

An Optimal Symmetric Secret Distribution of Star Networks¹

Bruhadeshwar Bezawada
Department of Computer Science
International Institute of
Information Technology
Hyderabad, India 500032

Sandeep S. Kulkarni
Department of Computer Science
and Engineering
Michigan State University
East Lansing MI 48824, USA

Abstract

In this paper, we focus on the problem of key distribution that permits local broadcast authentication. We focus on four types of graphs (1) star graphs, (2) bipartite graphs, (3) planer graphs and (4) fully connected graphs. For star graphs, we show that as n , the number of *satellite* nodes in the star network, tends to ∞ , it suffices to maintain $\log n + 1/2 \log \log n + 1$ secrets at the center node. However, $\log n + 1/2 \log \log n$ secrets do not. Even in the absence of the constraint of $n \rightarrow \infty$, we argue that these bounds are reasonably tight, i.e., there are several examples for finite values of n where $\lceil \log n + 1/2 \log \log n \rceil$ secrets do not suffice although $\lceil \log n + 1/2 \log \log n + 1 \rceil$ secrets suffice for virtually all cases of practical interest. We also show that our protocol can reduce the number of secrets in planar and fully connected bipartite graphs and fully connected graphs.

¹Email: bezawada@iiit.ac.in, sandeep@cse.msu.edu

Web: <http://www.cse.msu.edu/~sandeep>.

This work was partially sponsored by NSF CAREER CCR-0092724, DARPA Grant OSURS01-C-1901, ONR Grant N00014-01-1-0744, NSF grant EIA-0130724, and a grant from Michigan State University.

1 Introduction

Consider a communication network represented by a graph $G = (V, E)$ where V is the set of users/nodes and $E \subseteq V \times V$ is a symmetric relation. We use the set of edges to denote the need for authenticity between the communication between the nodes connected by the edge. In other words, $(v_1, v_2) \in E$ implies that v_1 needs to verify authenticity of messages received from v_2 and vice versa. (Note that the set of edges in this graph are possibly application dependent and may not be the same as those corresponding to physical communication links between nodes. Hence, other nodes in the network may be required for *routing* although the intermediate nodes would not be able decrypt/encrypt any communication they forward.)

As an illustration of such a network, consider a sensor network consisting of a base station and a set of sensors. In such a network, a sensor may *need* to communicate securely with a base station. This communication may itself be assisted by other sensors in terms of routing. However, the intermediate sensors can neither learn the contents of the message nor can generate messages on behalf of the sender. Another example may be a set of clients and servers where each client needs to talk to (a subset of) the servers. However, the clients (respectively, servers) do not need authentic communication among themselves.

Now, consider the set $NBR = \{v_2 | (v_1, v_2) \in E\}$. This is the set of nodes that v_1 communicates with. We consider two types of communication: unicast where v_1 communicates with one of its neighbors and the neighbor wants to verify that the message is indeed from v_1 , and (local) broadcast, where v_1 sends a message to all nodes in NBR and each node wants to verify that the message is indeed from v_1 . We denote this problem as authentication with broadcast capability.

One simple way to solve the authentication problem with broadcast capability is to assign a public/private pair to each node. Each node stores its public/private key and the public keys of the nodes it communicates with. A message encrypted with the private key of the sender will provide the required authentication. The problem with this approach, however, is that the cost of encryption/decryption with public/private keys is very high. Also, each node may need to store a large number of public keys (unless certificates are used thereby increasing the cost of encryption/decryption even further).

Based on this motivation, we focus on solving the authentication problem with broadcast capability using symmetric keys. Our goal is to identify key distribution algorithms that minimize the cost of the unicast/broadcast communication. We consider four types of communication graphs: a star graph, a planer graph, a bipartite graph, and a fully connected graph. Of these the star graph is motivated by examples such as sensor networks discussed above. A planer graph is motivated by the case where nodes are distributed in a plane and each node communicates with its (physical) neighbors. A bipartite graph is useful in cases of client-server applications mentioned above. And, finally, fully connected graphs capture the case where each node may be interested in authentic communication from each other node in the system.

We proceed as follows: We first precisely define the problem of key distribution with broadcast capability using the example of communication in a star network. In particular, we identify constraints on the broadcast communication from the center node to satellite nodes. Essentially, these constraints require that the *cost* of the broadcast communication should be proportional to the secrets that the center node maintains. Then, we present our approach for secret distribu-

tion for star network and show its optimality. Subsequently, we extend this approach for planer graphs, bipartite graphs and fully connected graphs.

The main contributions of this paper are as follows:

- We show that as the number of satellite nodes in the star network, say n , tends to ∞
 - There exists a protocol that maintains $\log n + 1/2 \log \log n + 1$ secrets at the center node.
 - Under the assumption that the number of secrets maintained by the satellite nodes is identical, we show that there does not exist a protocol that maintains $\log n + 1/2 \log \log n$ secrets at the center node.
- In the absence of the constraint of $n \rightarrow \infty$, we show that
 - There exists a protocol (shown constructively) that maintains $\lceil \log n + 1/2 \log \log n + 1 \rceil$ secrets at the center node for virtually all cases of practical interest.
 - Under the assumption that the number of secrets maintained by the satellite nodes is identical, there does not exist a protocol that maintains $\lceil \log n + 1/2 \log \log n \rceil$ secrets at the center node for many cases of practical interest.
- We show that our protocol can also be extended to acyclic, fully connected bipartite networks and fully connected networks. Compared to existing protocols [1], these protocols reduce the number of secrets that each node needs to maintain.

- We show that our protocol provides tradeoff between internal attackers and external attackers.

Organization of the paper. The rest of the paper is organized as follows: In section 2, we define the problem of key distribution with broadcast capability. In Section 3, we provide our solution for star graph and show its optimality. We extend it to planer and fully bipartite graphs in Section 4 and to fully connected graphs in Section 5. Finally, we discuss the related work in Section 6 and conclude in Section 7.

2 Problem Statement

In this section, we precisely define the problem of authentication with broadcast capability. For simplicity, initially, we focus on star networks where there is one center node and other satellite nodes that only communicate with the center node. This problem statement first focuses on defining the *number of stored secrets* and then relates it to the cost of broadcast authentication. To motivate the need for such a definition, consider the following solution for key distribution in a star network. In this solution, the center node maintains a secret x . Each satellite node has an ID ranging from $1 \dots n$. The secret associated with satellite node j is $f(x, j)$. Clearly, $f(x, j)$ could be used for unicast communication to/from j and the center node. (For example, $f(x, j)$ could be used to generate a message digest that the receiver can verify. Since $f(x, j)$ is known only to the center node and satellite node j , only they can generate the corresponding message digest.)

In this case, the number of secrets that each user *stores* is 1. While the center node has access to many secrets, that it can generate at runtime, they need not be stored. Thus, we need to distinguish between the number

of *stored* secrets and the number of *generated* secrets that are potentially unbounded. We use $stored(v_1)$ to denote secrets stored at v_1 and $generated(v_1)$ to denote secrets that v_1 could generate from them.

While the above solution works for unicast, it is very inefficient for broadcast. In particular, for broadcast, the center node must provide authentication code using $f(x, j)$, where, $1 \leq j \leq n$. Each satellite node can verify the authentication code by generating the code locally, using its secret $f(x, j)$, and comparing the generated code with the code sent by the center node. Clearly, in the above solution, the cost of generating authentication codes, which is $O(n)$, for broadcast is very high.

Now, given are two users, j and k where j has secret s and k has secret $f(s)$ where f is some function. Then, a message encrypted with s cannot be decrypted by k (unless strong requirements are added for function f and the algorithm for encryption). Since we would like to allow the use of any symmetric key based encryption approaches, for j and k to communicate securely, they must share common secret(s). Hence, for broadcast communication, where the sender signs the message separately with multiple secrets, the receiver must have one or more of the secrets that are actually used for generating the signature. Hence, to provide broadcast capability, we impose the following constraint on the problem of secret distribution in star network: We require that user maintain a set of secrets. Whenever the user sends a (local) broadcast message to its neighbors, it provides authentication codes using (a subset of) of the secrets it has. Upon receiving the message, the receiver will verify the signatures it can verify. (Note that the receiver cannot verify all signatures since it does not have all the secrets used by the sender.) It is required that when the receiver verifies the signatures it can, it must be the case that the message

is indeed authentic. Thus, we define the secret distribution with broadcast capability as follows:

Problem Statement 2.1 (Key Distribution With Broadcast Capability).

Given is an undirected graph $G(V, E)$ where V is the set of users and E is the set of edges such that given two users v_1 and v_2 in V , $(v_1, v_2) \in E$ iff v_1 needs the communication with v_2 to be authentic. Then, assign to each node v_1 a set of secrets, denoted by $stored(v_1)$, such that for any $(v_1, v_2) \in E$:

1. (Existence of Common Secrets usable for broadcast): $stored(v_1) \cap generated(v_2) \neq \phi$, where $generated(v_2)$ is the set of secrets that can be generated from $stored(v_2)$ (subject to constraints specified later)
2. (Authenticity): For any $v_3 \in V$: $v_3 \neq v_1 \wedge v_3 \neq v_2$: $stored(v_1) \cap generated(v_2) \not\subseteq generated(v_3)$.

Note that the problem statement requires broadcast capability, i.e., if a node, say v_1 , sends a message and includes a signature block that contains a signature from each of the secret it stores then any node that receives this message can verify the signatures by using the secrets in $stored(v_1) \cap generated(v_2)$. For the sake of comparison, we also present the problem statement (although it is not solved in this paper) that does not guarantee authentication capability for broadcast.

Problem Statement 2.2: (Key Distribution Without Broadcast Capability).

Given is an undirected graph $G(V, E)$ where V is the set of users and E is the set of edges such that given two users v_1 and v_2 in V , $(v_1, v_2) \in E$ iff v_1 needs the communication with v_2 to be authentic. Then, assign to each node v_1 a set of secrets, denoted by $stored(v_1)$, such that for any $(v_1, v_2) \in E$:

1. (Existence of Common Secrets): $generated(v_1) \cap generated(v_2) \neq \phi$.
2. (Authenticity): For any $v_3 \in V$: $v_3 \neq v_1 \wedge v_3 \neq v_2$: $generated(v_1) \cap generated(v_2) \not\subseteq generated(v_3)$.

Assumption about *generated* secret set. The above problem definition is independent of how $generated(v_1)$ is defined. Hence, constraints –in the form of adversary capability– must be added to make the problem solvable. We make standard assumption about adversary capability, i.e., the adversary can combine the secrets it has e.g., by XOR-ing them/adding them, etc. However, such combination does not allow the adversary to *guess* the secrets that other users have. Thus, $generated(v_1)$ is obtained by using $stored(v_1)$ and applying any approaches that adversary can use to combine those secrets.

3 Optimal Solution for Authenticated Communication in Star Network

In this section, we focus on Problem 2.1 for the case where the network is a star network i.e., it consists of a center node a set of satellite nodes that communicate with it. Let the set of secrets at the center node be K and $|K| = k$. Each satellite node receives a unique subset of size l , $l > 0$, from this set. Note that, by construction, given any two distinct satellite nodes v_2, v_3 , we have $K \cap generated(v_2) \not\subseteq generated(v_3)$. Thus, the constraints of Problem 2.1 are satisfied. We term this protocol instance as $p(k, l)$ ².

²Technically, $p(k, l)$ is a family of protocols. However, for brevity of presentation, we denote $p(k, l)$ as a protocol. However, we note that all the results in this paper attributed to *protocol* $p(k, l)$ are valid for any member of the $p(k, l)$ protocol family

Using $p(k, l)$, authentication can be achieved in the communication as follows:

- To authenticate a message m broadcasted by the center node to the satellite node, the center node generates authentication codes with each of the k secrets. Each authentication code consists of the message digest md of the message computed using a secret held by the center. The center appends the k authentication codes thus generated to the message and broadcasts the resulting message. Now, when a satellite node receives this message, it uses its subset of l secrets to compute l authentication codes. The satellite node then verifies these authentication codes with the corresponding authentication codes sent by the satellite node. Note that, each satellite node can verify only those authentication codes for which it has the corresponding generating secret.
- To authenticate a unicast message m from the center to a particular satellite node, the center node first computes an *XOR* of the subset of the keys that this satellite node knows. Now, the center node uses the combined secret to compute the message authentication code for this message. The center node appends this authentication code to the message and unicasts the message to the satellite node.
- To authenticate a unicast message m from a satellite node to the center, the satellite node uses the same approach used by the center node for authenticating unicast messages.

In the remaining section, we consider the case where the number of satellite nodes, n , tends to ∞ : First, we show (cf. Theorem 5) that $\log n + c \log \log n$ secrets do

not suffice if $c < 1/2$. Using the proof of this results, we show (cf. Theorem 6) that $\log n + 1/2 \log \log n + 1$ secrets suffice to handle n nodes whereas $\log n + 1/2 \log \log n$ do not.

Theorem 1. $p(k, l)$ provides authentication for the communication in the star network.

Proof. The proof follows from the construction of $p(k, l)$. \square

Theorem 2. Protocol $p(k, l)$ can accommodate upto $C(k, l)$ satellite nodes.

Proof. This follows directly, as this $C(k, l)$ is the number of unique subsets of keys that can be generated. \square

Corollary 3. For a given value of k , choosing $l = k/2$ maximizes the number of satellite nodes that can be accommodated. \square

Theorem 5. As $n \rightarrow \infty$, protocol $p(k, l)$ where $k = (\log n + c \log \log n)$, where $c < 1/2$ and $l = k/2$ cannot provide authentication for the star network with n nodes.

Proof. As before we evaluate $C(k, l)$ using Stirling's approximation and show that this value is less than n . Based on Theorems 1 and 2, this result proves this theorem. From the above proof,

$$C(k, k/2) \approx \frac{2^{k*2}}{\sqrt{2\pi k}}$$

{Now, letting $k = \log n + c \log \log n$ }

$$\begin{aligned} &= \frac{2^{(\log n + c \log \log n)*2}}{\sqrt{2\pi(\log n + c \log \log n)}} \\ &= n \frac{2^{*2^c \log \log n}}{\sqrt{2\pi(\log n + c \log \log n)}} \\ &= n \frac{2^{*(2^{\log \log n})^c}}{\sqrt{2\pi(\log n + c \log \log n)}} \\ &= n \frac{2^{*(\log n)^c}}{\sqrt{2\pi(\log n + c \log \log n)}} \end{aligned}$$

Note that, as $n \rightarrow \infty$, the multiple of n in the above formula tends to 0. Hence, as $n \rightarrow \infty$, $C(k, k/2) < n$. \square

Based on the above theorem, if $c = 1/2$ then the multiple of n is $\frac{2}{\sqrt{2\pi}}$. Since this number of less than 1, $\log n + c \log \log n$ secrets do not suffice when $c = 1/2$. However, if the number of secrets is $(\log n + 1/2 \log \log n + 1)$, then in the above formula, the multiple of n is $\frac{4}{\sqrt{2\pi}}$. Since this number is greater than 1, $(\log n + 1/2 \log \log n + 1)$ secrets suffice for key distribution in a star network as $n \rightarrow \infty$. Thus, we have

Theorem 6. As $n \rightarrow \infty$, protocol $p(k, l)$ where $k = (\log n + 1/2 \log \log n)$, and $l = k/2$ cannot provide authentication for the star network with n nodes. And, as $n \rightarrow \infty$, protocol $p(k, l)$ where $k = (\log n + 1/2 \log \log n + 1)$, and $l = k/2$ can provide authentication for the star network with n nodes. \square

When compared to the secret distribution for star network in [1] our secret distribution requires lesser number of secrets. In [1], the center maintains $2 \log n$ secrets and in our secret distribution protocol the center only needs to maintain $\log n + O(\log \log n)$ secrets. The number of secrets stored by the satellite nodes is reduced as well. In [1], each satellite node maintain $\log n$ secrets and in our secret distribution protocol, each satellite node maintains atmost $(\log n)/2 + O(\log \log n)$ secrets.

Also, since the above theorem applies for the case where $n \rightarrow \infty$, the natural question is about what happens for small values of n . Here, we note that we checked whether $\lceil \log n + 1/2 \log \log n \rceil$ secrets suffice for $n < 50000$. For $2 \leq n \leq 1000$, we found this number to be insufficient for 510 values and for $1000 \leq n \leq 50000$, we found that this number to be insufficient for 20313 values. By contrast, if we consider $\lceil \log n + 1/2 \log \log n + 1 \rceil$ secrets then this number suffices for the case where the number of nodes is less than 50000. As an illustration, we refer the reader to Figure 1.

We would like to note that while the reduction in the number of secrets is from $2 \log n$ to $\log n + O(\log \log n)$, this reduction is especially valuable when we consider the number of nodes that can be supported with a given set of secrets at the center. For example, Figure 2 compares the number of nodes that can be supported with our scheme with that in [1]. If 10 secrets are available at the center then the scheme in [1] can support upto 32 satellite nodes whereas our protocol can tolerate upto 252 satellite nodes. Or, if 20 secrets are available at the center then the scheme in [1] can support upto 1024 satellite nodes whereas our protocol can tolerate upto 184756 satellite nodes.

3.1 Optimality of Secret Distribution in Star Networks

Consider the problem of secret distribution in star network where the number of secrets stored by the satellite nodes is equal. Thus, based on the constraints in Section 2, a solution is of the form where, for some values of k and l , the center node maintains k secrets and each satellite node maintains a subset of size l from that set. In other words, a solution is of the form $p(k, l)$ for some value of k and l . Now, based on Theorem 6, as $n \rightarrow \infty$, $\log n + 1/2 \log \log n + 1$ is the minimum number of secrets that need to be maintained by the center node.

4 Extensions to Acyclic, Planar and Complete Bipartite Networks

In Section 3, we described our secret distribution protocol, $p(k, l)$, for a star network. In this section, we extend it for the cases where

the communication graph is acyclic, planar, or complete bipartite.

Acyclic Networks. Since an acyclic undirected graph consists of a set of trees, we describe the secret distribution for a tree. The same algorithm can be applied for each tree separately to obtain the secret distribution for acyclic graphs. Given a tree, we choose one of the nodes in it as a root and consider the corresponding rooted version. In particular, this allows us to define a parent of node in the tree and to define whether a node is a leaf. Now, consider a non-leaf node and its children in this network. This sub-network is a star graph where the non-leaf node is the center node and the children are satellite nodes. Now, we apply the $p(k, l)$ secret distribution with appropriate values of k and l for this star graph. We repeat this process for each star graph obtained by considering a non-leaf node and its children. The secrets assigned to a node in the tree are the same as the union of the secrets assigned to it in any star graph considered in this fashion.

Now, in the star graph, the non-leaf node which is the center node gets $\log d + O(\log \log d)$ secrets and each satellite node gets $1/2 \log d + O(\log \log d)$ secrets, where d is the degree of the center node. Furthermore, for any given non-leaf node, it is a center node in at most one star graph considered above. Likewise, it is a satellite node in at most one star graph (where its parent is the center node) considered above. Thus, the number of secrets at any node is at most $3/2 \log d + O(\log \log d)$, which is less than that in [1] where $2 \log d$ secrets are maintained.

Planar Networks. The extension to planar network is similar to that in [1]. In particular, in [1], a well-known result from graph theory (e.g., [2]) is used for extension to planar graphs. The result in [2] states that any planar graph G can be decomposed into at most three acyclic graphs, called *factors*. Each factor has the same nodes as the origi-

nal graph G and the degree of each factor is at most the degree of the original graph. Now, secrets can be independently distributed to each factor to obtain the secret distribution of planar network. By using our scheme for acyclic networks, it would be possible to reduce the number of secrets used in planar networks as well.

Fully Connected Bipartite Graphs.

A complete bipartite graph is a graph $G(V, E)$ such that its vertex set can be partitioned into two disjoint sets, V_1 and V_2 , the edge set, E is induced by these two vertex sets such that every vertex in V_1 is connected to every other vertex in V_2 . No edge exists between any of the vertices in V_1 (respectively, V_2). Examples of such communication graph is one where the vertex set V_1 contains servers and the vertex set V_2 contains clients. The communication is from servers to clients and vice-versa. No server (respectively, client) needs to communicate with each other.

For this network, we employ the following secret distribution technique. We treat the vertex set V_1 as a single center node, say C , and the vertex set V_2 as satellite nodes. As this represents a star network with C as star node, we instantiate a $p(k, l)$ secret distribution for this network. Thus, in this distribution, the node C needs to store $\log |V_2| + O(\log \log |V_2|)$ secrets. Since C represents the vertex set V_1 , each of the nodes in V_1 are given all these secrets. We repeat this procedure by treating the vertex set V_2 as the center node and the vertex set V_1 as the satellite nodes. This secret distribution would provide an additional $1/2 \log |V_1| + O(\log \log |V_1|)$ secrets to each node in V_1 . Thus, the number of secrets given to nodes in V_1 is $3/2 \log d + O(\log \log d)$, where $d = \max(|V_1|, |V_2|)$ is the maximum degree of any node in the communication graph.

5 Extension for Fully Connected Communication Graph

In this section, we describe how the protocol $p(k, l)$ can be extended to solve Problem 2.1 in all-to-all communication, i.e., where communication graph is fully connected. Our key distribution protocol is designed in stages; the first stage extends the protocol $p(k, l)$ by considering a fully connected graph as a set of star graphs. This stage is intended for the case where the number of nodes in the graph is small. Then, stage 2 uses the scheme in stage 1 in an hierarchical manner.

5.1 Stage 1

In this stage, we design a protocol that establishes secrets for a group of d users. We first select a unique node, say X , and consider the star graph where X is the center node and the remaining $d-1$ nodes are satellite nodes. The center node maintains a set of $k = f(d)$ secrets, where $f(d) = \lceil \log d + 1/2 \log \log d + 1 \rceil$. Each satellite node receives a unique subset of size $l = f(d)/2$ from this set. Note that this is an instance of $p(k, l)$ where $k = f(d)$ and $l = f(d)/2$. Now, we repeat the above protocol by considering each node as a center node. Thus, for a fully connected network of d users, each user is a center node in one star graph and a satellite node in the $d-1$ star graphs. As a center node, each node stores $f(d)$ secrets and as a satellite node it stores $d-1$ secrets, one for each of the $d-1$ center nodes. Thus, we have

Theorem 5.1. The number of secrets stored by each user in Stage 1 is: $f(d) + \frac{d-1}{2} f(d) = \frac{d+1}{2} f(d)$. \square

Theorem 5.2. The key distribution in the above solution solves the problem 2.1. \square

5.2 Stage 2

Now, we consider d communication networks from Stage 1, which gives us a network of d^2 users, and denote this grouping by G_p . We treat each basic structure of d users as a single (virtual) node U_i where $1 \leq i \leq d$ i.e., in any secret distribution scheme, any secret that is given to the virtual node U_i is actually given to all the users that are part of U_i .

With this setting, we consider the star graph where U_i is the center node for some $1 \leq i \leq d$ and all users in $G_p - U_i$ are satellite nodes. Thus, there are $d(d-1)$ ($\approx d^2$) satellite nodes. We instantiate $p(k, l)$ for such a network. Thus, with this approach, each node in U_i will get $f(d^2)$ secrets whereas each node in $G_p - U_i$ will get $f(d^2)/2$ secrets. Now, we repeat this process by considering the star graphs where for each i , U_i is the center node. As in Stage 1, each user is a satellite node in $d-1$ star graphs and the virtual node it is in is the center node in one star graph. Thus, each user maintains $\frac{d+1}{2}f(d^2)$ secrets in Stage 2.

For an arbitrary number of users N , we repeat the protocol of Stage 2. Thus at stage k we group d (virtual) users considered in stage $k-1$. For N users, the number of such stages required will be $\lceil \log N \rceil$ to secure communication between all the users. The number of secrets stored by each user is: $\frac{d+1}{2}.y$ where, the term y is as shown:

$$y = f(d) + f(d^2) + \dots + f(d^{\lceil \log_d N \rceil}) \quad (3)$$

Theorem 5.3. The key distribution in the above solution solves the problem 2.1. \square

5.3 Storage Analysis

Now, we evaluate the storage at each user by evaluating formula (3), where $f(d) = \log d + 1/2 \log \log d + 1$.

$$\begin{aligned} y &= \log d + \log d^2 + \log d^3 + \dots + \log d^{\lceil \log_d N \rceil} + O(\log \log d) \\ &= \log d + 2 \log d + 3 \log d + \dots + (\lceil \log_d N \rceil) * \log d + O(\log d) \\ &= \frac{(\lceil \log_d N \rceil)(\lceil \log_d N \rceil + 1)}{2} (\log d) + O(\log N * \log \log N) \\ &= \frac{(\lceil \log_2 N \rceil)(\lceil \log_2 N \rceil + 1)}{2 \log d} + O(\log N * \log \log N) - (2) \end{aligned}$$

The storage at each user is given by: $\frac{(d+1)}{2}.y$ where the value of y is substituted from (2). By letting $d = 2$, the number of secrets per user is $\frac{\log^2 N}{2} + O(\log N * \log \log N)$.

6 Related Work

Secure communication among a group of nodes in a communication network using symmetric key mechanisms has been widely studied. In particular, it has been shown by [3, 4], that the number of secrets stored by each node can be less than $O(n)$ for a system of n nodes. In [3, 4], the authors describe a secret distribution protocol which allow any two users to communicate securely while storing only $O(\sqrt{n})$ secrets. These protocols, are designed for the case where the communication graph is fully connected.

In [5], the authors improve upon this storage requirement and describe a secret distribution protocol that requires each user to store $O(\log^2 n)$ secrets. In their work, the authors show the existence of $O(\log n)$ secret distribution protocol for star network like scenarios for a communication network without actually constructing the protocol. Reducing the storage to logarithmic bounds has benefits in resource constrained networks such as sensor networks. In a sensor network each node has limited memory for storing secrets. Examples of such resource constrained networks include sensor networks [6, 7], ad-hoc networks [8–10] and mobile networks [11, 12].

In [1], the authors describe a logarithmic secret distribution protocol for a star network where the storage is logarithmic for both the center, which stores $2 \log n$ secrets and the satellite nodes, which store $\log n$ secrets. They extend their results to achieve logarithmic secret distribution for several classes of communication networks including, star networks, acyclic networks, cycle-limited networks, planar networks and dense bipartite networks.

7 Conclusion

In this paper, we presented the problem of key distribution with broadcast capability for authentication. Thus, the key distribution was such that it allowed each node to provide authentication for unicast communication as well as broadcast communication where it sends a message to all nodes it intends to communicate with. Furthermore, by designing solutions for several commonly used communication topologies, our solution allows the designer to use key distribution that is ideal for the communication graph at hand. Moreover, for the case where the communication graph cannot be determined easily, one can simply utilize the solution for fully connected graphs.

We presented solutions for the cases where the communication graph is a (1) star graph, (2) acyclic graph, (3) bipartite graph and (4) fully connected graph. For the star graph, we presented a lower bound on secret distribution with broadcast capability. We showed that this lower bound is indeed achievable and tight. In particular, we showed that as $n \rightarrow \infty$, $\log n + 1/2 \log \log n + 1$ secrets at the center node suffice. However, $\log n + 1/2 \log \log n$ secrets do not. Even in the absence of the constraint of $n \rightarrow \infty$, we find that these bounds are reasonably tight, i.e., there are several examples for finite values of

n where $\lceil \log n + 1/2 \log \log n \rceil$ secrets do not suffice although $\lceil \log n + 1/2 \log \log n + 1 \rceil$ secrets suffice for virtually all cases of practical interest.

The result in this paper reduces the number of secrets to almost half compared to that in [1] where $2 \log n$ secrets are used at the center node and $\log n$ secrets are used at the satellite nodes. By contrast, our protocol uses $1/2 \log n + O(\log \log n)$ secrets at satellite nodes. Also, the result in this paper reduces the number of secrets maintained for fully connected graph by approximately half when compared to the protocol in [13].

Using this protocol, we showed that it is possible to reduce the secrets maintained in an acyclic, planar and fully bipartite communication graph. However, the optimality of the number of secrets in these protocols is still an open question. Also, in [1], authors have presented how their approach can be used in limited-cycle graphs and arbitrary bipartite graphs. However, since this extension is based on a specific numbering scheme, it cannot be directly applied with protocol $p(k, l)$. One of the future work in this area is to identify lower bounds for secrets in such graphs.

References

- [1] Gouda M. G., Kulkarni S. S., and Elmalah S. E. Logarithmic keying of communication networks. In *8th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS-06*, 2006.
- [2] Colbourn C. J. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

- [3] Li Gong and David J. Wheeler. A matrix key-distribution scheme. *Journal of Cryptology*, 2(1):51–59, 1990.
- [4] Kulkarni S. S., Gouda M. G., and Arora A. Secret instantiation in ad-hoc networks. *Computer Communications*, (29):200–215, 2006.
- [5] Aiyer A.S., Alvisi L, and Gouda M. G. Key grids: A protocol family for assigning symmetric keys. In *IEEE International Conference on Network Protocols*, 2006.
- [6] Eschenauer L and Gligor V. D. A key-management scheme for distributed sensor networks. In *9th ACM Conference on Computer and Communications Security*, pages 41–47, New York, NY, USA, 2002. ACM Press.
- [7] Perrig A., Szewczyk R., Tygar J. D., Wen V., and Culler D.E. Spins: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, 2002.
- [8] Hubaux J.P Buttyan K and Capkun S. The quest for security in mobile ad hoc networks. In *Mobihoc '01: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 146–155, New York, NY, USA, 2001. ACM Press.
- [9] Yang H., Luo H., Ye F., Lu S., and Zhang L. Security in mobile ad hoc networks: challenges and solutions. *IEEE Wireless Communications*, (11):38–47, 2004.
- [10] Bezawada Bruhadeshwar and Sandeep S. Kulkarni. User revocation in secure ad-hoc networks. In *ICDCIT*, pages 377–388, 2005.
- [11] Tatebayashi M., Matsuzaki N., and Newman D. Key distribution protocol for digital mobile communication systems. *Lecture Notes in Computer Science: Advances in Cryptology*, (435):324–333, 1989.
- [12] Mu Y. and Varadharajan V. On the design of security protocols for mobile communications. *Lecture Notes in Computer Science: Information Security and Privacy*, 1172:134–145, 1996.
- [13] Neeraj Mittal. Space-efficient keying in wireless communication networks. Technical Report Number : UTDCS-26-07, 2007.

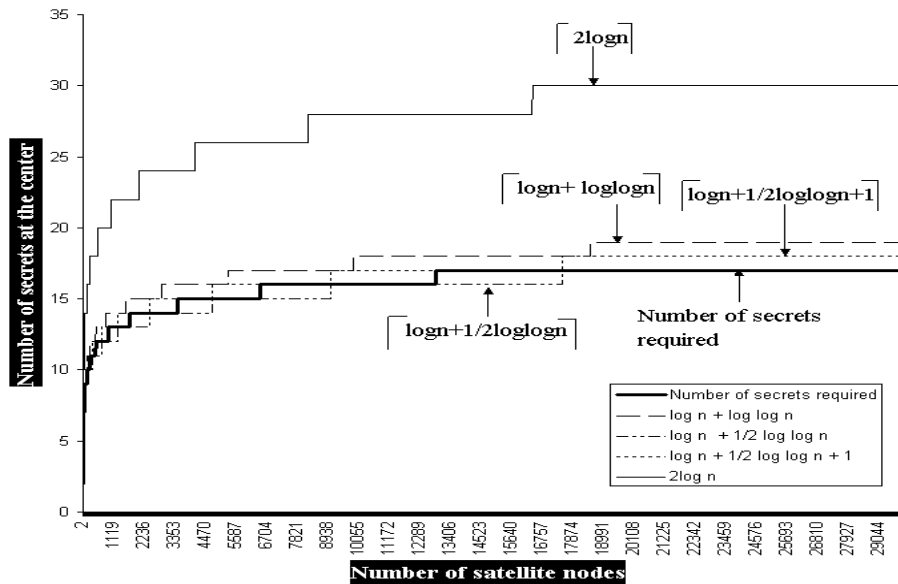


Figure 1: Number of secrets stored by the center node

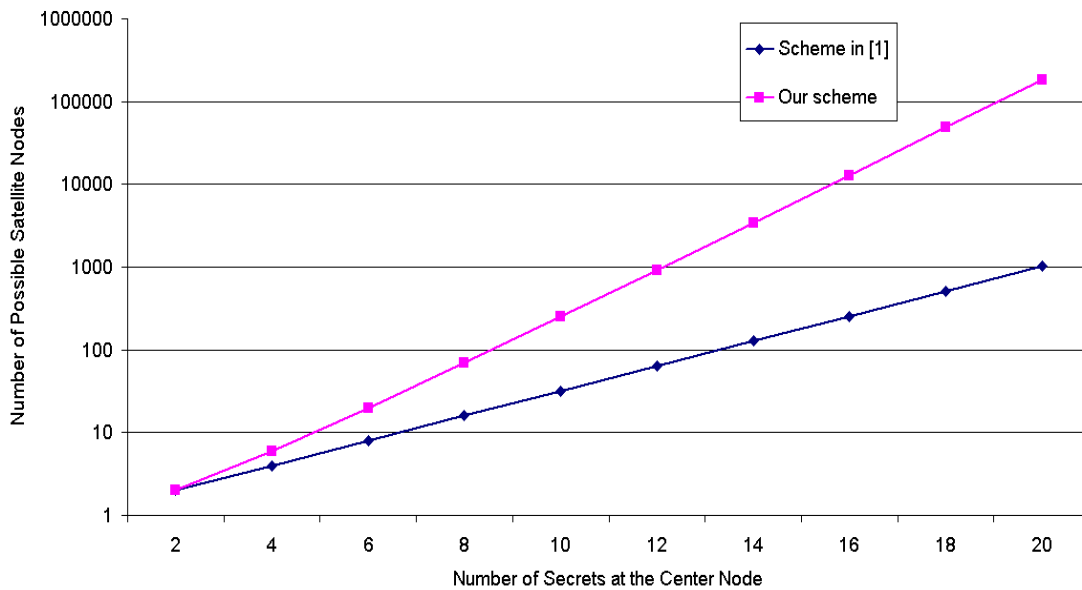


Figure 2: Number of users supported by the center node