

Space-Efficient Keying in Wireless Communication Networks

Neeraj Mittal

Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75083, USA
neerajm@utdallas.edu

Abstract

We investigate the problem of assigning keys to nodes in a network such that each node is able to communicate with every other node in a secure manner. We propose a key assignment scheme that assigns at most $\lceil \log n \rceil^2$ keys to each node, where n is the number of nodes in the network. The key that two nodes should use to securely communicate with each other is derived from at most $\lceil \log n \rceil$ keys and can be computed in $O(\log n)$ time. Finally, the collusion resistance of our scheme degrades gracefully as the number of colluding nodes increase in contrast to a recently proposed key assignment scheme that has zero collusion resistance (that is, two colluding nodes can eavesdrop on all communication in the network). To our knowledge, the key assignment scheme presented in this paper is the most efficient secure key assignment scheme in terms of space that has been proposed so far.

Key words: secure communication, key pre-distribution, poly-logarithmic key assignment, complete bipartite graph, complete graph, collusion resistance

1. Introduction

Communication in wireless networks, including sensor networks and ad-hoc networks, is particularly vulnerable to eavesdropping because the communication medium is broadcast in nature and an adversary can easily overhear messages exchanged in the network. Communication between two nodes can be secured by encrypting messages using a secret shared by the two nodes. Nodes in the network may also be vulnerable to capture by an adversary. In such cases, it is desirable to encrypt messages using a secret that is only known to the two nodes involved in the communication but is unknown to any other node. This gives rise to the *secure key assignment problem* in which each node is assigned a set of keys. The key a node uses to securely

communicate with another node is derived from a subset of keys it is carrying such that at least one of the keys in the subset is missing from the keys assigned to any other node [3, 1, 4].

The secure key assignment problem also arises during *secure multihop communication*. If two nodes that wish to communicate are not close to each other, then messages exchanged between them have to be routed through other nodes in the network. It may be required that the intermediate nodes forwarding a message cannot learn the contents of the message and, moreover, cannot create messages that incorrectly appear to be from the sender [7].

A simple solution to the secure key assignment problem is to associate a unique key with every pair of nodes that can communicate with each other. If the network contains n nodes and every pair of nodes can potentially communicate (that is, the communication graph is fully connected), then each node has to carry $n - 1$ keys. We are interested in a key assignment scheme that *minimizes* the number of keys each node has to carry [7, 1, 4]. Space efficiency is important in networks where each node has a relatively small memory to store its assigned keys. Examples of such networks include sensor networks [2, 8], ad-hoc networks [5, 9] and mobile networks [6, 10].

Related Work: Gong and Wheeler [3] presented a grid-based scheme for key assignment in which each node has to carry only $O(\sqrt{n})$ keys. Kulkarni *et al.* proposed a variation of this grid-based scheme in [7], and also showed that the scheme is optimal under the assumption that no two nodes share more than two keys. Aiyer *et al.* proposed a family of $\lceil \log n \rceil$ key assignment schemes in [1]. The k th scheme in the family assigns $O(k(k-1)\sqrt[k]{n})$ keys to each node in the network, where $1 \leq k \leq \lceil \log n \rceil$. Thus, when $k = \lceil \log n \rceil$, each node carries $O(\log^2 n)$ keys. They also proved that any key assignment scheme has to assign at least $\lceil \log n \rceil$ keys to each node in network. Further, they gave a *non-constructive* proof for the existence of a scheme in which each node carries only $12\lceil \log n \rceil$ keys.

Recently, Gouda *et al.* [4] presented several *logarithmic* key assignment schemes for certain classes of communication graphs including star graphs, acyclic graphs, limited-cycle graphs and planar graphs. Their schemes assign $O(\log d)$ keys to each node, where d is the maximum degree of a node in the network. They proved that their schemes are *optimal* within a constant factor.

Our Contributions: We propose a secure key assignment scheme for a complete communication graph that assigns $\lceil \log n \rceil^2$ keys to each node. The key that a node has to use to securely communicate with another node is derived from at most $\lceil \log n \rceil$ keys stored at the node and can be computed in $O(\log n)$ time. Aiyer *et al.*'s scheme when $k = \lceil \log n \rceil$ [1], on the other hand, assigns $4^{\lceil \log n \rceil} (\lceil \log n \rceil - 1)$ keys to each node in the network. Therefore our scheme uses only *one-fourth* the amount of space used by the Aiyer *et al.*'s scheme. Further, in Aiyer *et al.*'s scheme the key that two nodes should use to communicate with each other securely is derived from as many as $\lceil \log n \rceil^2$ keys and requires $O(\log^2 n)$ time to compute.

Our scheme has another advantage over Aiyer *et al.*'s scheme (when $k = \lceil \log n \rceil$). Their scheme has *zero collusion resistance*, which means that as few as two nodes can combine their assigned keys together and listen to all communication occurring in the network. Our scheme, on the other hand, has much better collusion resistance. The r -collusion resistance of our scheme defined as the minimum fraction of channels that are still secure when r nodes collude and denoted by $\rho.r$ is lower bounded by:

$$\rho.r \geq \frac{1}{\psi.r} \left(1 - \frac{r}{2\psi.r}\right) \left(1 - \frac{\psi.r - 1}{n - 1}\right)$$

where $2 \leq r \leq \frac{n}{4}$, $\psi.r$ is the *smallest power of two larger than or equal to r* , and $n \geq 4$. When $r \ll n$, $\frac{\psi.r - 1}{n - 1} \approx 0$ implying that $\rho.r \approx \frac{1}{\psi.r} \left(1 - \frac{r}{2\psi.r}\right)$. Observe that, when two nodes collude, *at least one-quarter* of the channels are still secure with our scheme (for sufficiently large n) but all channels may be compromised with Aiyer *et al.*'s scheme.

Organization of the Paper: The rest of the paper is organized as follows. We describe our secure key assignment scheme and analyze its space- and time-complexities in Section 2. We analyze the collusion resistance of our scheme in Section 3. We present our conclusions and outline directions for future research in Section 4.

2. Key Assignment Scheme for Complete Graph

We model the pairs of nodes that can potentially communicate with each other using a *communication graph*. There

is an edge between two nodes in the communication graph if and only if the two nodes can potentially exchange messages with each other. (Note that, by definition, a communication graph is symmetric.) Sometimes, when describing our scheme, we use the phrase “a channel has been made secure” to mean that keys have been assigned to the two nodes incident on the channel such that the communication along the channel is resilient against eavesdropping by another node in the network.

We first describe a scheme for solving the secure key assignment problem when the communication graph is a complete bipartite graph. This scheme is *optimal* within a constant factor. We then use this scheme as a subroutine to derive an efficient solution to the secure key assignment problem when the communication graph is the complete graph.

2.1. Keying in Complete Bipartite Graphs

Our key assignment scheme for a complete bipartite graph is an extension of Gouda *et al.*'s logarithmic key assignment scheme for a star graph [4]. Before describing our key assignment scheme for a complete bipartite graph, we describe an algorithm to associate a unique key with every node in a set S using a key pool of size $2 \cdot \lceil \log |S| \rceil$ [4].

2.1.1. Associating a unique key with a node using a logarithmic key pool

Consider a set of nodes S . We assign a *unique* identifier to every node in S from the range $[0, |S|)$. Clearly, the identifier assigned to a node contains $\lceil \log |S| \rceil$ bits. For convenience, let $s = \lceil \log |S| \rceil$. Let $S.i$ denote the node in S with identifier i , where $0 \leq i < |S|$. We create two *mutually exclusive* key pools denoted by $K.S$ and $\overline{K}.S$. Each key pool contains s keys. For a key pool P , let $P.i$ denote the i th key in the key pool, where $0 \leq i < |P|$.

For each node in S , we select a subset of s keys from $K.S \cup \overline{K}.S$. The subset for node $S.i$ is denoted by $D.(S, i)$, and is used to “derive” the key associated with node $S.i$. Let $b.(i, j)$ denote the j th most significant bit in the binary representation of i . The subset $D.(S, i)$ is constructed as follows:

```

D.(S, i) := ∅;
for each j ∈ [0, s) do
  if (b.(i, j) = 1) then add K.S.j to D.(S, i);
  else add  $\overline{K}.S.j$  to D.(S, i); endif;
endfor;

```

It can be easily verified that:

$$D.(S, i) = D.(S, j) \quad \equiv \quad i = j$$

Let $key.(S, i)$ denote the *unique* key associated with node $S.i$. The key $key.(S, i)$ is obtained by taking *bit-wise exclusive-or* of all keys in $D.(S, i)$.

2.1.2. Assigning keys to nodes in a complete bipartite graph

A complete bipartite graph is given by two mutually exclusive sets of nodes A and B such that there is an edge between every pair of nodes in $A \times B$. (Note that, since a communication graph is symmetric, we can also use $B \times A$ to denote the set of edges.) We denote the complete bipartite graph on A and B as $A \bowtie B$. For convenience, $a = \lceil \log|A| \rceil$ and $b = \lceil \log|B| \rceil$. Let $common.(i, j)$ denote the symmetric key that the two nodes $A.i$ and $B.j$, where $0 \leq i < |A|$ and $0 \leq j < |B|$, use to communicate with each other in a secure manner. The symmetric key $common.(i, j)$ is given by:

$$common.(i, j) \triangleq key.(A, i) \oplus key.(B, j)$$

where \oplus denotes the bit-wise exclusive-or operation. We now describe the set of keys that are assigned to each node in the network, also referred to as its *key ring*. Let $R.(A, i)$ (respectively, $R.(B, j)$) denote the key ring for node $A.i$ (respectively, $B.j$). The key ring for node $A.i$ is given by:

$$R.(A, i) = \{key.(A, i)\} \cup K.B \cup \overline{K}.B$$

whereas the key ring for node $B.j$ is given by:

$$R.(B, j) = \{key.(B, j)\} \cup K.A \cup \overline{K}.A$$

Now, suppose node $A.i$ wants to communicate with node $B.j$. Node $A.i$ can generate the key $common.(i, j)$ by first computing the set $D.(B, j)$ from $K.B \cup \overline{K}.B$ using the algorithm described above. It can then compute the key $common.(i, j)$ by taking bit-wise exclusive-or of all keys in $\{key.(A, i)\} \cup D.(B, j)$. Likewise, node $B.j$ can easily compute the key $common.(i, j)$ in a manner ‘‘symmetric’’ to that of node $A.i$. We refer to this scheme as **KACBG** (**Key Assignment for Complete Bipartite Graph**).

Security from eavesdropping: Our key assignment scheme is secure from eavesdropping because no other node can generate the key $common.(i, j)$. Specifically, no other node in A can generate the key $common.(i, j)$ because it does not know $key.(A, i)$ (although it can generate $key.(B, j)$). Likewise, no other node in B can generate the key $common.(i, j)$ because it does not know $key.(B, j)$ (although it can generate $key.(A, i)$). Therefore we have:

Theorem 1 *KACBG is secure.*

Space-complexity: The key pools $K.A$ and $\overline{K}.A$ contain a keys each. Likewise, key pools $K.B$ and $\overline{K}.B$ contain b keys each. Therefore, we have:

Theorem 2 *KACBG assigns $2b + 1$ keys to a node in A and $2a + 1$ keys to a node in B , where $a = \lceil \log|A| \rceil$ and $b = \lceil \log|B| \rceil$.*

The scheme is optimal within a constant factor because, as shown in [4], a node with degree d has to store at least $\lceil \log d \rceil$ keys. Clearly, each node in A has degree $|B|$ and each node in B has degree $|A|$. Thus, we have:

Theorem 3 *KACBG is optimal within a constant factor with respect to space-complexity.*

Time-complexity: To compute a key for communicating with its neighbor, a node in A selects b keys from $K.B \cup \overline{K}.B$. Likewise, a node in B selects a keys from $K.A \cup \overline{K}.A$. Therefore, we have:

Theorem 4 *The key a node in A (respectively, B) uses to communicate with its neighbor in B (respectively, A) is derived from $b + 1$ (respectively, $a + 1$) keys and can be computed in $O(b + 1)$ (respectively, $O(a + 1)$) time.*

Remark 1 The key assignment scheme described here is optimal within a constant factor even if the bipartite graph is not complete as long as (1) every node in A has degree $\Theta(|B|^c)$ for some constant c and (2) every node in B has degree $\Theta(|A|^c)$ for some constant c . (The constants may be different for different nodes.) This is because $\Theta(\log(|A|^c)) = \Theta(\log|A|)$ and $\Theta(\log(|B|^c)) = \Theta(\log|B|)$ if c is a constant. For example, the scheme is optimal within a constant factor even if each node in A has degree only $\sqrt{|B|}$ and each node in B has degree only $\sqrt{|A|}$. We refer to such bipartite graphs as *dense* bipartite graphs. Thus it follows that:

Theorem 5 *KACBG is optimal within a constant factor as long as the bipartite graph is dense.*

Our scheme can be easily extended to dense *layered* communication graphs as well. A layered graph is a generalization of a bipartite graph. In a layered graph, vertices are partitioned into layers and only vertices between successive layers can be connected by edges. \square

2.2. Keying in Complete Graphs

Our key assignment scheme for a complete graph is based on the divide and conquer approach. Let C denote the set of nodes in the network with $n = |C|$. For ease of exposition assume that n is a power of two. (If n is not

a power of two, space- and time-complexity expressions hold when $\log n$ is replaced with $\lceil \log n \rceil$.) When $n = 2$, the network contains only one channel. This channel can be made secure by assigning a symmetric key to the channel and storing that key on both nodes. Now, assume that $n > 2$. As in the key assignment scheme for a complete bipartite graph, we assign a unique identifier to every node in C from the range $[0, n)$. For a sequence of bits x , let $C_{\langle x \rangle}$ denote the subset of nodes in C whose identifier has prefix x in its binary representation. For example, when $n = 8$, $C_{\langle 0 \rangle} = \{C.0, C.1, C.2, C.3\}$, $C_{\langle 01 \rangle} = \{C.2, C.3\}$, and $C_{\langle 010 \rangle} = \{C.2\}$. We use \circ to denote the concatenation operator for two bit sequences.

We first consider the channels in $C_{\langle 0 \rangle} \times C_{\langle 1 \rangle}$. The communication subgraph induced by these channels is a complete bipartite graph. Therefore our key assignment scheme for a complete bipartite graph KACBG can be used to make these channels secure. We then apply this idea *recursively* to $C_{\langle 0 \rangle}$ and $C_{\langle 1 \rangle}$ to secure channels in $C_{\langle 0 \rangle} \times C_{\langle 0 \rangle}$ and $C_{\langle 1 \rangle} \times C_{\langle 1 \rangle}$, respectively.

In our scheme, the key assignment to a node can be viewed to occur in multiple stages. In each successive stage, a constant fraction (specifically, one half) of channels that are still insecure (that is, no keys have been assigned to those channels) are secured by an appropriate instance of KACBG. Specifically, in stage u , where $0 \leq u < \log n$, channels in $C_{\langle x \circ 0 \rangle} \times C_{\langle x \circ 1 \rangle}$ are secured for each x such that $|x| = u$. For example, in stage 2, channels in $C_{\langle 000 \rangle} \times C_{\langle 001 \rangle}$, $C_{\langle 010 \rangle} \times C_{\langle 011 \rangle}$, $C_{\langle 100 \rangle} \times C_{\langle 101 \rangle}$, and $C_{\langle 110 \rangle} \times C_{\langle 111 \rangle}$ are secured. This implies that there are $\log n$ stages and there is one instance of KACBG for every possible prefix of length at most $\log n - 1$. We use KACBG_x to refer to the instance for prefix x . Further, we use CBG_x to refer to the (complete) bipartite graph on which instance KACBG_x operates, that is, the bipartite graph $C_{\langle x \circ 0 \rangle} \bowtie C_{\langle x \circ 1 \rangle}$. Figure 1 illustrates our key assignment scheme when $n = 8$.

We now describe the behavior of our scheme with respect to a given node. Intuitively, in each successive stage, one half of a node's channels that are still insecure are secured by assigning new keys to the node. Let $\pi.(i, j)$ denote the prefix of length j in the binary representation of i . We define $\pi.(i, 0)$ to be ϵ (the empty prefix). (Observe that $\epsilon \circ x = x \circ \epsilon = x$.) Further, for a bit c , let \bar{c} denote the complement of c . Now, consider a node $C.i$. In stage u , where $0 \leq u < \log n$, the channels between the node $C.i$ and the nodes in the set $C_{\langle \pi.(i, u) \circ \bar{b}.(i, u+1) \rangle}$ are secured. For example, suppose $n = 8$ and $i = 5$. There are $\log 8 = 3$ stages, and the binary representation of 5 is 101. In stage 0, the channels between the node $C.5$ and the nodes in the set $C_{\langle 0 \rangle} = \{C.0, C.1, C.2, C.3\}$ are secured. In stage 1, the channels between the node $C.5$ and the nodes in the set $C_{\langle 11 \rangle} = \{C.6, C.7\}$ are secured. Finally, in stage 2,

the channels between the node $C.5$ and the nodes in the set $C_{\langle 100 \rangle} = \{C.4\}$ are secured.

Observe that, when securing channels between nodes in $C_{\langle x \circ 0 \rangle}$ and $C_{\langle x \circ 1 \rangle}$ for some prefix x , KACBG needs to assign an identifier to each node in $C_{\langle x \circ 0 \rangle}$ (respectively, $C_{\langle x \circ 1 \rangle}$) that is unique among all nodes in $C_{\langle x \circ 0 \rangle}$ (respectively, $C_{\langle x \circ 1 \rangle}$). However, we do not need to assign a separate identifier to a node in each stage. In fact, we can use $\log n - |x| - 1$ least significant bits in the *global identifier* assigned to a node in C as an identifier to be used by instance KACBG_x at stage $|x|$.

Computing the key for secure communication between two nodes:

We now describe how two nodes can compute the common symmetric key to communicate with each other securely. Consider two nodes $C.i$ and $C.j$ such that $i \neq j$. To compute the key that node $C.i$ can use to securely communicate with node $C.j$, node $C.i$ first computes the *longest* common prefix in the binary representations of i and j , say x . It can be verified that the channel between the two nodes is secured at stage $|x|$ by instance KACBG_x . Node $C.i$ computes the symmetric key for securing its channel with node $C.j$ using the rules of KACBG_x . It can be verified that this key is derived by taking bit-wise exclusive-or of $\log n - |x|$ keys, and can be computed in $O(\log n)$ time.

We refer to this scheme as KACG (Key Assignment for Complete Graph).

Security from eavesdropping: The key assignment scheme for a complete graph is secure from eavesdropping because every channel in the network is secured by some instance of KACBG which, in turn, is secure from eavesdropping. Therefore we have:

Theorem 6 *KACG is secure.*

Space-complexity: At stage u , where $0 \leq u < \log n$, each node is assigned $2(\log n - u - 1) + 1$ keys. Therefore, the total number of keys a node has to carry is given by:

$$\begin{aligned} & \sum_{u=0}^{\log n - 1} [2(\log n - u - 1) + 1] \\ &= 2 \sum_{u=0}^{\log n - 1} (\log n - u) - \log n \\ &= 2(1 + 2 + \dots + \log n) - \log n \\ &= \log^2 n \end{aligned}$$

It follows that:

Theorem 7 *KACG assigns at most $\log^2 n$ keys to each node.*

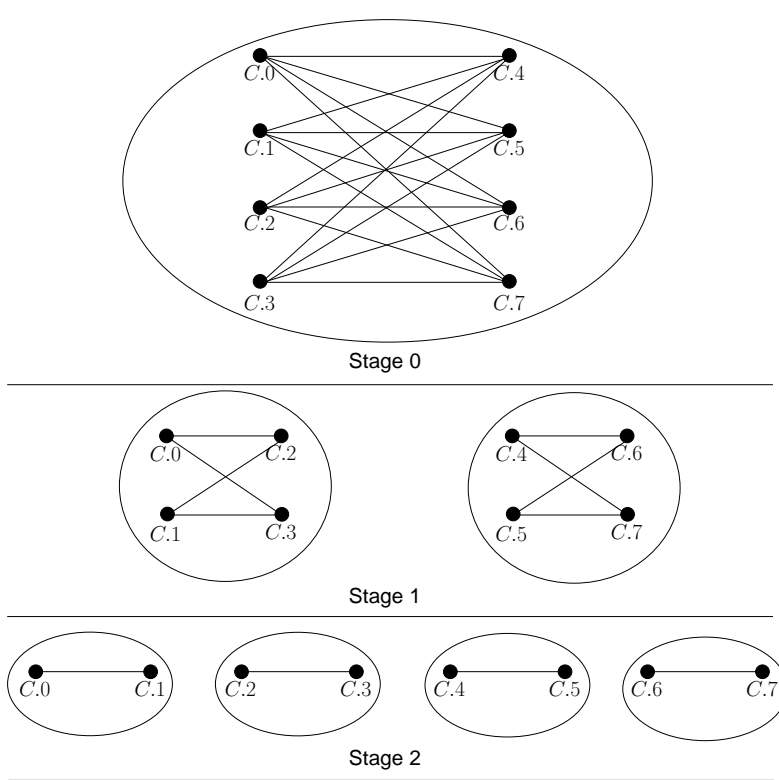


Figure 1. An illustration of our key assignment scheme for a complete graph when $n = 8$.

Time-complexity: If the channel between two nodes is secured at stage u , then the key associated with the channel is derived from $\log n - u$ keys. It follows that:

Theorem 8 *The key a node uses to communicate with its neighbor is derived from at most $\log n$ keys and can be computed in $O(\log n)$ time.*

3. Analysis of Collusion Resistance

In this section, we analyze the behavior of our scheme when two or more nodes collude together to eavesdrop on communication occurring in the network. We define the *r-collusion resistance* of a key assignment scheme as the *minimum* fraction of channels that are still secure in the network when r nodes collude. Since our key assignment scheme for a complete graph is derived from that for a complete bipartite graph, we first analyze the collusion resistance of the keying scheme for complete bipartite graphs.

3.1. Collusion Resistance of KACBG

Consider a complete bipartite graph $A \bowtie B$ on two mutually exclusive sets of nodes A and B . It can be easily verified that if any node in A colludes with any node in B ,

then the two nodes can compute $common.(i, j)$ for every i and j . Formally,

Lemma 9 *If any node in A colludes with any node in B , then all channels in $A \bowtie B$ are compromised.*

Observe that if both the colluding nodes belong to A (or to B), then all channels between *non-colluding* nodes are still secure. From Lemma 9, we have:

Theorem 10 *The r -collusion resistance of KACBG is zero for all $r \geq 2$.*

3.2. Collusion Resistance of KACG

Let C denote the set of nodes in the network with $n = |C|$. For the analysis, assume that n is a power of two. We first show that the collusion resistance of KACG is zero if at least $\frac{n}{2}$ nodes collude.

Theorem 11 *The r -collusion resistance of KACG is zero for all $r \geq \frac{n}{2}$.*

Proof: Consider a subset S of C that consists of all even-numbered nodes from C . Formally,

$$S \triangleq \{C.i \mid i \text{ is even}\}$$

Clearly, S contains at least $\frac{n}{2}$ nodes. We show that if nodes in S collude, then all channels in the network are compromised. It is sufficient to show that every channel between two odd-numbered nodes is compromised. Consider two odd-numbered nodes, say $C.i$ and $C.j$ with $i \neq j$ and let x be the longest common prefix in the binary representations of i and j . It can be verified that x is also the longest common prefix in the binary representations of $i - 1$ and $j - 1$. Further, both nodes $C.(i - 1)$ and $C.(j - 1)$ belong to S . It follows from Lemma 9 that all channels secured using the instance KACBG_x , which includes the channel between the nodes $C.i$ and $C.j$, are compromised due to collusion by nodes in S . \square

We now analyze the collusion resistance of KACG when at most $\frac{n}{4}$ nodes collude. For a given r and a subset of nodes S of C with $|S| = r$, let $\sigma.(r, S)$ denote the fraction of channels (between non-colluding nodes) that are still secure when nodes in S collude. Observe that:

$$\rho.r = \min_S \sigma.(r, S) \quad (1)$$

We define another function α on r and S that acts as a *lower bound* on the function σ and therefore can be indirectly used to compute a lower bound on ρ . Let $\psi.r$ denote the *smallest power of two greater than or equal to* r . Consider the stage $u_0 = \log(\psi.r)$. (Note that $\log(\psi.r) = \lceil \log r \rceil$.) As discussed before, at stage u_0 , the set of nodes are partitioned into $\psi.r$ groups. There is one group for each prefix of length u_0 , and the group for prefix x is denoted by $C_{\langle x \rangle}$. Further, all nodes in the group $C_{\langle x \rangle}$ have the prefix x in the binary representation of their identifier. Let Π denote the set of prefixes at stage u_0 . Also, let s denote the size of each group in the partition. Clearly, s is given by:

$$s = \frac{n}{\psi.r} \quad (2)$$

We categorize the set of channels into two types: *inter-group channels* and *intra-group channels*. An inter-group channel involves nodes from two *different* groups. An intra-group channel involves nodes from the *same* group. Observe that all inter-group channels are secured *before* stage u_0 . On the other hand, all intra-group channels are secured at stage u_0 or later.

To compute a lower bound on the function σ , we assume (conservatively) that colluding nodes can eavesdrop on all inter-group channels (that is, all inter-group channels have been compromised due to collusion). We compute a lower bound on the fraction of intra-group channels that are still secure in spite of collusion. Note that each group contains $\frac{s(s-1)}{2}$ channels.

To estimate the minimum fraction of intra-group channels that are still secure in spite of collusion, we observe that

intra-group channels belonging to different groups are secured by *different* instances of KACBG. Keys used by different instances of KACBG are independent of each other. As a result, the fraction of intra-group channels of a group compromised due to collusion of nodes in S depends *only* on the subset of nodes of S that belong to that group. As an example, suppose $n = 8$ (see Figure 1). When $r = 2$, $\psi.r = 2$ and $u_0 = 1$. There are two groups at stage 1, namely $C_{\langle 0 \rangle}$ and $C_{\langle 1 \rangle}$. Assume that nodes $C.1$ and $C.3$ collude, that is, $S = \{C.1, C.3\}$. Clearly, the fraction of intra-group channels of the group $C_{\langle 0 \rangle}$ that are compromised due to collusion only depends on nodes in the subset $C_{\langle 0 \rangle} \cap S$, which is given by $\{C.1\}$. Likewise, the fraction of intra-group channels of the group $C_{\langle 1 \rangle}$ that are compromised due to collusion only depends on nodes in the subset $C_{\langle 1 \rangle} \cap S$, which is given by $\{C.3\}$.

We define a function v that maps the set S to a vector, denoted by $v.S$. Intuitively, an entry in the vector $v.S$ captures the number of nodes from S that belong to a group at stage u_0 . Let $v.S[x]$ denote the entry in the vector $v.S$ for the group $C_{\langle x \rangle}$. By definition, the entry $v.S[x]$ is given by:

$$v.S[x] \triangleq |C_{\langle x \rangle} \cap S|$$

Clearly, we have:

$$\sum_{x \in \Pi} v.S[x] = r \quad (3)$$

We define a function f that acts as a lower bound on the fraction of intra-group channels of a group that are still secure in spite of collusion. It is given by:

$$f.c \triangleq \begin{cases} 1 - \frac{c}{2} & : c \leq 2 \\ 0 & : c > 2 \end{cases}$$

Lemma 12 *The minimum fraction of intra-group channels of a group $C_{\langle x \rangle}$ that are still secure in spite of collusion is given by $f.(v.S[x])$.*

Proof: If $v.S[x] = 0$, then $C_{\langle x \rangle} \cap S = \emptyset$ implying that all intra-group channels of $C_{\langle x \rangle}$ are secure. If $v.S[x] = 1$, then let p be the node in $C_{\langle x \rangle} \cap S$. Clearly, only intra-group channels of $C_{\langle x \rangle}$ that are compromised are those incident on the node p . Therefore the fraction of intra-group channels compromised is given by $\frac{2(s-1)}{s(s-1)} = \frac{2}{s}$. If $r \leq \frac{n}{4}$, then $\psi.r \leq \frac{n}{4}$ because n is a power of two. This implies that $s \geq 4$, which, in turn, means that at least one-half of intra-group channels of $C_{\langle x \rangle}$ are still secure. Finally, if $v.S[x] \geq 2$, then all intra-group channels of $C_{\langle x \rangle}$ may be compromised in the worst case. \square

Finally, the function α is defined as:

$$\alpha.(r, S) \triangleq \frac{s(s-1)}{n(n-1)} \sum_{x \in \Pi} f.(v.S[x])$$

We show that the function σ is lower-bounded by the function α .

Lemma 13 *The function α is a lower-bound on the function σ . Formally, for all r and S :*

$$\sigma.(r, S) \geq \alpha.(r, S)$$

Proof: Since each group has $\frac{s(s-1)}{2}$ intra-group channels, it follows from Lemma 12 that the minimum number of intra-group channels that are still secure in spite of collusion is given by:

$$\sum_{x \in \Pi} \left(f.(v.S[x]) \cdot \frac{s(s-1)}{2} \right)$$

Therefore, we have:

$$\begin{aligned} \sigma.(r, S) &\geq \frac{2}{n(n-1)} \cdot \sum_{x \in \Pi} \left(f.(v.S[x]) \cdot \frac{s(s-1)}{2} \right) \\ &= \frac{s(s-1)}{n(n-1)} \sum_{x \in \Pi} f.(v.S[x]) \\ &= \alpha.(r, S) \end{aligned}$$

This establishes the lemma. \square

Let S_{\min} denote a set of nodes with $|S_{\min}| = r$ that minimizes the value of the function α for a given value of r .

Lemma 14 *Each entry in $v.S_{\min}$ has value at most 2.*

Proof: Assume, on the contrary, that some entry in $v.S_{\min}$, say x , has value greater than 2. Since $v.S_{\min}$ contains $\psi.r$ entries, from (3), it follows that some entry in $v.S_{\min}$, say y , has value 0. Consider a node p in $C_{\langle x \rangle} \cap S_{\min}$ and a node q in $C_{\langle y \rangle}$. Let $S = (S_{\min} \cup \{q\}) \setminus \{p\}$. It can be verified that:

$$f.(v.S_{\min}[x]) = 0 \quad \text{and} \quad f.(v.S[x]) = 0$$

Further,

$$f.(v.S_{\min}[y]) = 1 \quad \text{and} \quad f.(v.S[y]) = \frac{1}{2}$$

Clearly, $\alpha.(r, S) < \alpha.(r, S_{\min})$, which contradicts the definition of S_{\min} . \square

We now compute the value of the function α for $S = S_{\min}$.

Lemma 15 *We have,*

$$\alpha.(r, S_{\min}) = \left(1 - \frac{r}{2\psi.r}\right) \left(\frac{s-1}{n-1}\right)$$

Proof: From Lemma 14, it follows that:

$$f.(v.S_{\min}[x]) = 1 - \frac{v.S_{\min}[x]}{2} \quad (4)$$

Using definition of α , we have:

$$\begin{aligned} \alpha.(r, S_{\min}) &= \frac{s(s-1)}{n(n-1)} \cdot \sum_{x \in \Pi} f.(v.S_{\min}[x]) \\ &\quad \{\text{using (4)}\} \\ &= \frac{s(s-1)}{n(n-1)} \cdot \sum_{x \in \Pi} \left(1 - \frac{v.S_{\min}[x]}{2}\right) \\ &\quad \{|\Pi| = \psi.r \text{ and using (3)}\} \\ &= \frac{s(s-1)}{n(n-1)} \cdot \left(\psi.r - \frac{r}{2}\right) \\ &\quad \{\text{simplifying}\} \\ &= \frac{s(s-1)}{n(n-1)} \cdot (\psi.r) \cdot \left(1 - \frac{r}{2\psi.r}\right) \\ &\quad \{\text{using (2)}\} \\ &= \left(1 - \frac{r}{2\psi.r}\right) \left(\frac{s-1}{n-1}\right) \end{aligned}$$

This establishes the lemma. \square

It follows that:

Theorem 16 *When $2 \leq r \leq \frac{n}{4}$,*

$$\rho.r \geq \frac{1}{\psi.r} \left(1 - \frac{r}{2\psi.r}\right) \left(1 - \frac{\psi.r-1}{n-1}\right)$$

where $\psi.r$ denotes the smallest power of two larger than or equal to r .

Proof: We have:

$$\begin{aligned} \rho.r &= \min_S \sigma.(r, S) \\ &\quad \{\text{using Lemma 13}\} \\ &\geq \min_S \alpha.(r, S) \\ &\quad \{\text{definition of } S_{\min}\} \\ &= \alpha.(r, S_{\min}) \\ &\quad \{\text{using Lemma 15}\} \\ &\geq \left(1 - \frac{r}{2\psi.r}\right) \left(\frac{s-1}{n-1}\right) \\ &\quad \{\text{using (2)}\} \\ &= \frac{1}{\psi.r} \left(1 - \frac{r}{2\psi.r}\right) \left(\frac{n-\psi.r}{n-1}\right) \\ &\quad \{\text{simplifying}\} \\ &= \frac{1}{\psi.r} \left(1 - \frac{r}{2\psi.r}\right) \left(1 - \frac{\psi.r-1}{n-1}\right) \end{aligned}$$

This establishes the theorem. \square

Note that lower bound on collusion resistance strongly depends on the definition of function f . Our definition of function f provides a somewhat conservative lower bound on the fraction of channels that are still secure in spite of collusion. For example, we assume that all intra-group channels are compromised even if the group contains only two nodes from the colluding set of nodes. We believe that the collusion resistance of our scheme is actually better than what is indicated by Theorem 16.

4. Conclusion and Future Work

In this paper, we have presented a novel key assignment scheme that ensures that any two nodes can communicate in a secure manner without the possibility of eavesdropping by another node. Our scheme has better space- and time-complexity than the existing scheme proposed by Aiyer *et al.* [1]. Further, our scheme is able to tolerate collusion between nodes to a much better extent than their scheme.

Although our key assignment scheme for a complete graph is the most space-efficient scheme that has been proposed so far to our knowledge, it is not optimal since Aiyer *et al.* proved that there exists a secure key assignment scheme that assigns only $O(\log n)$ keys to each node [1]. The proof is however non-constructive in nature. Optimal key assignments are known for certain classes of communication graphs (*e.g.*, star graphs, acyclic graphs, planar graphs, and complete bipartite graphs). We plan to investigate more efficient key assignment schemes for complete graphs and other classes of communication graphs in the future.

References

- [1] A. S. Aiyer, L. Alvisi, and M. G. Gouda. Key Grids: A Protocol Family for Assigning Symmetric Keys. In *Proceedings of the 14th International Conference on Network Protocols*, pages 178–186, Santa Barbara, California, USA, Nov. 2006.
- [2] L. Eschenauer and V. Gligor. A Key Management Scheme for Distributed Sensor Networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47, Nov. 2002.
- [3] L. Gong and D. J. Wheeler. A Matrix Key-Distribution Scheme. *Journal of Cryptology: The Journal of the International Association for Cryptologic Research*, 2(1):51–59, 1990.
- [4] M. G. Gouda, S. S. Kulkarni, and E. S. Elmallah. Logarithmic Keying of Communication Networks. In *Proceedings of the 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 314–323, Dallas, Texas, USA, Nov. 2006.
- [5] J. Hubaux, L. Buttyán, and S. Capkun. The Quest for Security in Mobile Ad-Hoc Networks. In *Proceedings of the 2nd*

ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pages 4–5, Long Beach, California, USA, Oct. 2001.

- [6] J. Kong, P. Zefros, H. Luo, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks. In *Proceedings of the 9th International Conference on Network Protocols*, pages 251–260, Riverside, California, USA, Nov. 2001.
- [7] S. S. Kulkarni, M. G. Gouda, and A. Arora. Secret Instantiation in Ad-Hoc Networks. *Journal of Computer Communications*, 29(2):200–215, Jan. 2006.
- [8] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 189–199, July 2001.
- [9] M. Tatebayashi, N. Matsuzaki, and D. B. Newman. Key Distribution Protocol for Digital Mobile Communication Systems. In *Advances in Cryptology—Proceedings of the Annual International Cryptology Conference (CRYPTO)*, pages 324–334, 1989.
- [10] V. Varadharajan and Y. Mu. Design of Secure End-to-End Protocols for Mobile Systems. In *Proceedings of the IFIP World Conference on Mobile Communications*, pages 258–266, Canberra, Australia, Sept. 1996.