Complexity of Increasing the Secure Connectivity in Wireless Ad Hoc Networks

Seyit A. Camtepe

Queensland University of Technology

Abstract. We consider the problem of maximizing the secure connectivity in wireless ad hoc networks, and analyze complexity of the post-deployment key establishment process constrained by physical layer properties such as connectivity, energy consumption and interference. 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 shared 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 extends the first problem by 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 prove 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.

1 Introduction

Efficient key management schemes are essential to ensure the integrity and confidentiality in wireless ad hoc networks. An example of such networks are wireless sensor networks operating in adversarial conditions. Many different key management schemes are proposed for the wireless sensor networks. Some solutions assign (a.k.a. pre-distribute) 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 can be found in [1, 2].

We consider the problem of how to maximize the number of secure links in a wireless sensor network in order to increase its secure connectivity after deployment. In most deployment schemes, sensor nodes are randomly scattered

over a large application area which might be inaccessible or infeasible to access after the deployment. Even with the controlled placement of sensor nodes, due to environmental challenges and deployment errors, the post-deployment network configuration might be unknown a priori. After the deployment, each node discovers its neighbors and tries to find a key to secure its wireless links in key discovery phase. Key management schemes are mostly blind to after deployment properties [1, 2]; therefore, many physical links may be left unprotected (i.e., without a 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 (Figure 1). 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 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 as illustrated in Figure 1. Focus of this work is to understand complexity of the key establishment process in distributed wireless sensor networks subject to the physical layer properties such as connectivity, energy consumption and interference. In the broader sense, we would like to understand feasibility of existing key management schemes which trust on post deployment key establishment processes for secure connectivity.

Utilization of multi-hop wireless networks is investigated as the 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 [3–5] and shown to be NP-complete in various formulations. Secure capacity of a randomly deployed network is analyzed in [6] where each node receives a key-chain due to the *random key pre-distribution scheme* [7]. In [8], 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 is required on analyzing the complexity of increasing the secure connectivity and secure capacity in wireless ad hoc networks.

Contribution: Our contribution is theoretical as we formulate the different variants of the problem and analyze their complexity. In particular, we present two approaches: (i) establish new symmetric keys for 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 key (problem **P2**). Both of the problems are variants of the graph augmentation problem which are in general NP-hard for fixed cost functions and accept polynomial time constant approximation schemes (PTAS) [9].

Problem **P1** is a variant of the 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. Al-

though this problem can also be formulated as an optimal graph augmentation problem on the *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 link induces on the other nodes. We prove that neither **P1** nor **P2** accepts PTAS.

Organization: Rest of the paper is organized as follows: in Section 2, we describe the network model and basic notations. We break the problem of optimally increasing secure connectivity into three optimization problems. In Section 3, we formulate the first problem **P1** as an instance of edge augmentation problem on the key graph. In Section 4, we formulate the second problem **P2** as a constrained optimization problem with interference constraints on the physical graph. Finally, in Section 5, we conclude.

2 Network Model and Problem Definition

2.1 Network Model and Notations

We model a wireless sensor 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 (sender) and n_r (receiver, $1 \leq s, r \leq N$) is represented by $d(n_s, n_r)$. In this work, we assume that each node n_s has discrete power levels $(1, 2, 3, \ldots, l^i_{max}$ where $1 \leq i \leq N)$. 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 r. We use the Signal-to-Interference-plus-Noise-Ratio (SINR) model because the graph-theoretic modeling of interference ignores the fact that interference coming from different transmitters accumulate and can not be limited to a 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 the uniform transmission range and symmetric links are assumed. **Physical Graph** $G_P = (V, E_P)$ represents a 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 keys 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.

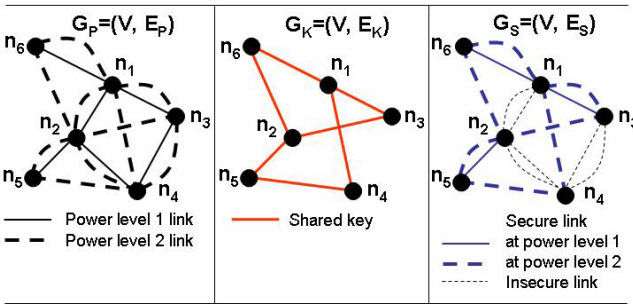

Fig. 1. Physical graph $G_P = (V, E_P)$, Key graph $G_K = (V, E_K)$ and Secure graph $G_S = (V, E_S)$ where $E_S = E_P \bigcap E_K$.

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 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). Table 1 lists notations used throughout this paper.

Table 1. Abbreviations

WN	Network with N nodes $\{n_1, \ldots, n_N\}$	$\mathcal{F}$	Set of flows (s, t)
T,	Set of transmitters $(n_{i,l})$	R	Set of receivers (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 level l	$f^{s,t}$	Flow (s, t), $f^{s,t} \in \{0, 1\}$
$f_{i,j}^{s,t}$	Flow on edge (i, j) due to flow $f^{s,t}$	$n_{i,l}$	i^{th} node transmitting at level l
l_{max}^i	Maximum power level of 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	Physical graph $G_P(V, E_P)$
G_K	Key graph $G_K(V, E_K)$	G_S	Secure graph $G_S(V, E_S)$
G_A	Auxiliary graph $G_A(V_A, E_A)$		

2.2 Problem Definition

Upon deployment of a wireless sensor network, the induced secure graph may be under-utilized because, although $G_S = G_K \cap G_P$ is connected many physical links may not be secured by a key resulting in inefficient routing as shown in Figure 2-C. It may be even disconnected as depicted in Figure 2-E.

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

- **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).

3 Problem P1: Augmenting the Key Graph G_K

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 an *under-utilized wireless sensor 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 nodes 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, rather than using their direct link, yielding an under-utilized 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, the order in which DH key establishment is executed should be identified. In the secure graph of Figure 2-(A,B), establishing a key first for (n_1, n_2) 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 the edge set $E \setminus E'$ 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

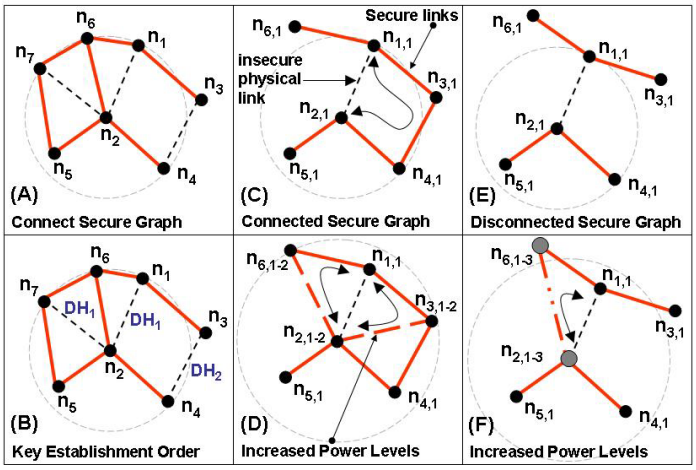

Fig. 2. (A) Under-utilized secure graph $G_S = (V, E_S)$. (B) Order of Diffie-Hellman key establishment for the minimized cost (e.g., establishing key for (n_1, n_2) first results in shorter secure path for (n_3, n_4)). (C) Secure graph is connected. Nodes $n_{1,1}$ and $n_{2,1}$ have a physical link 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 the power level 2 to provide shorter secure paths for nodes n_1 and n_2. (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 the power level 3 to provide the secure connectivity.

cost assignment on the edges it has an approximation (PTAS) which achieves a factor of 2 for $\kappa = 2$ [9]. There is a rich literature of previous work for such tractable variants of **P1** that offer both deterministic [10, 11] and randomized [12] approaches.

However, the cost function to be minimized in **P1** is different from classical graph augmentation since the 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.

4 Problem P2: 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. The optimization problem

here is to determine which nodes should increase their power levels to provide the 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. Enforcing a bound on *instantaneous interference* to ensure acceptable SINR for wireless communications yields to a mixed integer non-linear optimization problem [13]. Thus, 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 [13] 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 [13].

4.1 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-A). 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-A). These edges have cost associated with them as the amount of energy consumed to transfer one unit of flow. For simplicity, all edges considered to have unlimited 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.

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 the received signal strength and the interference-plus-noise due to surrounding transmitters do not exceed a threshold as specified in Equation 1. Then, our optimization problem becomes finding a minimum cost set of edges on the auxiliary graph subject to the interference

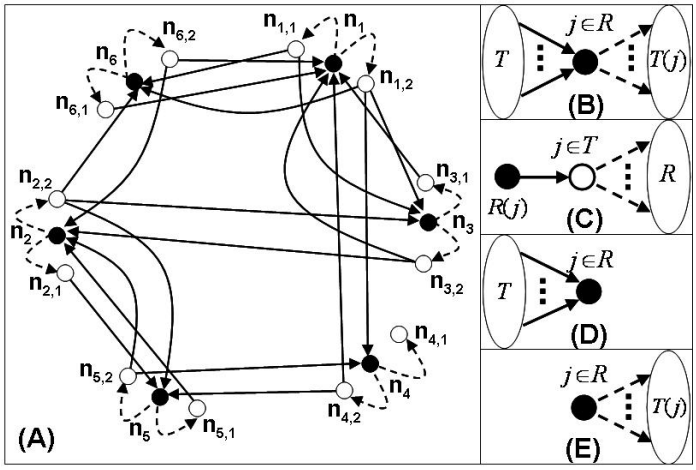

Fig. 3. (A) Auxiliary graph $G_A = (V_A, E_A)$ corresponding to the secure graph $G_S = (V, E_S)$ of Figure 1. Black vertices are receivers $R = \{n_1, n_2, n_3, n_4, n_5, n_6\}$. Each node has two transmit power levels which are the white transmitter vertices $T = \{n_{1,1}, n_{1,2}, n_{2,1}, n_{2,2}, \ldots, n_{6,1}, n_{6,2}\}$. Each solid edge has a cost associated which is 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 unlimited capacities but the network is capacitated due to the interference because there is a limit on the amount of interference a receiver can handle due to SINR model. (B) Receiver flow conservation for Equation 2, (C) Transmitter flow conservation for Equation 3, (D) Receiver utilization for Equation 5, and (E) Transmitter utilization for Equation 6.

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 the interference as constraints.

Having formulated the problem as an auxiliary graph, it can be shown that both problems **P2.1** and **P2.2** are $NP-Hard$ and $MAX-SNP-Hard$ based on a reduction from MAX3SAT (see the appendix for formal proofs). Thus, they are intractable and it is NP-Hard to approximate them within a factor $1+\epsilon$ for some fixed $\epsilon > 0$.

4.2 Problem P2.1: Mathematical Programming Formulation

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.

Definition 1 (MaxKeyEstabFlow Problem P2.1). *Given $G_A = (V_A, E_A)$ the auxiliary graph 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 [14] 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 the 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, the problem is to find largest routable subset of $\mathcal{F}$ in G_A subject to: (i) flow conservation (receiver and transmitter), (ii) flow symmetry, (iii) utilization (receiver and transmitter), and (iv) reception quality:

(i.a) Receiver flow conservation constraint requires that the difference between flows coming and leaving a receiver (as in Figure 3-B) 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 $\forall (s,t) \in \mathcal{F}$:

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

(i.b) Transmitter flow conservation constraint requires that all flows coming and leaving a transmitter (as in Figure 3-C) due to a flow between (s,t) should be equivalent. Thus, for each $j \in T$ and $\forall (s,t) \in \mathcal{F}$:

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

(ii) Flow symmetry constraint requires that when there is a flow on link $(n_{i,l}, n_j)$ $(1 \leq l \leq l^i_{max})$ due to the flow between $(s,t) \in \mathcal{F}$, there should be a

flow on link $(n_{j,l'}, n_i)$ $(1 \leq l' \leq l^j_{max})$ 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 the secure graph. This assumption helps in that whenever one of the transmitters $n_{i,l}$ or $n_{j,l'}$ can not be activated due to the interference, the other one should not be. Thus, for each node pair n_i and n_j, and $\forall(s,t) \in \mathcal{F}$:

$$\sum_{l=1}^{l^i_{max}} f^{s,t}_{n_{i,l},n_j} - \sum_{l'=1}^{l^j_{max}} f^{t,s}_{n_{j,l'},n_i} = 0. \tag{4}$$

(iii.a) Receiver utilization constraint requires that the receiver utilization (as in Figure 3-D) due to a flow should not exceed the unity. Thus, for each $j \in R$ and $\forall(s,t) \in \mathcal{F}$:

$$\sum_{i \in T} f^{s,t}_{i,j} \in \{0, 1\}. \tag{5}$$

(iii.b) Transmitter utilization constraint requires that the transmitter utilization (as in Figure 3-E) due to a flow should not exceed unity. Thus, for each $j \in R$ and $\forall(s,t) \in \mathcal{F}$:

$$\sum_{i \in T(j)} f^{s,t}_{j,i} \in \{0, 1\}. \tag{6}$$

(iv) Reception Quality constraint states that flow $f^{s,t}_{i,j}$ (flow on edge (i,j) due to flow $f^{s,t}$) exists (non-zero) if the ratio between the received signal strength and the interference-plus-noise, 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 $f^{s,t}_{i,j}$ which are the indicator of a flow on each transmitter k and on the receiver j respectively:

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

For each $j \in R$:

$$\frac{\frac{P^l_i}{d(n_i,n_j)^\alpha}}{Noise + \sum_{k \in T \setminus \{i\}} \frac{P^l_k \times \delta^k}{d(n_k,n_j)^\alpha}} \geq \beta \times f^{s,t}_{i,j} \tag{7}$$

Our mathematical program then becomes:

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

Proof sketch: We prove that MaxKeyEstabFlow is NP-hard by 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 the $3CNF$ form. We define a reduction from MAX3SAT to MaxKeyEstabFlow in two steps. First, given a boolean formula in the $3CNF$ 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 (a.k.a. both variables can not be set as TRUE). 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 4.1. Thus, the objective of finding a truth assignment to the variables so that maximum 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 the 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 the MAX3SAT (φ) = c $\Leftrightarrow$ MaxKeyEstabFlow ($\tau(\varphi)$) = c.m. (see the appendix for formal proofs).

4.3 Problem P2.2 Mathematical Programming Formulation

Definition 2 (MinCostKeyEstabFlow Problem P2.2). *Given the auxiliary graph representation $G_A = (V_A, E_A)$ 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, 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 the 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 (receiver and transmitter), (ii) flow symmetry, (iii) utilization (receiver and transmitter), and (iv) reception quality. *Flow bound* is additional constraint which requires total flow granted by the flow assignment should be at least $\mathcal{X}$. Thus:

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

Our mathematical program becomes:

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

$C_{i,j} = E^T + E^R$ 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.

Proof sketch: We prove that MinCostKeyEstabFlow is NP-hard by using a reduction from the *Weighted MAX3SAT* problem where each clause has a weight, and the problem is to maximize the sum of the weights of satisfied clauses. The *Weighted MAX3SAT* is both NP-hard and MAX-SNP [15] problem. We use similar approach as in MaxKeyEstabFlow to show that MinCostKeyEstabFlow problem is both NP-hard and MAX-SNP (see the appendix for formal proofs).

5 Conclusion and Discussions

Focus of this work is first to formulate the key establishment problem in wireless sensor networks together with the physical layer properties, then to analyze its complexity. We present mathematical programming formulations *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 [9] which is the richest class of all. Our results show that post-deployment key establishment in distributed wireless sensor networks is a hard problem. Most key management schemes trusting on post deployment key establishment for secure connectivity may not be feasible and applicable to practical solutions. Research should focus on making efficient use of deployment knowledge, or on developing deterministic key management schemes (such as [16–18]) which can ensure that any pair of nodes secure their communication using symmetric or asymmetric keys without explicit key establishment flows.

References

1. Zhang, J., Varadharajan, V.: Wireles ssensor network key management survey and taxonomy. J. Netw. and Com. App. **33** (2010)
2. Camtepe, S.A., Yener, B.: Key Management. In: Wireless Sensor Network Security. IOS Press, Cryptology and Information Security Series (2008)
3. Santi, P., Maheshwari, R., Resta, G., Das, S., Blough, D.M.: Wireless link scheduling under a graded sinr interference model. In: ACM FOWANC. (2009)
4. Goussevskaia, O., Oswald, Y.A., Wattenhofer, R.: Complexity in geometric sinr. In: ACM MobiHoc. (2007)
5. Moscibroda, T., Wattenhofer, R., Zollinger, A.: Topology control meets sinr: the scheduling complexity of arbitrary topologies. In: ACM MobiHoc. (2006)
6. Bhandari, V., Vaidya, N.H.: Secure capacity of multi-hop wireless networks with random key pre-distribution. In: IEEE MCN. (2008)
7. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: ACM CCS. (2002)
8. Liu, D., Ning, P., Du, W.: Group-based key predistribution for wireless sensor networks. ACM TOSN **4**(2) (2008)
9. Hochbaum, D.S.: Approximation Algorithms for NP-Hard Problems. PWS Publishing Company (1997)

10. Naor, D., Gusfield, D., Martel, C.: A fast algorithm for optimally increasing the edge connectivity. SIAM J. of Comp. **26**(4) (1997)
11. Nagamochi, H., Ibaraki, T.: Augmenting edge-connectivity over the entire range in o(nm) time. J. Alg. **30**(2) (1999)
12. Benczúr, A.A., Karger, D.R.: Augmenting undirected edge connectivity in (n^2) time. In: ACM-SIAM SODA. (1998)
13. Savas, O., Alanyali, M., Yener, B.: Joint route and power assignment in asynchronous multi-hop wireless networks. In: MedHocNet. (2004)
14. Costa, M.C., Létocart, L., F.Roupin: Minimal multicut and maximal integer multiflow: a survey. Elsevier J. of Op. Res. **162** (2005)
15. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comp. and Sys. Sci. **43**(3) (1991)
16. Camtepe, S.A., Yener, B.: Combinatorial design of key distribution mechanisms for wireless sensor networks. IEEE/ACM TON **15**(2) (2007)
17. Blom, R.: An optimal class of symmetric key generation systems. In: EUROCRYPT. (1984)
18. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Advances in Cryptology. (1992)

A Proofs

Proof. (P2.1 MaxKeyEstabFlow is in NP-hard) We prove that MaxKeyEstabFlow is NP-Hard using a reduction from the 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 the $3CNF$ form with clauses $\{C_1, C_2, \ldots, C_m\}$. We define a reduction τ from MAX3SAT to MaxKeyEstabFlow in two steps. The first step reduces a MAX3SAT problem instance into a WSN problem instance, and the second step derives an auxiliary graph formulation.
Step 1: Given a boolean formula φ in the $3CNF$ 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 the RTS/CTS signalling to check availability of the channel at receivers.
3. Only the sensor nodes x_j and $\overline{x}_j$ create interference on each other (a.k.a. boolean variables x_j and $\overline{x}_j$ can not be *true* 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 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 key if variable x_j (a.k.a., $\overline{x}_j$) appears in i^{th} clause. All other pairs of key-chains should not share a key.

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 communication.
6. Place the sensor nodes on a unit disk area: (a) Draw $v \times v$ ($v = \lceil \sqrt{n} \rceil$) grid for n variables. (b) Each grid location should be a square of size $2\alpha I \times 2\alpha I$ where I is the distance below which SINR on receiving node due to other nodes is less than the threshold based on Formula 1. αI (for a constant α) is the distance over which interference is negligible. (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 $d(x_j, \overline{x}_j) < I$. Thus, SINR on x_j as receiver can be less than the threshold only due to $\overline{x}_j$.

Step 2: Given a WSN deployment which is reduced from a boolean formula φ in the $3CNF$ form with n variables and m clauses, develop an auxiliary graph formulation as described in Section 4.1 and as illustrated in Figure 4:

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 unlimited.
3. Create set of flows $\mathcal{F} = \{(C_i, D_i), (D_i, C_i) | 1 \leq i \leq m\}$.

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, and then formulates it as 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 flows 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.

A solution to the problem instance $\tau(\xi)$ of MaxKeyEstabFlow in auxiliary graph representation can be converted to a 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 the 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

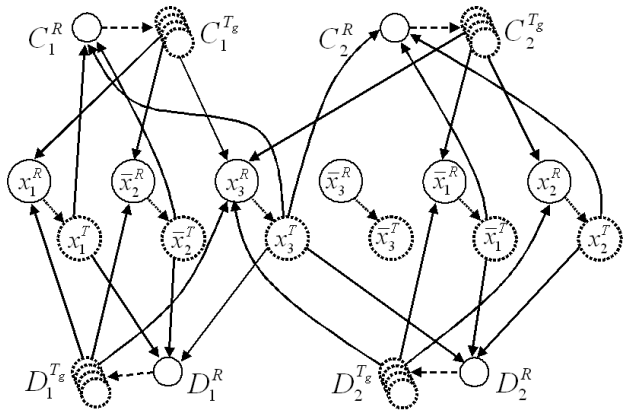

Fig. 4. Auxiliary graph $G_A = (V_A, E_A)$ reduced from sample boolean formula $\varphi = ((x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3))$. There is only one transmit power level for the nodes corresponding to the boolean variables. Nodes C_1, C_2, D_1, D_2 have L_{max} 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 unlimited capacities. Finally set of flow is $\mathcal{F} = \{(C_1, D_1), (C_2, D_2), (D_1, C_1), (D_2, C_2)\}$.

does not change the number of the satisfied clauses in ξ, but some satisfied clauses may 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 flows (C, D) (i.e. flows $(C, D), (D, C) \in \mathcal{F}$) which are granted. (1)$MAX3SAT \rightarrow MaxKeyEstabFlow$: assume that $\tau(\xi)$ has an optimal solution $d > c$. Then it would be possible to satisfy more than c clauses by simply setting $True$ value for the respective variables. This contradicts the fact that ξ has an optimal solution c. (2) $MaxKeyEstabFlow \rightarrow MAX3SAT$: assume that ξ has an optimal solution $d > c$. Then it would be possible to grant flow for d source-destination pairs without contradicting the interference constraint. This contradicts the fact that $\tau(\xi)$ has an optimal solution c. □

Definition 3. *[9, 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').*

Proof. (P2.1 MaxKeyEstabFlow is in MAX-SNP) MAX3SAT is a MAX-SNP problem [15] where its optimum c is a fraction equivalent to the maximum num-

ber 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. 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 flows that can be granted. 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 each of the corresponding transmitter nodes has one unit of flow, meaning that corresponding flow (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 $c.m$ flows granted. Each granted flow (C_i, D_i) means one satisfied clause C_i so that MAX3SAT(φ) $\geq c$. Thus:

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

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. (P2.2 MinCostKeyEstabFlow is in both NP-hard and MAX-SNP-hard) We use the *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 a both NP-Hard and MAX-SNP-Hard [15] 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 flows 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 $P = NP$. □