Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 5 (2007) 801–814
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Modeling adaptive node capture attacks in multi-hop wireless networks

Patrick Tague, Radha Poovendran *

Network Security Lab (NSL), Department of Electrical Engineering, University of Washington, PAC AE100R, Campus Box 352500, Seattle, WA 98195-2500, United States

Available online 19 January 2007

Abstract

We investigate the problem of modeling node capture attacks in heterogeneous wireless ad hoc and mesh networks. Classical adversarial models such as the Dolev–Yao model are known to be unsuitable for describing node capture attacks. By defining the amortized initialization overhead cost as well as the cost of capturing a node, we show that finding the node capture attack yielding the minimum cost can be formulated as an integer-programming minimization problem. Hence, there is no polynomial solution to find the minimum cost node capture attack. We show that depending on the adversary's knowledge of the constraint matrix in the integer-programming problem, different greedy heuristics can be developed for node capture attacks. We also show under what conditions privacy-preserving key establishment protocols can help to prevent minimum cost node capture attacks. Individual node storage randomization is investigated as a technique to mitigate the effect of attacks which are not prevented by the use of privacy-preserving protocols. It is shown that probabilistic heuristic attacks can be performed effectively even under storage randomization.

Keywords: Adversary modeling; Integer programming; Key establishment; Node capture attacks

1. Introduction

In order to provide secure network services in wireless ad hoc or mesh networks, nodes in the network must be able to collaboratively establish secure multi-hop and/or multi-path routes, using secure single-hop *links*, via key establishment. Hence, key establishment requires the use of authenticated protocols for discovering single-hop and multi-hop neighbors. The heterogeneous nature of wireless mesh networks [1] further requires the use of key establishment protocols which can be executed between nodes of varying capability.

Public-key cryptography is a possible solution for key establishment in ad hoc networks. Though recent work [2,3] demonstrates cases in which public-key cryptography can be implemented on some resource-constrained devices, it is not yet feasible for all multi-hop networks. In the absence of public-key protocols, key establishment must be performed using symmetric (shared) keys which

* Corresponding author. Tel.: +1 206 221 6512; fax: +1 206 543 3842.
E-mail addresses: tague@u.washington.edu (P. Tague), rp3@u.washington.edu (R. Poovendran).

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.01.002

are assigned to nodes prior to network deployment, a solution known as *key predistribution*.

An existing solution for symmetric key establishment is the assignment of a pairwise key to each pair of nodes in the network. Pairwise keys allow for secure key establishment with low complexity, but the storage requirement of $(N-1)$ pairwise keys for each of the N nodes may be prohibitive for large networks. Moreover, if the network is to be expanded via node addition, the existing nodes must be updated with pairwise keys for the added nodes. Thus, network expansion would require either $\mathcal{O}(N)$ communication overhead to update the existing nodes or reuse of at least some of the existing network keys.

Another extreme solution is the assignment of a global key to every node in the network. However, if an adversary is able to obtain the global key, the security of the entire network is compromised.

A promising approach to symmetric key establishment which attempts to balance the trade-offs between complexity, storage, and security is the assignment of each key to multiple nodes in the network. Through key reuse, the complexity of public-key protocols, the storage overhead of pairwise key protocols, and the easy compromise of global key protocols can be mitigated. Probabilistic key predistribution, as introduced in the seminal work [4], has been extensively studied in recent literature (e.g. [5–12]) and applied to classical key distribution techniques (e.g. [13–15]). Deterministic key predistribution techniques have also been investigated (e.g. [16–18]). For a formal treatment, the reader is referred to [12].

Most of the recent works have focused on the application of key predistribution in homogeneous networks (e.g. wireless sensor networks). However, the same principles can be applied to wireless mesh networks with heterogeneous structure by tailoring the network and security parameters (e.g. communication range, available key storage) to each class of nodes.

Due to the physical properties of an open wireless medium, wireless networks deployed in hostile environments are vulnerable to numerous attacks [19,20]. Since secure network services are built on the security of the key establishment protocol, an adversary can compromise key establishment and thus compromise the security of the services built on top of it.

An adversary eavesdropping on insecure protocol exchanges in the network can make decisions to selectively capture individual nodes. The attack of each node leads to the recovery of the cryptographic keys assigned to the node. When keys are reused, this further allows the adversary to eavesdrop on any secure communication links established using the recovered keys. Hence, such a *node capture attack* in key reusing networks leads to an effective wire-tapping adversary [21]. Due to heterogeneity in mesh networks, the adversary's decisions of which nodes to capture may also depend on the structure of each class of nodes.

Randomization techniques are often posed as defense mechanisms against attacks on cryptographic protocols. Hence, a natural question to ask is whether the randomization of the number of keys assigned to each node can be used to prevent attacks on the key establishment protocol.

1.1. Our contributions

In this paper, we aim to provide a formal characterization of node capture attacks. In addition, motivated by the use of randomization as a diversity mechanism in communication theory and a defense mechanism in cryptography, we study the application of storage randomization to key establishment. Our contributions are summarized as follows:

- We show that node capture attacks on networks of heterogeneous nodes can be modeled using an integer-programming minimization problem which generalizes the NP-hard set cover problem.
- We provide practical strategies for node capture attacks based on a known heuristic for the integer-programming minimization problem.
- We show that attacks in which an adversary aims to eavesdrop on specific secure links can be prevented by the use of privacy-preserving key establishment protocols.
- We show that attacks in which an adversary aims to recovery all existing cryptographic quantities in the network cannot be prevented by privacy-preserving protocols. We investigate the application of storage randomization techniques to key establishment protocols. We show that efficient heuristic attacks can be performed by an adaptive adversary in the presence of storage randomization.

The organization of this article is as follows. Assumptions about the network and adversary are stated in Section 2. Motivating examples are provided in Section 3. Node capture attacks are modeled in Section 4, and practical strategies for node

capture attacks are presented in Section 5. Storage randomization is presented in Section 6 including an analysis of the technique from the perspective of the adversary. Node capture attacks on a mesh network are formulated as examples in Section 7. The article is summarized in Section 8.

2. Preliminaries

The assumptions about the wireless ad hoc network and the adversaries present in the network are stated as follows.

2.1. Network model

The wireless ad hoc network is made up of a set of N nodes $\mathscr{N} = \{1, \ldots, N\}$ which may hold varying roles and vary in computational ability. A set $\mathscr{Y} = \{y_1, \ldots, y_R\}$ of keys or secrets is derived according to a given key predistribution scheme. For each node $n \in \mathscr{N}$, a set $J_n \subseteq \{1, \ldots, R\}$ is chosen, and a set S_n is derived from the subset of $\mathscr{Y}$ indexed by J_n. The sets J_n and S_n are assigned to node n prior to network deployment.

The key establishment protocol is assumed to be such that any pair of nodes $n_1, n_2 \in \mathscr{N}$ can compute the intersection set $J_{n_1} \cap J_{n_2}$ of indices. This intersection set indicates the elements of S_{n_1} and S_{n_2} which are used by nodes n_1 and n_2, respectively, to secure the pairwise link (n_1, n_2).

2.2. Adversary model

The capabilities of adversaries present in the ad hoc network are stated as follows. The adversary has sufficient computation, storage, and energy resources and can tap all links in the network to record any key establishment messages. The adversary is able to expend finite energy required to capture network nodes and recover cryptographic information from storage of captured nodes. Furthermore, an adversary who has captured a node n can actively participate in the key establishment protocol by assuming the identity of node n.

The primary goal of the adversary is the recovery of sufficient cryptographic information from a set of captured nodes in order to eavesdrop on and compromise a desired set of secure communication links in the wireless network. Attacks which can be mounted on network protocols and secure network services as a result of the node capture attack are not addressed in this article.

3. Motivating examples

The following examples are provided to illustrate the need to model node capture attacks on key establishment. In the examples, each node n is assigned a subset S_n of the set $\mathscr{Y} = \{k_1, k_2, \ldots, k_{12}\}$ of keys. The network consists of nodes $\mathscr{N} = \{1, 2, \ldots, 9\}$ and each set S_n is of size $|S_n| = 4$. The examples illustrate node capture attacks on existing key predistribution schemes using probabilistic and deterministic methods of key assignment. In both examples, the set J_n of indices is assumed to be computable by an adversary.

3.1. Probabilistic assignment

Selection of $S_n \subseteq \mathscr{Y}$ of size $|S_n| = 4$ is done by randomly choosing four elements of $\mathscr{Y}$ without replacement for each of the nine nodes. Details of such an assignment can be found in [4]. In this example, the resulting sets S_n are given by

$$S_1 = \{k_2, k_5, k_9, k_{10}\}, \quad S_2 = \{k_3, k_4, k_5, k_9\},$$
$$S_3 = \{k_1, k_5, k_7, k_9\}, \quad S_4 = \{k_1, k_2, k_4, k_6\},$$
$$S_5 = \{k_4, k_6, k_7, k_{11}\}, \quad S_6 = \{k_6, k_7, k_8, k_{10}\},$$
$$S_7 = \{k_1, k_3, k_8, k_{12}\}, \quad S_8 = \{k_5, k_7, k_9, k_{12}\},$$
$$S_9 = \{k_1, k_3, k_5, k_8\}.$$

The first attack is aimed at the recovery of $\mathscr{Y}$ by capturing the minimum number of nodes in $\mathscr{N}$. Due to the small number of nodes in this example, the adversary can exhaustively search for the optimal set of nodes to capture. For the given sets S_n, the optimal solution is to capture nodes 1, 5, and 7, as these nodes form a partition of the set $\mathscr{Y}$.

The second attack is aimed at the recovery of the elements of $\mathscr{Y}$ that maximize the number of secure links which can be eavesdropped. To achieve this goal, a value can be assigned to each node n equal to the number of secure links between remaining nodes that can be eavesdropped using the information recovered from n. For the sets S_n as given, the value is computed as in Table 1.

Table 1 suggests that the adversary capturing node 3 will be able to eavesdrop on nine secure links used by uncaptured nodes in the network. The adversary can continue the attack by updating the value of each node with respect to the information S_3 recovered from node 3, yielding the values given in Table 2. Table 2 further suggests capture of nodes 5 and 9 will allow the adversary to eavesdrop on the four secure links which remain.

Table 1
The value of each node n is computed as the number of secure links which can be eavesdropped using the information in S_n

Node n	Link (n_1,n_2)	Value
1	(2,3), (2,8), (8,9)	3
2	(1,3), (1,8), (1,9), (8,9)	4
3	(1,2), (1,8), (1,9), (2,8), (4,7), (4,9), (5,8), (6,8), (8,9)	9
4	(2,5), (3,7)	2
5	(2,4), (3,6), (4,6), (6,8)	4
6	(3,5), (5,8)	2
7	(3,4), (4,9), (6,9)	3
8	(1,2), (1,3), (1,9), (2,3), (3,5), (3,6)	6
9	(2,7), (3,4), (3,7), (4,7), (6,7)	5

The list of secure links (n_1,n_2) between nodes $n_1,n_2 \in \mathcal{N}$ which can be eavesdropped using the information in S_n is given. The value is given by the number of eavesdropped links.

Table 2
The value of each node n given in Table 1 is updated to reflect the capture of node 3 and the recovery of S_3

Node n	link (n_1,n_2)	Value
1	–	0
2	–	0
4	(2,5)	1
5	(2,4), (4,6)	2
6	–	0
7	(6,9)	1
8	(5,6)	1
9	(2,7), (6,7)	2

3.2. Deterministic assignment

Selection of $S_n \subseteq \mathcal{Y}$ of size $|S_n| = 4$ is performed by constructing a deterministic (v,b,r,k)-configuration [17] in which the set J_n is given by the nth block of the design. A (v,b,r,k)-configuration consists of b blocks, each containing k of the v total points, such that each of the v points appear in exactly r blocks. A (v,b,r,k)-configuration has the additional property that no two blocks intersect at more than 1 point. In the following example, the sets S_n determined by the blocks of a $(12,9,3,4)$-configuration are given by

$$
\begin{aligned}
S_1 &= \{k_1,k_4,k_7,k_{10}\},\\
S_2 &= \{k_1,k_5,k_8,k_{11}\},\\
S_3 &= \{k_1,k_6,k_9,k_{12}\},\\
S_4 &= \{k_2,k_4,k_9,k_{11}\},\\
S_5 &= \{k_2,k_5,k_7,k_{12}\},\\
S_6 &= \{k_2,k_6,k_8,k_{10}\},\\
S_7 &= \{k_3,k_4,k_8,k_{12}\},\\
S_8 &= \{k_3,k_5,k_9,k_{10}\},\\
S_9 &= \{k_3,k_6,k_7,k_{11}\}.
\end{aligned}
$$

The first attack is aimed at the recovery of $\mathcal{Y}$ by capturing the minimum number of nodes in $\mathcal{N}$. By inspection of the $(12,9,3,4)$-configuration, any two sets S_{n_1} and S_{n_2} intersect at exactly one point, so the optimal node capture attack is to select any 4 nodes such that no three nodes share a common value.

The second attack is aimed at the recovery of the elements of $\mathcal{Y}$ that maximize the number of secure links which can be eavesdropped. To achieve this goal, a value can be assigned to each node using the same technique as described in Section 3.1. Since every point in the $(12,9,3,4)$-configuration appears in exactly three blocks, the recovery of each $y \in \mathcal{Y}$ allows the adversary to eavesdrop on exactly one secure link used by uncaptured nodes. Thus, the capture of any one node will allow the adversary to eavesdrop on four links, and the optimal solution is again to capture nodes such that no three nodes share a common value. Hence, the two attacks are identical for this specific block design.

3.3. Need for a new model

The examples given in Sections 3.1 and 3.2 as well as those in [22,23] indicate that node capture attacks can not be modeled using classical adversary models [24].

4. Modeling node capture attacks

Motivated by the examples in Section 3, a mathematical model for node capture attacks is presented relating the information sought by the adversary to the sets S_n assigned to nodes in $\mathcal{N}$. In the given model, attacks are quantified with respect to the cost and the benefit of an attack to the adversary. The node capture attack which achieves the desired benefit for the minimum cost to the adversary is mapped to an NP-hard minimization problem, and approximate solutions are investigated through the use of a known heuristic.

4.1. Formulation of the model

The cost of mounting a node capture attack is a metric of particular interest to an adversary with bounded resources. In what follows, a cost metric is presented, and a node capture attack model aiming to minimize the cost of an attack is formulated.

In planning a node capture attack, the adversary may be required to record message exchanges in the key establishment protocol throughout the network.

In addition, the adversary may be required to perform additional initialization overhead prior to the attack. The amortized cost associated with these operations is denoted by c_0.

In executing a node capture attack, the adversary may be required to compute the identity of each node to capture via a node capture attack algorithm. In addition, the adversary must expend energy to physically capture each node and access information from the node's storage. Due to variations in hardware complexity and the level of tamper-resistance present in each node, the resources required for physical capture and access of a node may be different for each node. The cost associated with computation, capture, and access of a node $n \in \mathcal{N}$ is denoted by c_n, which is necessarily positive.

Letting the binary variable x_n indicate whether a node $n \in \mathcal{N}$ has been captured and collecting the variables c_n and x_n into the $N \times 1$ vectors $\mathbf{c}$ and $\mathbf{x}$, respectively, the cost associated with a given node capture attack can be expressed as

$$C(\mathbf{x}) = c_0 + \mathbf{c}^{\mathrm{T}}\mathbf{x}. \tag{1}$$

For a general attack, let $\mathscr{Z} = \{z_1, \ldots, z_M\}$ denote the collection of M elements (e.g. elements or subsets of $\mathscr{Y}$) of interest to the adversary. In order to plan a node capture attack, the adversary must characterize the relationship between each set S_n and each $z_i \in \mathscr{Z}$. The relationship can be characterized by defining a variable $a_{i,n}$ which is non-zero if and only if a subset of S_n aids in the recovery of z_i. The variables $a_{i,n}$ can be collected into the $M \times N$ constraint matrix $\mathbf{A}$ representing the attack.

Given a node capture vector $\mathbf{x}$ and a constraint matrix $\mathbf{A}$, the quantity $a_{i,n}x_n$ denotes the contribution of the node $n \in \mathcal{N}$ to the recovery of the element z_i. The total contribution to the recovery of z_i is thus computed as

$$\sum_{n \in \mathcal{N}} a_{i,n}x_n. \tag{2}$$

Depending on the structure of the key predistribution scheme, the adversary may be required to obtain a certain amount of information about an element z_i before it can be recovered. Hence, let s_i denote the quantity such that z_i is recovered by the adversary if and only if

$$\sum_{n \in \mathcal{N}} a_{i,n}x_n \geqslant s_i. \tag{3}$$

The $M \times 1$ vector $\mathbf{s}$ of quantities s_i for $i = 1, \ldots, M$ thus determines the sufficient condition for the success of a node capture attack. In addition, the variables s_i can be used to express the adversary's preference for certain elements $z_i \in \mathscr{Z}$. If the elements z_i are of equal importance to the adversary, the variables $a_{i,n}$ and s_i for a given i can be normalized by a variable γ_i so that s_i/γ_i is equal for all values $i = 1, \ldots, M$. Note that such a normalization has no effect on the inequality in (3). If an element z_i is of greater importance to the adversary, however, the variables $a_{i,n}$ and s_i for the given i can be appropriately weighted to reflect the relative importance of the element z_i.

Based on the preferential treatment of elements $z_i \in \mathscr{Z}$, the adversary may be interested in the weighted fraction of the M elements in $\mathscr{Z}$ which have been recovered for a given node capture vector $\mathbf{x}$, where the weight of each z_i is given by the normalized quantities s_i. This metric, referred to as the *benefit* of the node capture attack, can be computed as follows.

Define the $M \times 1$ binary vector $v_{\mathbf{x}}$ such that the ith element $v_{\mathbf{x},i}$ is equal to *1* if an only if (3) is true. The benefit $B(\mathbf{x})$ of a given attack is then given by

$$B(\mathbf{x}) = \frac{\mathbf{v}_{\mathbf{x}}^{\mathrm{T}}\mathbf{s}}{\|\mathbf{s}\|_1}, \tag{4}$$

where $\|\cdot\|_1$ denotes the ℓ_1 (absolute vector sum) norm [25]. Note that when the variables s_i are equal for all $i = 1, \ldots, M$, the benefit $B(\mathbf{x})$ given in (4) reduces to

$$B(\mathbf{x}) = \frac{\|\mathbf{v}_{\mathbf{x}}\|_1}{M}. \tag{5}$$

For a given node capture attack, an adversary will be primarily interested in determining the vector $\mathbf{x}$ for which the inequality (3) is satisfied for all i, i.e.

$$\mathbf{A}\mathbf{x} \geqslant \mathbf{s}. \tag{6}$$

Based on the definition of the benefit metric in (4), the condition (6) is satisfied if and only if the benefit is $B(\mathbf{x}) = 1$. Furthermore, the adversary can determine the vector $\mathbf{x}$ which satisfies (6) for the minimum cost given by (1) using the following minimization problem:

$$\begin{aligned}
\text{Given:}\quad & \mathbf{A} = [a_{i,n}]_{M \times N}, \quad a_{i,n} \geqslant 0,\\
& \mathbf{s} = [s_i]_{M \times 1}, \quad s_i \geqslant 0,\\
& \mathbf{c} = [c_n]_{N \times 1}, \quad c_n > 0.\\
\text{Minimize:}\quad & \mathbf{c}^{\mathrm{T}}\mathbf{x}\\
\text{such that}\quad & \mathbf{A}\mathbf{x} \geqslant \mathbf{s},\\
& \mathbf{x} \in \{0,1\}^N.
\end{aligned}$$

The minimization problem corresponds to a special case of the integer-programming minimization problem in [26, Section 3] and, hence, can be analyzed using the results therein.

The benefit $B(\mathbf{x})$ of an attack can be averaged over the set of attack realizations to yield the expected benefit $b(x)$ as a function of the number of captured nodes x given by

$$b(x) = E[B(\mathbf{x}) | \|\mathbf{x}\|_1 = x], \tag{7}$$

where $E[\cdot]$ denotes the expected value over all realizations of the attack. The metric in (7) is equivalent to the widely-used (e.g. [4–7,10,11,17,12]) measure of the *resilience* of the key predistribution scheme to a node capture attack. Hence, this metric is of interest to both the adversary and the designer of the key predistribution scheme.

4.2. Analysis of the model

In order to perform a node capture attack with minimal cost, the adversary must be able to solve the minimization problem given in Section 4.1. However, the minimization problem is a special case of that in [26, Section 3] which contains the NP-hard set cover problem [26–28] as a special case. The special case corresponding to the set cover problem is that in which $\mathbf{A}$ is binary and each element of $\mathbf{s}$ and $\mathbf{c}$ is equal to 1. Hence, by the reduction property [28], determining the node capture attack with minimal cost is an NP-hard problem.

The hardness of finding the minimal cost attack suggests that approximate solutions are required. The heuristic solution provided in [26, Section 3] thus provides algorithms for node capture attacks which approximate the optimal solution. Attack algorithms are provided for two cases, depending on the information available about the constraint matrix $\mathbf{A}$ to the adversary.

4.2.1. Attacks when the constraint matrix is known

When each entry $a_{i,n}$ of the constraint matrix $\mathbf{A}$ is available to the adversary, the heuristic solution provided in [26, Section 3] provides Algorithm 1. In the approximate solution given by Algorithm 1, $update(a_{i,n}, \hat{n})$ denotes the function used to update each entry of $\mathbf{A}$ with respect to the information obtained from the captured node $\hat{n}$.

Algorithm 1. Node Capture Attack

Given: $\mathbf{A} = [a_{i,n}]_{M\times N}$, $a_{i,n} \geqslant 0$
Given: $\mathbf{s} = [s_i]_{M\times 1}$, $s_i \geqslant 0$
Given: $\mathbf{c} = [c_n]_{N\times 1}$, $c_n > 0$
$\mathbf{x} \leftarrow \mathbf{0}$
While $\mathbf{s} \neq \mathbf{0}$ do
  $\hat{n} \leftarrow \arg\max_{n\in\mathscr{N}} (\sum_{i=1}^{M} a_{i,n}/c_n)$
  $x_{\hat{n}} \leftarrow 1$
  $s_i \leftarrow s_i - a_{i,\hat{n}}$ for all i
  $a_{i,n} \leftarrow update(a_{i,n}, \hat{n})$ for all i, n
end while

Due to [26, Theorem 2.1], if $\mathbf{x}^*$ is the optimal solution of the minimization problem in Section 4.1 and $\mathbf{x}$ is the solution obtained by Algorithm 1, then

$$\frac{\mathbf{c}^{\mathrm{T}}\mathbf{x}}{\mathbf{c}^{\mathrm{T}}\mathbf{x}^*} \leqslant h\left(\max_{n\in\mathscr{N}} \sum_{i=1}^{M} a_{i,n}\right), \tag{8}$$

where $h(d)$ is the dth harmonic number given by

$$h(d) = \sum_{i=1}^{d} \frac{1}{i}. \tag{9}$$

The bound in (8), referred to as the ratio bound in [27], can be used to determine the increase in cost which can be realized using Algorithm 1.

The use of Algorithm 1 can be interpreted with respect to node capture attacks as follows. Given the constraint matrix $\mathbf{A}$, an adversary using Algorithm 1 can explicitly compute the vector $\mathbf{x}$ and determine the set $\{n \in \mathscr{N} : x_n = 1\}$ of nodes to capture to approximate the solution to the minimization problem given in Section 4.1. Furthermore, the deviation from the minimum achievable cost as in (1) can be bounded using (8).

At each step of the algorithm, the heuristic solution chooses the node $\hat{n}$ which, when scaled by the cost $c_{\hat{n}}$, contributes maximally to the recovery of $\mathscr{Z}$. In terms of the set cover problem, this corresponds to the well-known heuristic which chooses the set of minimum overlap or maximum non-overlap. Attacks that are based on only partial information about the constraint matrix are investigated as follows.

4.2.2. Attacks with partial information about the constraint matrix

If each entry $a_{i,n}$ of the constraint matrix $\mathbf{A}$ is not explicitly available to the adversary, the adversary may still be able to use the heuristic solution provided in [26, Section 3]. Noting that the heuristic

choice of $\hat{n}$ in line 6 of Algorithm 1 depends only on the column sums A_n of $\mathbf{A}$ given by

$$A_n = \sum_{i=1}^{M} a_{i,n}, \tag{10}$$

the remainder of the algorithm is investigated to see if the individual values $a_{i,n}$ are required. Since the adversary will be able to compute the value $a_{i,\hat{n}}$ after capturing the node $\hat{n}$, the update of $\mathbf{s}$ in line 8 of Algorithm 1 can be performed without the values $a_{i,n}$. Hence, if the adversary can compute the update function $update(A_n, \hat{n})$ after capturing the node $\hat{n}$, an equivalent approximate attack can be performed using Algorithm 2.

Algorithm 2. Node Capture Attack

1: **Given**: $A_n \geqslant 0$ for $n \in \mathscr{N}$
2: **Given**: $\mathbf{s} = [s_i]_{M\times 1}$, $s_i \geqslant 0$
3: **Given**: $\mathbf{c} = [c_n]_{N\times 1}$, $c_n > 0$
4: $\mathbf{x} \leftarrow \mathbf{0}$
5: **While** $\mathbf{s} \neq \mathbf{0}$
6: $\quad \hat{n} \leftarrow \arg\max_{n\in\mathscr{N}}(A_n/c_n)$
7: $\quad x_{\hat{n}} \leftarrow 1$
8: $\quad s_i \leftarrow s_i - a_{i,\hat{n}}$ for all i
9: $\quad A_n \leftarrow update(A_n, \hat{n})$ for all n
10: **end while**

Algorithm 2 is of particular interest if a privacy-preserving key establishment protocol (e.g. that mentioned in [4]) based on a cryptographic proof-of-knowledge [29] is used. Such a protocol does not allow the adversary to compute the set J_n for each $n \in \mathscr{N}$.

For example, as will be shown in Section 5.1, if $S_n \subseteq \mathscr{Y} = \mathscr{Z}$, $|S_n| = K$ for all $n \in \mathscr{N}$, and $\mathbf{A}$ is a binary matrix, the adversary will be able to compute the number of elements τ_n in each set S_n that are already known. Hence, though the individual values $a_{i,n}$ cannot be determined, the adversary can compute the column sums A_n of $\mathbf{A}$ given by

$$A_n = K - \tau_n. \tag{11}$$

5. Attack strategies

In what follows, two node capture attack strategies are formulated with respect to the network and adversary models presented in Section 2 and the node capture attack algorithms presented in Section 4.2. The strategies discussed herein are the *set coverage* and *subset coverage* strategies, which are so named because of their relationships to the well-known set cover problem [27,26]. The set coverage and subset coverage strategies are similar to those applied in the examples of Section 3.

5.1. Set coverage

The set coverage strategy is modeled according to the well-known set cover problem. In this strategy, the collection $\mathscr{Z}$ of items sought by the adversary is equal to the set of secrets $\mathscr{Y}$. The adversary's primary goal is to capture a set of nodes whose sets S_n *cover* the set $\mathscr{Y}$ and thus can be used to compromise the security of every secure link in the network. In this attack, each element $y \in \mathscr{Y}$ is of equal importance to the adversary, so the elements s_i are equal.[1]

A set coverage attack can be formulated using the minimization problem in Section 4.1 and Algorithm 1 as follows. Each entry s_i of the vector $\mathbf{s}$ is equal to the number of elements t_i derived from y_i which must be obtained to recover the secret y_i. For example, the value t_i can be equal to the threshold of a secret-sharing scheme [30,13,14,7,6,10,11] applied to the elements of $\mathscr{Y}$. Each entry $a_{i,n}$ of the binary matrix $\mathbf{A}$ is equal to 1 if and only if an element in S_n was derived from $y_i \in \mathscr{Y}$. Hence, the column sum A_n of the matrix $\mathbf{A}$ is equal to the number of elements in S_n which are unknown to the adversary. To perform a set coverage attack using Algorithm 1, the key establishment protocol must allow the adversary to compute the set J_n for each node $n \in \mathscr{N}$. The following result characterizes the performance of a set coverage attack using Algorithm 2.

Lemma 5.1. *Given any key establishment protocol such that $|S_n|$ is computable by the adversary for each $n \in \mathscr{N}$, a set coverage attack can be performed deterministically using Algorithm 2.*

Proof 1. Let J denote the set of indices of elements in $\mathscr{Y}$ recovered by the adversary from previously captured nodes. Since the adversary has obtained all of the information stored within each captured node, the intersection set $J \cap J_n$ is necessarily computable for each $n \in \mathscr{N}$, as the adversary can simply play the role of each captured node in the key establishment protocol. Algorithm 2 can then be performed using the values $A_n = |S_n| - \tau_n$, similar to (11). Note that the result does not require $|S_n|$ to be fixed for all $n \in \mathscr{N}$. □

[1] A simplified version of this strategy was used to develop a probabilistic attack in [22].

The primary implication of Lemma 5.1 is that the use of a privacy-preserving key establishment protocol based on a cryptographic proof-of-knowledge [29] does not prevent the adversary from performing set coverage attacks. Techniques to mitigate the effect of set coverage attacks are investigated further in Section 6 using the result of Lemma 5.1.

5.2. Subset coverage

The subset coverage strategy is also modeled according to the well-known set cover problem. In this strategy, each element in the collection $\mathscr{Z}$ of items sought by the adversary is a subset $z_{(n_1,n_2)}$ of $\mathscr{Y}$ indexed by the intersection set $J_{n_1} \cap J_{n_2}$ and corresponding to the secrets used by nodes $n_1, n_2 \in \mathscr{N}$ in establishing a secure link. Since the same elements of $\mathscr{Y}$ can be used by multiple pairs of nodes in the network, $\mathscr{Z}$ is a multi-set of subsets of $\mathscr{Y}$ whose union is not necessarily all of $\mathscr{Y}$. Under this strategy, the adversary's primary goal is to capture a set of nodes whose sets S_n *cover* as many of the subsets of $\mathscr{Y}$ appearing in $\mathscr{Z}$ as is possible, corresponding to the compromise of as many secure links as in the network as is possible.

A subset coverage attack can be formulated using the minimization problem in Section 4.1 and Algorithm 1 as follows. Similar to that of the set coverage strategy, each entry $s_{(n_1,n_2)}$ of the vector $\mathbf{s}$ is equal to the number of elements $t_{(n_1,n_2)}$ derived from $z_{(n_1,n_2)}$ which must be obtained to recover the set $z_{(n_1,n_2)}$. Each entry $a_{(n_1,n_2),n}$ of the binary matrix $\mathbf{A}$ is equal to 1 if and only if $J_{n_1} \cap J_{n_2} \subseteq J_n$. Furthermore, to perform a subset coverage attack using Algorithm 1, the key establishment protocol must allow the adversary to compute the set J_n.

If the adversary cannot compute the set J_n for each node $n \in \mathscr{N}$, it is impossible to determine the subsets $z_{(n_1,n_2)}$ of $\mathscr{Y}$ corresponding to each secure link. Furthermore, there is no method for computing or updating the column sums A_n of the matrix $\mathbf{A}$. Hence, subset coverage attacks can be prevented by the use of a privacy-preserving key establishment protocol.

5.3. Variations on subset coverage

Due to the fact that the condition $J_{n_1} \cap J_{n_2} \subseteq J_n$ is a relatively strong condition if $|J_{n_1} \cap J_{n_2}|$ is large, the subset coverage strategy can be generalized by assigning to each $a_{(n_1,n_2),n}$ a fractional value. Each entry $a_{(n_1,n_2),n}$ in the rational matrix $\mathbf{A}$ is thus defined as the ratio of $|J_{n_1} \cap J_{n_2} \cap J_n|$ to $|J_{n_1} \cap J_{n_2}|$, corresponding to the fraction of the set $z_{(n_1,n_2)}$ to which the set S_n aids in the recovery. The use of this *fractional subset coverage strategy* compensates for various cases that the subset coverage strategy cannot. For example, if $z_{(n_1,n_2)} = \{k_1, k_2, k_3\}$, but there is no node $n \in \mathscr{N} \setminus \{n_1, n_2\}$ such that $J_{n_1} \cap J_{n_2} \subseteq J_n$, there may exist a pair of nodes $n_3, n_4 \in \mathscr{N} \setminus \{n_1, n_2\}$ such that $J_{n_1} \cap J_{n_2} \subseteq J_{n_3} \cap J_{n_4}$. Though intuitively, the fractional subset coverage strategy may lead to the compromise of secure links at a lower rate with respect to the number of captured nodes, it will lead to the compromise of a larger number of secure links overall.

A second variation on the subset coverage strategy is formulated due to the extensive computational cost involved in computing the $\binom{N}{2}$ intersection sets z_i and $(N-2)|\mathscr{Z}|$ values $a_{i,n}$ in initializing the attack. A slightly sub-optimal benefit can be traded for a significant decrease in computation by estimating the number of links which can be secured as a function of each $y \in \mathscr{Y}$. By assuming that $|J_{n_1} \cap J_{n_2}| \leqslant 1$ for all $n_1 \neq n_2 \in \mathscr{N}$, the *fast subset coverage* strategy is formulated. Letting $\lambda(y_i)$ denote the number of nodes such that $i \in J_n$, an adversary who captures t_i of the $\lambda(y_i)$ nodes with $i \in J_n$ can potentially compromise $\binom{\lambda(y_i)-t_i}{2}$ secure links. Fast subset coverage attacks can thus be modeled by letting $\mathscr{Z} = \mathscr{Y}, a_{i,n} = \binom{\lambda(y_i)-t_i}{2}$ if and only if $i \in J_n$, and $s_i = t_i\binom{\lambda(y_i)-t_i}{2}$.

Similar to the case of subset coverage attacks, fractional and fast subset coverage attacks can only be formulated if the index sets J_n are computable by the adversary and, thus, can be prevented by the use of a privacy-preserving key establishment protocol.

6. Storage randomization to mitigate set coverage attacks

Lemma 5.1 in Section 5 implies that the only way to prevent a set coverage attack on a key predistribution scheme is to prevent the adversary from computing the value A_n as given in (11). This first requires the use of a privacy-preserving key establishment protocol. Since the value τ_n in (11) can be computed deterministically for any key establishment protocol, this also requires the property that

the adversary cannot compute the value $|S_n|$. One way to achieve this property is to randomize the number of elements $|S_n|$ assigned to each node $n \in \mathcal{N}$.

Storage randomization can be achieved by imposing a probability distribution on $|S_n|$. Letting $\kappa_n = |S_n|$, the distribution on κ_n is denoted by $P(k) = \Pr[\kappa_n = k]$.

6.1. Probability distribution on κ_n

It is important to note that the design of the probability distribution $P(k)$ describing κ_n must be done prior to key assignment and network deployment. Thus, the designer has no prior knowledge about the adversary. However, the adversary may be able to perform attacks based on statistical analysis of κ_n based on the designed distribution $P(k)$. Hence, the designer must choose $P(k)$ in order to maximize the uncertainty of κ_n.

The uncertainty of the random variable κ_n for each node n can be quantified by the entropy $H(\kappa_n)$ given by

$$H(\kappa_n) = -\sum_k P(k)\log_2 P(k). \tag{12}$$

Assuming that κ_n is allowed to vary probabilistically over an interval $[\kappa_{\min}, \kappa_{\max}]$, the entropy of κ_n can be maximized [31] by choosing $P(k)$ as a uniform distribution over the interval.

The randomization of $\kappa_n = |S_n|$ impacts the one-hop connectivity of the secure wireless network, considering both the physical constraints of limited communication range and the logical constraints of the existence of a secure link. Hence, the secure network connectivity model formalized in [12] is generalized as follows.

6.2. Effect of storage randomization on secure network connectivity

The probability that a wireless network is k-connected using only secure one-hop links is provided by [12, Theorem 2] as a function of the average number of nodes D which are able to establish a secure one-hop link with a given node. The value of D is given by [12, Theorem 2] as $D = (N-1)p$ where p is the probability that a given node n with $|S_n| = K$ is such that J_n intersects J_{n_1} non-trivially for a node n_1. Since D is computed with respect to a single node, D, p, and K can be respectively replaced with D_n, p_n, and κ_n such that $D_n = (N-1)p_n$ where p_n. The average number of nodes $\bar{d}$ which are able to establish a secure one-hop link with a given node is thus computed by averaging D_n over all possible values of κ_n. The probability p_n is given by [12, Theorem 7] as

$$p_n = 1 - f^{\kappa_n}, \tag{13}$$

where $f = (N-\mu)/(N-1)$ and μ is the average number of nodes sharing an element of $\mathcal{Y}$. The quantity $\bar{d}$ is thus computed as

$$\bar{d} = \sum_{k=1}^{\kappa_{\max}} (N-1)(1-f^k)P(k), \tag{14}$$

$$= (N-1)\left(1 - \sum_{k=\kappa_{\min}}^{\kappa_{\max}} \frac{f^k}{\kappa_{\max} - \kappa_{\min} + 1}\right). \tag{15}$$

The connectivity of the network is then given by applying the result of [12, Theorem 2] with $D = \bar{d}$ given by (15). Hence, the connectivity of the network can be characterized as a function of the parameters in the connectivity model of [12] and the additional parameters $\kappa_{\min}$ and $\kappa_{\max}$.

6.3. Effect of storage randomization on adaptive adversary

The security offered by the use of storage randomization can be evaluated by probabilistically analyzing the behavior of κ_n as a function of the prior distribution $P(k)$, the fraction z/R of $\mathcal{Y}$ known to the adversary, and the overlap value τ_n computable for each $n \in \mathcal{N}$.

To allow for random variation of κ_n in the heuristic attack algorithms of Section 4.2, the column sum A_n of the matrix $\mathbf{A}$ given in (11) can be generalized to

$$A_n = \kappa_n - \tau_n. \tag{16}$$

The heuristic in line 6 of Algorithm 2 is thus generalized as

$$\hat{n} = \arg\max_{n \in \mathcal{N}} \frac{\kappa_n - \tau_n}{c_n}, \tag{17}$$

which is a function of the random variables κ_n. Hence, (17) can only be computed by the adversary if κ_n can be appropriately estimated.

6.4. Analysis of storage randomization

A suitable estimate for κ_n can be computed as the conditional expected value of κ_n given τ_n and the fraction z/R of $\mathcal{Y}$ which is known. The conditional

probability $P(k|\tau) = \Pr[\kappa_n = k|\tau_n = \tau]$ describing the desired behavior can be computed using the law of total probability [32] as

$$P(k|\tau) = \frac{Q(\tau|k)P(k)}{\sum_{k=\kappa_{\min}}^{\kappa_{\max}} Q(\tau|k)P(k)}, \tag{18}$$

where $Q(\tau|k) = \Pr[\tau_n = \tau|\kappa_n = k]$. Assuming as in [12] that the sets J_n for $n \in \mathscr{N}$ are determined independently, the probability $Q(\tau|k)$ that the adversary with z elements of $\mathscr{Y}$ shares τ of the k elements assigned to a node n is given by the binomial probability

$$Q(\tau|k) = \binom{k}{\tau}(1-q)^{\tau}q^{k-\tau}, \tag{19}$$

where $q = 1 - z/R$. Thus (18) can be written explicitly as

$$P(k|\tau) = \frac{\binom{k}{\tau}(1-q)^{\tau}q^{k-\tau}P(k)}{\sum_{t=\kappa_{\min}}^{\kappa_{\max}}\binom{t}{\tau}(1-q)^{\tau}q^{t-\tau}P(t)}. \tag{20}$$

The conditional expected value $\kappa(\tau)$ of κ_n given the overlap value $\tau_n = \tau$ can then be computed as

$$\kappa(\tau) = \sum_{k=\kappa_{\min}}^{\kappa_{\max}} kP(k|\tau), \tag{21}$$

where $P(k|\tau)$ is given in (20). Furthermore, since $P(k)$ is assumed to be a uniform distribution, the conditional expected value $\kappa(\tau)$ is given by (21) and (20) as

$$\kappa(\tau) = \sum_{k=\kappa_{\min}}^{\kappa_{\max}} \frac{k\binom{k}{\tau}q^k}{\sum_{t=\kappa_{\min}}^{\kappa_{\max}}\binom{t}{\tau}q^t}. \tag{22}$$

6.5. Probabilistic node capture attacks

The heuristic step in the node capture algorithm given by (17) can thus be approximated by estimating κ_n by the conditional expected value $\kappa(\tau_n)$ given τ_n. The probabilistic version of the heuristic is thus given by

$$\hat{n} = \arg\max_{n\in\mathscr{N}} \frac{\kappa(\tau_n) - \tau_n}{c_n}, \tag{23}$$

which yields the maximum expected contribution to the success of the node capture attack. The node capture attack given by Algorithm 2 is thus generalized in Algorithm 3.

Algorithm 3. Node Capture Attack

1: **Given**: $\tau_n \geqslant 0$ for $n \in \mathscr{N}$
2: **Given**: $\mathbf{s} = [s_i]_{M\times 1}$, $s_i \geqslant 0$
3: **Given**: $\mathbf{c} = [c_n]_{N\times 1}$, $c_n \geqslant 0$
4: $\mathbf{x} \leftarrow \mathbf{0}$
5: **While** $\mathbf{s} \neq \mathbf{0}$
6: $\quad \hat{n} \leftarrow \arg\max_{n\in\mathscr{N}}((\kappa(\tau_n) - \tau_n)/c_n)$
7: $\quad x_{\hat{n}} \leftarrow 1$
8 $\quad s_i \leftarrow s_i - a_{i,\hat{n}}$ for all i
9 $\quad \tau_n \leftarrow update(\tau_n, \hat{n})$ for all n
10 **end while**

Note that the probabilistic node capture attack in Algorithm 3 does not require the adversary to deterministically compute any information about the key predistribution scheme. The only requirement is that the adversary knows the values $\kappa_{\min}$ and $\kappa_{\max}$ in order to compute $\kappa(\tau)$ using (22) for the given value $q = 1 - z/R$.

The adversary may be able to compute the value τ^* which maximizes the function $\kappa(\tau) - \tau$ for a given value of q. In this case, the computation required in line 6 may be reduced. If the adversary finds a node n such that $\tau_n = \tau^*$, the remaining nodes do not need to be examined, as the maximum expected contribution is achieved by node n.

7. Case study

In what follows, the effect of node capture attacks is investigated for selected examples using the attack model proposed in Section 4. The network of interest for the examples herein is a wireless mesh network consisting of a three-class hierarchy of network nodes in which κ_n is a function of node class. Such a network is depicted in Fig. 1.

The highest class of nodes $\mathscr{N}_1$ in the hierarchy consists of N_1 similar nodes which make up the mesh backbone. Backbone nodes are assumed to be equipped with a degree of tamper-resistance which makes them very costly to attack, and the cost c_n associated with the capture of each backbone node is equal to a constant C_1.

The second class of nodes $\mathscr{N}_2$ consists of N_2 similar nodes which serve as clusterheads in the mesh network. Clusterheads are assumed to have a lesser degree of tamper-resistance than backbone nodes, and the cost c_n associated with the capture of each clusterhead is equal to a constant C_2.

The third class of nodes $\mathscr{N}_3$ consists of N_3 similar mobile sensor nodes. These nodes are assumed to

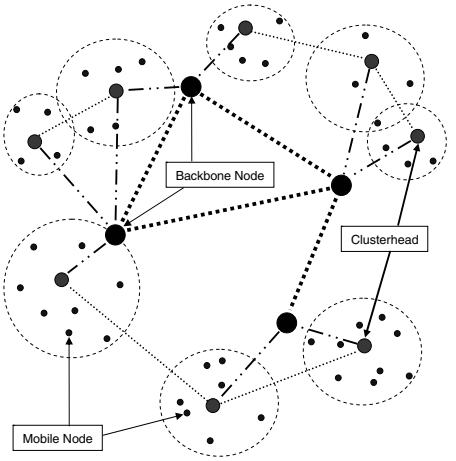

Fig. 1. A wireless mesh network consisting of a three-class hierarchy of network nodes is illustrated.

have no tamper-resistance, and the cost c_n associated with the capture of each clusterhead is equal to a constant C_3.

The set of network nodes is thus represented by $\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3$, consisting of $N = N_1 + N_2 + N_3$ nodes. Due to the structure of the network, it is assumed that $N_1 \ll N_2 \ll N_3$ and $C_1 \gg C_2 \gg C_3$.

7.1. Fractional subset coverage attack

The first example illustrates an attack in which the set $S_n \subseteq \mathcal{Y}$ of elements assigned to node n is of size $\kappa_n = K_i$ if $n \in \mathcal{N}_i$ for $i = 1,2,3$. The goal of the adversary in this example is to maximize the *fraction of communications between clusterheads and backbone nodes which can be eavesdropped* by capturing nodes in $\mathcal{N}$. This goal corresponds to the fractional subset coverage strategy presented in Section 5.3 modeled as follows.

The collection $\mathcal{Z}$ sought by the adversary consists of non-empty subsets $z_{(n_1,n_2)} = S_{n_1} \cap S_{n_2}$ where $n_1 \in \mathcal{N}_1$ and $n_2 \in \mathcal{N}_2$. The elements $a_{(n_1,n_2),n}$ of the matrix $\mathbf{A}$ are thus defined as

$$a_{(n_1,n_2),n} = \frac{|z_{(n_1,n_2)} \cap S_n|}{|z_{(n_1,n_2)}|}, \tag{24}$$

and the elements $s_{(n_1,n_2)}$ of $\mathbf{s}$ are equal to 1. Furthermore, the function $update(a_{(n_1,n_2),n}, \hat{n})$ in line 9 of Algorithm 1 is given by

$$update(a_{(n_1,n_2),n}, \hat{n}) = a_{(n_1,n_2),n} - a_{(n_1,n_2),\hat{n}} + \frac{|z_{(n_1,n_2)} \cap S_n \cap S_{\hat{n}}|}{|z_{(n_1,n_2)}|}. \tag{25}$$

The adversary can then mount the heuristic attack given by Algorithm 1. In this attack, the benefit $B(\mathbf{x})$ which results is exactly equal to the fraction of secure communication links between clusterheads and backbone nodes which can be eavesdropped by capturing the nodes indicated by the vector $\mathbf{x}$. Hence, Algorithm 1 allows the adversary to heuristically minimize the cost required in order to eavesdrop on all possible secure links of interest.

The benefit which results from this attack is simulated for the parameters $N_1 = 5$, $N_2 = 20$, $N_3 = 500$, $K_1 = 100$, $K_2 = 25$, $K_3 = 10$, and $R = 1{,}000$. The benefit $B(\mathbf{x})$ is illustrated in Fig. 2 as a function of $\|\mathbf{x}\|_1$ and indicating the number of captured

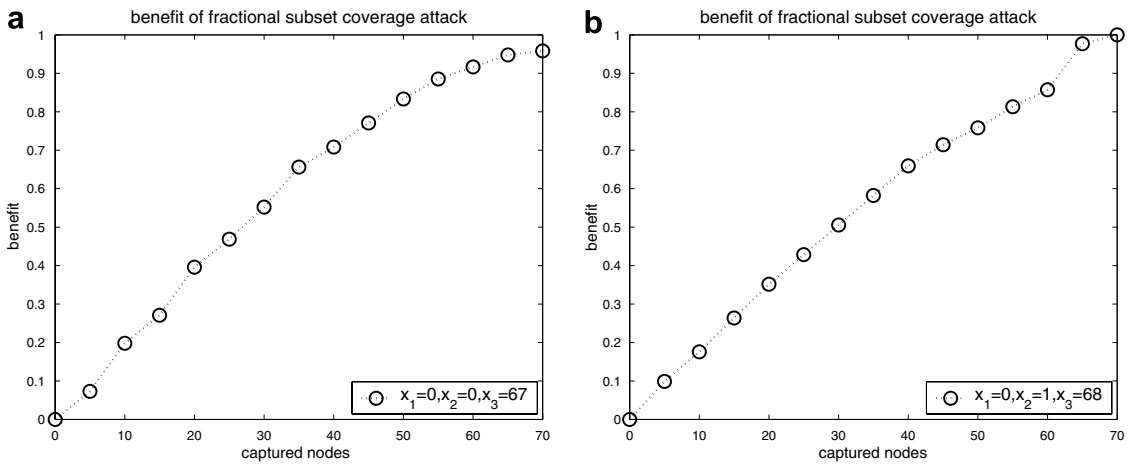

Fig. 2. The benefit of the attack formulated in Section 7.1 is simulated. In (a), the cost of capturing each node is such that $C_1 = 10C_2 = 100C_3$. In (b), the cost of capturing each node is such that $C_1 = 5C_2 = 20C_3$. x_i denotes the number of captured nodes in $\mathcal{N}_i$.

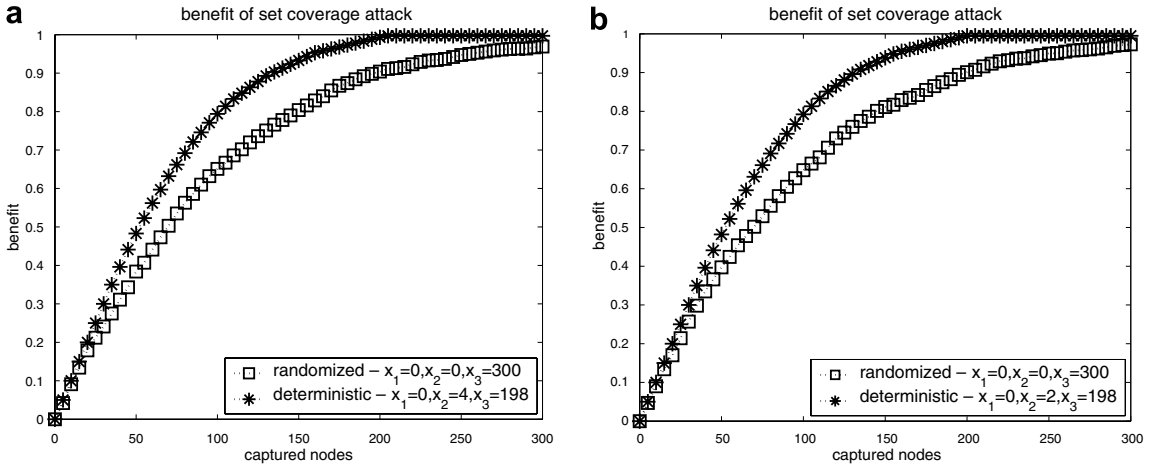

Fig. 3. The benefit of the attack formulated in Section 7.2 is simulated. In (a), the cost of capturing each node is such that $C_1 = 10C_2 = 100C_3$. In (b), the cost of capturing each node is such that $C_1 = 5C_2 = 20C_3$. x_i denotes the number of captured nodes in $\mathcal{N}_i$.

nodes x_i in each class $\mathcal{N}_i$. To demonstrate the effect of node heterogeneity, the attack is performed for two sets of costs C_i for $i = 1, 2, 3$.

The less variable costs in Fig. 2(b) explain the sudden jump in the attack benefit toward the end of the attack. After several nodes in $\mathcal{N}_3$ have been captured, the number of secure links that can be eavesdropped due to the capture of the node in $\mathcal{N}_2$ exceeds that of any other node in $\mathcal{N}_3$ by at least the factor C_2/C_3.

7.2. Randomized set coverage attack

The second example illustrates an attack in which the set $S_n \subseteq \mathscr{Y}$ of elements assigned to node n is of size κ_n is uniformly distributed in an interval $[\kappa_{i,\min}, \kappa_{i,\max}]$ for $n \in \mathcal{N}_i$. The goal of the adversary is to maximize the *fraction of $\mathscr{Y}$ which is recovered* by capturing nodes in $\mathcal{N}$. This goal corresponds to the set coverage strategy with randomized storage discussed in Section 6 modeled as follows.

The collection $\mathscr{X}$ sought by the adversary is equal to the set $\mathscr{Y}$ and each element s_i in $\mathbf{s}$ is equal to 1. Due to storage randomization, the adversary must use Algorithm 3 in which the function $update(\tau_n, \hat{n})$ corresponds to the execution of the key establishment protocol by the adversary for each node n. The benefit $B(\mathbf{x})$ which results is exactly equal to the fraction of $\mathscr{Y}$ recovered by the adversary. Hence, Algorithm 3 allows the adversary to probabilistically minimize the cost required in order to recover the set $\mathscr{Y}$.

The benefit which results from this attack is simulated for the parameters $N_1 = 5$, $N_2 = 20$, $N_3 = 500$, and $R = 1000$ such that $\kappa_{1,\min} = 80$, $\kappa_{1,\max} = 120$, $\kappa_{2,\min} = 15$, $\kappa_{2,\max} = 35$, $\kappa_{3,\min} = 5$, and $\kappa_{3,\max} = 15$. The benefit $B(\mathbf{x})$ is illustrated in Fig. 3 as a function of $\|\mathbf{x}\|_1$ and indicating the number of captured nodes x_i in each class $\mathcal{N}_i$. To demonstrate the effect of node heterogeneity, the attack is performed for two sets of costs C_i for $i = 1, 2, 3$.

Due to the variation in κ_n and the use of the expected value of $\kappa(\tau)$, the randomized cases in Fig. 3(a) and (b) are much less likely to select a node in $\mathcal{N}_2$ or $\mathcal{N}_3$ with higher cost. This is due to the suppression of statistical outliers which results from the use of $\kappa(\tau)$ instead of κ_n. Note also that the statistical heuristic leads to a decay in the benefit of the attack as the number of captured nodes increases. This is because the benefit achieved from an individual node is not guaranteed as in the deterministic case.

8. Conclusions

In this paper, we presented a mathematical model for node capture attacks on key establishment protocols in heterogeneous wireless ad hoc and mesh networks. By characterizing the cost of capturing each node and the contribution of each node to the attack success, attacks are formulated using an integer-programming minimization problem. We conclude that there is no polynomial solution that can determine the node capture attack with minimum cost for heterogeneous or homogeneous networks. An efficient heuristic algorithm for node capture attacks was thus presented using a known heuristic for the integer-programming minimization

problem. We showed that attacks using a subset coverage strategy can be prevented through the use of a privacy-preserving key establishment protocol. We also investigated storage randomization as a technique to mitigate set coverage attacks. We conclude that even in the presence of storage randomization, the adversary can perform a probabilistic heuristic via statistical analysis at an increased cost. We also observed that the probabilistic heuristic outperforms the random capture of nodes. Future work includes determination of appropriate privacy-preserving key establishment protocols and further investigation of mitigation techniques for node capture attacks.

Acknowledgements

This work is supported in part by the following grants: ONR YIP, N00014-04-1-0479; ARO PECASE, W911NF-05-1-0491; NSA/DoD IASP Fellowship; and ARL Collaborative Technology Alliance (CTA) grant, DAAD19-01-2-0011.[2]

References

[1] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.

[2] N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz, Comparing elliptic curve cryptography and RSA on 8-bit CPUs, in: Proceedings of the Sixth Workshop on Cryptographic Hardware and Embedded Systems (CHES'04), Cambridge, MA, USA, 2004, pp. 119–132.

[3] V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, S.C. Shantz, Sizzle: a standards-based end-to-end security architecture for the embedded internet, in: Proceedings of the Third IEEE Conference on Pervasive Computing and Communications (PerCom'05), Kauai, HI, USA, 2005, pp. 247–256.

[4] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the Ninth ACM Conference on Computer and Communications Security (CCS'02), Washington, DC, USA, 2002, pp. 41–47.

[5] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 2003, pp. 197–213.

[6] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, in: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03), Washington, DC, USA, 2003, pp. 42–51.

[7] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03), Washington, DC, USA, 2003, pp. 52–61.

[8] M. Ramkumar, N. Memon, R. Simha, Pre-loaded key based multicast and broadcast authentication in mobile ad-hoc networks, in: Proceedings of the IEEE Conference on Global Communications (GLOBECOM'03), San Francisco, CA, USA, 2003, pp. 1405–1409.

[9] M. Ramkumar, N. Memon, An efficient random key predistribution scheme, in: Proceedings of the IEEE Conference on Global Communications (GLOBECOM'04), Dallas, TX, USA, 2004, pp. 2218–2223.

[10] D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, ACM Transactions on Information and System Security 8 (1) (2005) 41–77.

[11] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, A. Khalili, A pairwise key predistribution scheme for wireless sensor networks, ACM Transactions on Information and System Security 8 (2) (2005) 228–258.

[12] P. Tague, R. Poovendran, A general probabilistic model for improving key assignment in wireless networks, in: Proceedings of the Fourth International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'06), Boston, MA, USA, 2006.

[13] R. Blom, An optimal class of symmetric key generation systems, in: Advances in Cryptology: Proceedings of the EUROCRYPT'84, LNCS 209, Paris, France, 1984, pp. 335–338.

[14] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly-secure key distribution for dynamic conferences, in: Advances in Cryptology: Proceedings of the CRYPTO'92, LNCS 740, Santa Barbara, CA, USA, 1992, pp. 471–486.

[15] T. Leighton, S. Micali, Secret-key agreement without public-key cryptography, in: Advances in Cryptology: Proceedings of the CRYPTO'93, LNCS 773, Santa Barbara, CA, USA, 1993, pp. 456–479.

[16] S.A. Çamtepe, B. Yener, Combinatorial design of key distribution mechanisms for distributed sensor networks, in: Proceedings of the Ninth European Symposium on Research in Computer Security (ESORICS'04), LNCS 3193, Sophia Antipolis, France, 2004, pp. 293–308.

[17] J. Lee, D.R. Stinson, A combinatorial approach to key predistribution for distributed sensor networks, in: Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC'05), Los Angeles, CA, USA, 2005, pp. 1200–1205.

[18] H. Chan, A. Perrig, PIKE: Peer intermediaries for key establishment in sensor networks, in: Proceedings of the 24th Conference of the IEEE Communications Society (INFOCOM'05), Miami, FL, USA, 2005, pp. 524–535.

[19] F. Stajano, R. Anderson, The resurrecting ducking: Security issues for ad-hoc wireless networks, in: Proceedings of the Seventh Annual International Workshop on Security Protocols, Cambridge, UK, 1999, pp. 172–194.

[2] This document was prepared through collaborative participation in the Communications and Networks Consortium sponsored by the US Army Research Laboratory under the Collaborative Technology Alliance Program, DAAD19-01-2-0011. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the US Government.

[20] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D. Culler, SPINS: Security protocols for sensor networks, in: Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MobiCom'01), Rome, Italy, 2001, pp. 189–199.
[21] A.D. Wyner, The wire-tap channel, Bell System Technical Journal 54 (8) (1975) 1355–1387.
[22] D. Huang, M. Mehta, D. Medhi, L. Harn, Location-aware key management scheme for wireless sensor networks, in: Proceedings of the Second ACM Workshop on Security of Ad-Hoc and Sensor Networks (SASN'04), Washington, DC, USA, 2004, pp. 29–42.
[23] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: Proceedings of the 2005 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 2005.
[24] D. Dolev, A.C. Yao, On the security of public-key protocols, IEEE Transactions on Information Theory IT 29 (2) (1983) 198–208.
[25] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
[26] G. Dobson, Worst-case analysis of greedy heuristics for integer programming with nonnegative data, Mathematics of Operations Research 7 (4) (1982) 515–531.
[27] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, McGraw-Hill, 2000.
[28] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, 1979.
[29] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, Florida, 2001.
[30] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612–613.
[31] T.M. Cover, J.A. Thomas, Elements of Information Theory, second ed., John Wiley & Sons Inc., 2006.
[32] W. Feller, An Introduction to Probability Theory and Its Applications, vol. 1, John Wiley & Sons Inc., New York, 1957.

Patrick Tague is a Ph.D. student in the Electrical Engineering Department at the University of Washington in Seattle. He received his M.S. degree from the same department in 2007 and his B.S. degrees in Mathematics and Computer Engineering from the University of Minnesota in Minneapolis in 2003. His current research interests include analytical modeling of practical key distribution systems for wireless ad-hoc networks and attacks and defense mechanisms for distributed dynamic network systems.

Radha Poovendran is an Associate Professor at the Electrical Engineering Department of the University of Washington. He received his Ph.D. in Electrical Engineering from the University of Maryland, College Park in 1999. He is the founding director of the Network Security Laboratory (NSL) at the University of Washington. His research interests are in the areas of applied cryptography for multiuser environment, wireless networking, and applications of Information Theory to security. He is a recipient of the Faculty Early Caeer Awards from the NSF CAREER (2001), ARO YIP (2002), ONR YIP (2004), and the PECASE (2005) for his research contributions to the field of multiuser security.