sensor nodes on a unit disk area as the part of the reduction from MAX3SAT problem to MaxKeyEstabFlow in WSN (NP-hard proof of MaxKeyEstabFlow). Step-1 item-7 of the proof describes the replacement algorithm. Each sensor node x_j is placed at the center of a random grid location. Node $\overline{x}_j$ is placed at a random location in the grid at a distance which ensures interference with x_j. Sensor nodes C_i and D_i are placed at any random locations. No two sensor nodes (corresponding to boolean variables) except pairs x_j and $\overline{x}_j$ can create interference on each other.

Step 1: Given a boolean formula φ in $3CNF$ form with n variables and m clauses, create a WSN deployment in an Euclidian plane (for $1 \leq i \leq m$ and $1 \leq j \leq n$):

1) Create sets of sensor nodes: $C = \{C_i | 1 \leq i \leq m\}$, $D = \{D_i | 1 \leq i \leq m\}$, $X = \{x_j | 1 \leq j \leq n\}$ and $\overline{X} = \{\overline{x_j} | 1 \leq j \leq n\}$. Namely, create sensor nodes C_i and D_i for i^{th} clause, and sensor nodes x_j and $\overline{x}_j$ for j^{th} variable.
2) Sensor nodes x_j and $\overline{x}_j$ have a maximum power level of $l^j_{max} = 1$. Sensor nodes C_i and D_i have a maximum power level of $l^i_{max} = L_{max}$ which covers whole WSN and can use RTS/CTS signalling to check availability of channel at receivers.
3) Only the sensor nodes x_j and $\overline{x}_j$ create interference on each other. This means boolean variables x_j and $\overline{x}_j$ can not be $true$ at the same time; similarly, sensor nodes x_j and $\overline{x}_j$ can not transmit at the same time.
4) Distribute a key-chain KC to each sensor node. KC_{C_i} and KC_{x_j} (a.k.a., $KC_{\overline{x}_j}$) should share a distinct key if variable x_j (a.k.a., $\overline{x}_j$) appears in i^{th} clause. Similarly, KC_{D_i} and KC_{x_j} (a.k.a., $KC_{\overline{x}_j}$) should share a distinct key if variable x_j (a.k.a., $\overline{x}_j$) appears in i^{th} clause. All other pairs of key-chains should be distinct.
5) Create set of flows $\mathcal{F} = \{(C_i, D_i), (D_i, C_i) | 1 \leq i \leq m\}$. These are the pairs of nodes which have physical links but do not share keys to secure their communications.
6) Place the sensor nodes on a unit disk area as illustrated in Figure 6:
   a) Draw $v \times v$ ($v = \lceil \sqrt{n} \rceil$) grid for n variables.
   b) Each grid location should be a square of size $4I \times 4I$ where I is the minimum distance that two nodes can communicate without creating interference on each other.
   c) For each sensor node x_j, select a random empty grid coordinate and locate the node at the center of the grid location.
   d) Place each sensor node $\overline{x}_j$ at a random location where Euclidian distance between $d(x_j, \overline{x}_j) < I$. This and grid location sizes guarantee that no two sensor nodes (corresponding to boolean variables) except pairs x_j and $\overline{x}_j$ fall into each others radio or interference range.
   e) Place sensor nodes C_i and D_i at any random location.

Step 2: Given a WSN deployment which is reduced from a boolean formula φ in $3CNF$ form with n variables and m clauses, create an auxiliary graph representation as described in Section IV-A:

1) Create auxiliary graph $G_A = (V_A, E_A)$ (for $1 \leq i \leq m$, $1 \leq j \leq n$ and $1 \leq g \leq L_{max}$):
   a) Receiver nodes are $R = C \bigcup D \bigcup X \bigcup \overline{X}$.
   b) Add transmitter nodes $C_i^{T_g}$, $D_i^{T_g}$, x_j^T and $\overline{x}_j^T$.
   c) Add directed edges (C_i^R, $C_i^{T_g}$) and (D_i^R, $D_i^{T_g}$), (x_j^R, x_j^T) and ($\overline{x}_j^R$, $\overline{x}_j^T$).
   d) Add directed edges ($C_i^{T_g}$, x_j^R) (a.k.a., $\overline{x}_j^R$) and (x_j^T, C_i^R) (a.k.a, $\overline{x}_j^T$) if x_j (a.k.a, $\overline{x}_j$) shares a key with C_i.
   e) Add directed edges ($D_i^{T_g}$, x_j^R) (a.k.a., $\overline{x}_j^R$) and (x_j^T, D_i^R) (a.k.a, $\overline{x}_j^T$) if x_j (a.k.a, $\overline{x}_j$) shares a key with D_i.
2) Set edge capacities as infinite.
3) Create set of flows $\mathcal{F} = \{(C_i, D_i), (D_i, C_i) | 1 \leq i \leq m\}$. Figure 7 provides an auxiliary graph reduced from a sample boolean formula.

This algorithm transforms a boolean formula φ in $3CNF$ form with n variables and m clauses first into a WSN deployment with $2(m+n)$ nodes, then into an auxiliary graph G_A with $(2m(L_{max}+1)+4n)$ nodes and $O(m+2n)$ edges where $|\mathcal{F}| = 2m$. Objective of finding a truth assignment to the variables so that number of clauses that can be satisfied is maximized becomes finding maximum number of source-destination pairs in $\mathcal{F}$ which can be granted concurrently both on the WSN and on the auxiliary graph G_A subject to interference constraints. Thus, the transformation from MAX3SAT to MaxKeyEstabFlow can be carried out in polynomial time.

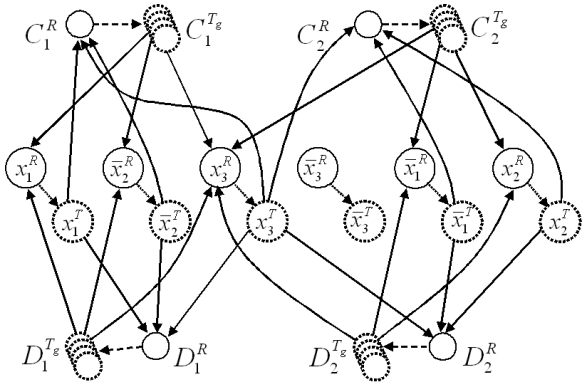

Fig. 7. Auxiliary graph $G_A = (V_A, E_A)$ reduced from sample boolean formula $\varphi = ((x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3))$ as an illustration of the reduction (τ: MAX3SAT $\rightarrow$ MaxKeyEstabFlow) described in the NP-hard proof of MaxKeyEstabFlow. There is only one transmit power level for the nodes corresponding to the boolean variables. Nodes C_1, C_2, D_1, D_2 have L_{max} transmit power levels. Set of receivers are $R = \{x_1^R, x_2^R, x_3^R, \overline{x}_1^R, \overline{x}_2^R, \overline{x}_3^R, C_1^R, C_2^R, D_1^R, D_2^R\}$, and set of transmitters are $T = \{x_1^T, x_2^T, x_3^T, \overline{x}_1^T, \overline{x}_2^T, \overline{x}_3^T, C_1^{T_g}, C_2^{T_g}, D_1^{T_g}, D_2^{T_g}\}$ for $1 \leq g \leq L_{max}$ where $V_A = R \bigcup T$. All edges have infinite capacities. Finally set of flow is $\mathcal{F} = \{(C_1, D_1), (C_2, D_2), (D_1, C_1), (D_2, C_2)\}$.

Solution to the problem instance $\tau(\xi)$ of MaxKeyEstabFlow in auxiliary graph representation can be converted to the solution of problem instance ξ of MAX3SAT in two easy steps in linear time. First, if total flow on the transmitter $x_j^T \geq 1$ (a.k.a. $\overline{x}_j^T \geq 1$) then set boolean variables $x_j = True$ (a.k.a. $\overline{x}_j = True$) and $\overline{x}_j = False$ (a.k.a. $x_j = False$) for $1 \leq j \leq n$. Note that interference constraint does not permit both flows $x_j^T \geq 1$ and $\overline{x}_j^T \geq 1$. Second, if total flow on both transmitters are $x_j^T = 0$ and $\overline{x}_j^T = 0$, then set either ($x_j = True$ and $\overline{x}_j = False$) or ($x_j = False$ and $\overline{x}_j = True$) for $1 \leq j \leq n$. This assignment does not change the number of the satisfied clauses in ξ, but some satisfied clauses have more than one variable set to $True$. Very similar steps apply for converting the solution to the problem instance $\tau(\xi)$ of MaxKeyEstabFlow in WSN to solution to the problem instance ξ of MAX3SAT in linear time. The flows on sensor

nodes x_j and $\overline{x}_j$ should be considered instead of the flows on transmitters x_j^T and $\overline{x}_j^T$.

Optimal solution to the instance ξ of MAX3SAT has c satisfied clauses if and only if the optimal solution to the instance $\tau(\xi)$ of MaxKeyEstabFlow on WSN and auxiliary graph representations has c source-destination pairs (C, D) (i.e. flows $(C, D), (D, C) \in \mathcal{F}$) which are granted. For $MAX3SAT \rightarrow MaxKeyEstabFlow$, assume that $\tau(\xi)$ has optimal solution $d > c$. Then it would be possible to satisfy more than c clauses by simply setting $True$ value for the variables which have more than one unit of flow on the corresponding transmitter node. This contradicts the fact that ξ has optimal solution c. Similarly for $MaxKeyEstabFlow \rightarrow MAX3SAT$, assume that ξ has optimal solution $d > c$. Then it would be possible to grant flow for d source-destination pairs by putting more than one unit of flow on transmitters (sensor nodes in the WSN deployment) corresponding to d variables, one $True$ variable from each satisfied clauses without contradicting interference constraints. This contradicts the fact that $\tau(\xi)$ has optimal solution c. ■

Definition 3: [37, Definition 10.4] A maximization problem Π is MAX-SNP-Hard if for every MAX-SNP problem Γ and every two constants $c \leq 1$, $\rho > 1$, there are two constants $c' \leq 1$, $\rho' > 1$ such that there is a gap preserving reduction from Γ to Π with parameters (c, ρ, c', ρ').

MAX3SAT is a MAX-SNP problem [45] where its optimum c is a fraction equivalent to the maximum number of satisfiable clauses divided by the total number of clauses. It is NP-Hard to approximate MAX3SAT within a fixed ratio $\rho = 1 + \epsilon$ for $\epsilon > 0$. For proving inapproximability results, we use *gap preserving reduction* as described in Definition 3.

Proof: (MaxKeyEstabFlow in MAX-SNP) For every $\epsilon > 0$, there is a gap preserving reduction from MAX3SAT to MaxKeyEstabFlow that has parameters (c, 1+ϵ, c$|\mathcal{F}|/2$, 1+ϵ) where $\mathcal{F}$ is the set of flows. We use the polynomial time reduction τ from MAX3SAT to MaxKeyEstabFlow described in the NP-hard proof of MaxKeyEstabFlow. Let φ be a boolean formula in $3CNF$ form with n variables and m clauses. MAX3SAT(φ) represents the maximum number of satisfiable clauses divided by the total number of clauses, and MaxKeyEstabFlow ($\tau(\varphi)$) represents the maximum number of source-destination pairs that can exchange messages. We will show that MAX3SAT (φ) = c $\Leftrightarrow$ MaxKeyEstabFlow ($\tau(\varphi)$) = c.m. First, assume that MAX3SAT(φ)=c. There must be $c.m$ satisfied clauses. Each satisfied clause C_i must have at least one satisfied variable where the corresponding transmitter node (sensor node in WSN deployment) may have one or more unit of flow, meaning that corresponding source-destination pair (C_i, D_i) can be granted. Thus, $MaxKeyEstabFlow\ (\tau(\varphi)) \geq c.m$. Second, assume that MaxKeyEstabFlow ($\tau(\varphi)$) = c.m. There must be $c.m$ source-destination pairs granted. Each granted source-destination pair (C_i, D_i) means one satisfied clause C_i so that MAX3SAT(φ) $\geq c$. Thus:

$$MAX3SAT(\varphi) = c \ \Rightarrow \ MaxKeyEstabFlow(\tau(\varphi)) = c.m$$
$$MAX3SAT(\varphi) < \tfrac{c}{1+\epsilon} \Rightarrow MaxKeyEstabFlow(\tau(\varphi)) < \tfrac{c.m}{1+\epsilon}.$$

This gap-preserving reduction from MAX3SAT shows that it is NP-Hard to approximate MaxKeyEstabFlow within factor $1 + \epsilon$. Thus, MaxKeyEstabFlow is MAX-SNP-Hard, meaning also that MaxKeyEstabFlow doesn't have a polynomial time approximation scheme (PTAS) unless $P = NP$. ■

Proof: (MinCostKeyEstabFlow in both NP-hard and MAX-SNP-hard) We use *Weighted MAX3SAT* problem where each clause has a weight, and the problem is to maximize the sum of the weights of satisfied clauses. *Weighted MAX3SAT* is both NP-Hard and MAX-SNP-Hard [45] problem. We can show that MinCostKeyEstabFlow problem is both NP-Hard and MAX-SNP-Hard by using a polynomial time reduction from *Weighted MAX3SAT* to MinCostKeyEstabFlow which is obtained by adding two simple steps to reduction algorithm τ of NP-hard proof of MaxKeyEstabFlow. Consider a boolean formula φ in $3CNF$ form with n variables and m clauses with weights (i.e. weight w_i for the clause C_i). First, for $1 \leq i \leq m$ and $1 \leq j \leq n$, set cost $(-w_i/2)$ for the edge (C_i, x_j) (a.k.a. $(C_i, \overline{x}_j)$) of WSN deployment where x_j (a.k.a $\overline{x}_j$) appears in clause C_i (set cost $(-w_i/2)$ for the edge $(C_i^{T_g}, x_j^R)$ (a.k.a. $(C_i^{T_g}, \overline{x}_j^R)$) of auxiliary graph representation where $1 \leq g \leq L_{max}$. All other edges have zero costs. Second, set $\mathcal{X} = 1$. Problem of maximizing the sum of the weights of the satisfied clauses becomes problem of minimizing the cost of granting one or more source-destination pairs subject to interference constraint. The rest of the proof follows the discussions in NP-hard and MAX-SNP proofs of MaxKeyEstabFlow. We conclude that MinCostKeyEstabFlow problem is both NP-Hard and MAX-SNP-Hard, meaning also that MinCostKeyEstabFlow doesn't have a polynomial time approximation scheme (PTAS) unless $P = NP$. ■