Random Key Predistribution Schemes for Sensor Networks

Haowen Chan Adrian Perrig Dawn Song

Carnegie Mellon University

{haowenchan,perrig,dawnsong}@cmu.edu

Abstract

Efficient key distribution is the basis for providing secure communication, a necessary requirement for many emerging sensor network applications. Many applications require authentic and secret communication among neighboring sensor nodes. However, establishing keys for secure communication among neighboring sensor nodes in a sensor network is a challenging problem, due to the scale of sensor nets, the limited computation and communication resources of sensors, their deployment in hostile environments yet their lack of tamper-resistant hardware.

The limited computation resources of sensor nodes prevent using traditional key distribution mechanisms in sensor networks, such as Diffie-Hellman based approaches. Pre-distribution of secret keys among neighbors is generally not feasible, because we do not know which sensors will be neighbors after deployment. Pre-distribution of secret keys for all pairs of nodes is not viable due to the large number of sensors and the limited memory of sensor nodes.

A new key distribution approach was proposed by Eschenauer and Gligor [11] to achieve secrecy for node-to-node communication: sensor nodes receive a random subset of keys from a key pool before deployment. In the field, neighboring nodes exchange information to find one common key within their random subset and use that key as their shared secret to secure subsequent communication.

In this paper, we generalize the Eschenauer-Gligor key distribution approach. First, we propose two new mechanisms, the q-composite random key predistribution scheme and the multi-path key reinforcement scheme, which substantially increases the security of key setup such that an attacker has to compromise many more nodes to achieve a high probability to compromise communication. Second, we propose a new mechanism, random-pairwise keys scheme, to enable node-to-node authentication without involving a base station and perfect resilience against node capture. We also show how we enable distributed node revocation based on this scheme. To the best of our knowledge, no previous scheme supports efficient node-to-node authentication without involving a base station and distributed node revocation. We give detailed analysis and simulation results to each proposed scheme and show under which situations a scheme should be used to achieve the best security.

1 Introduction

Wide-spread deployment of sensor networks is on the horizon. Networks of thousands of sensors may address some of our challenging problems: real-time traffic monitoring, building safety monitoring (structural, fire, and physical security monitoring), military sensing and tracking, distributed measurement of seismic activity, real-time pollution monitoring, wildlife monitoring, wildfire tracking, etc. Many of these applications are dependent on secure operation of the sensor network, and have serious consequences if the sensor network is compromised or disrupted.

A typical sensor network has hundreds to many thousands of sensor nodes. Each sensor node is typically low-cost, limited in computation and information storage capacity, highly power constrained, and communicates over a short-range wireless network interface. Most sensor networks have a base station as a gateway to communicate with external infrastructure. Individual sensor nodes communicate locally with other sensors, and send their sensor readings to the base station. The deployment methods of sensors varies from individual installation of each sensor node, to aerial scattering from an airplane or rocket.

Achieving secure communication in sensor networks is a challenging problem for many reasons. Wireless communication is a medium that is difficult to secure against such attacks as eavesdropping and packet injection. Due to the lack of physical interfaces, attackers find it easier to claim multiple identities to the network; this attack is described by Douceur as the Sybil attack [10].

The limited computation and power resources of sensor nodes also often makes it undesirable to use public-key algorithms, such as Diffie-Hellman key agreement [9] or RSA [20]. Currently, a sensor node may require on the order of tens of seconds up to minutes to perform these operations [8, 7]. Even with possible future advances in low-power cryptographic hardware, the performance gap between asymmetric and symmetric cryptosystems is several orders of magnitude and unlikely to be closed soon. In designing security systems for sensor networks, therefore, it is highly desirable to use only fast, efficient cryptographic operations.

Communication is further impaired due to the low bandwidth of typical sensor network platforms. For example, the UC Berkeley Mica platform's transmitter has a bandwidth of 10 Kbps, and limits packet size to about 30 bytes. Often times, transmission reliability is low, making reliable message transmission of large blocks particularly expensive.

Finally, the requirement that sensors be low-cost also prevents manufacturers from making them tamper-resistant. An attacker could thus collect sensor nodes deployed in untrusted environments and compromise their cryptographic keys.

In sensor network security, an important open problem is how to bootstrap secure communications between sensor nodes. This *key-setup* problem is complicated not only by the limitations of sensors described above, but also by the fact that many deployment methods such as aerial scattering are random and thus key-setup schemes may not assume prior knowledge of the deployment context of any node. Eschenauer and Gligor recently proposed a *random key* pre-distribution scheme: each sensor node receives a random subset of keys from a key pool before deployment; to agree on a key for communication, two nodes find one common key within their subsets and use that key as their shared secret [11]. We review their approach (which we call the *basic* random key scheme) in Section 3.

In this paper, we propose new mechanisms for random key pre-distribution to enable secure key establishment among neighboring sensors. First, we propose two new mechanisms, the q-composite random key predistribution scheme and the multi-path key reinforcement scheme, which substantially increases the security of key setup such that an attacker has to compromise many more nodes to achieve a high probability to compromise communication. Second, we propose a new mechanism, random-pairwise keys scheme, to enable node-to-node authentication without involving a base station and perfect resilience against node capture. We also show how we enable distributed node revocation based on this scheme. To the best of our knowledge, no previous scheme supports efficient node-to-node authentication without involving a base station and distributed node revocation. We give detailed analysis and simulation results to each proposed scheme and show under which situations a scheme should be used to achieve the best security.

The rest of the paper is organized as follows. We first establish several security evaluation criteria for sensor network key distribution in Section 2. We then give an overview of the basic scheme by Eschenauer and Gligor in Section 3. We describe our new mechanism q-composite random key predistribution scheme in Section 4, and our multi-path key reinforcement scheme in Section 5. We explain our new mechanism, the random-pairwise keys scheme, for node authentication and distributed node revocation in Section 6. Finally, we discuss about related work in Section 7 and summarize our results in Section 8. Appendix A lists the variable naming conventions we use in this paper.

2 Problem Statement and Evaluation Metrics

We first discuss the topology and architecture of a typical sensor network. Next, we present the goals and evaluation metrics for sensor network key distribution.

2.1 Sensor Network Architecture

Generally, sensor nodes communicate over a wireless network. A typical sensor network forms around one or more *base stations*, which interface the sensor network to the outside network. The communication patterns within a sensor network fall into three categories: node to node communication (e.g., aggregation of sensor

readings), node to base station communication (e.g., sensor readings), base station to node communication (e.g., specific requests).

An example of a sensor network node's hardware configuration is the Berkeley Mica Motes. They feature a 8-bit 4 MHz Atmel ATmega 128L processor with 128K bytes program store, and 4K bytes SRAM. The processor only supports a minimal RISC-like instruction set, without support for multiplication or variable-length shifts or rotates. The ISM band radio receiver communicates at a peak rate of 40Kbps at a range of 100 feet.

2.2 The problem of bootstrapping security in sensor networks

Bootstrapping security between sensor nodes is a non-trivial problem. The problem generally involves the establishment of some kind of shared secret between nodes that allows them to set up initial secure links with one another, forming an initial secure infrastructure which forms the basis for bootstrapping additional security functionality. We refer to this problem as the *key setup* problem. While one could use a trusted base station to establish keys between neighboring sensors, such an approach generates large volumes of network traffic which is concentrated on the nodes closest to the base station, resulting in bottlenecks, high latency and low reliability. Furthermore, using the base station as a basis for bootstrapping security between sensor nodes would make it an attractive target for compromise – the more priviledges a base station has, the more a successful attacker gains. For example, a base station may store all secret node keys, e.g., in the SPINS architecture [19]. A better architecture would thus distribute security functions (such as secret keys or node authentication functionality) throughout the network in order to minimise any vulnerable spots or bottlenecks in the network.

A key-setup scheme for sensor networks needs to supply the following requirements:

1. Deployed nodes must be able to establish secure node-to-node communication with enough neighbors within communication range in order to form a connected network.
2. The scheme should be functional without involving the base station in security functions.
3. The sensor network must be dynamic, so that additional legitimate nodes deployed at a later time can form secure connections with already-deployed nodes. This implies that key-setup information must always be present and cannot simply be erased after deployment to prevent compromise in event of capture.
4. Unauthorized nodes should not be allowed to establish communications with network nodes and thus gain entry into the network.
5. The scheme must work without prior knowledge of which nodes will come into communication range of each other after deployment.
6. The computational and storage requirement of the scheme must be low. Existing public key schemes cannot be used.

2.3 Evaluation Metrics

Sensor networks have many characteristics that make them more vulnerable to attack than conventional computing equipment. Simply assessing a scheme based on its ability to provide secrecy is hence insufficient. We present several criteria that represent desirable characteristics in a key-setup scheme for sensor networks.

- *Resilience against Node Capture*. We assume the adversary can mount a physical attack on a sensor node after it is deployed and read secret information from its memory. We evaluate a scheme's resilience towards node capture by estimating the fraction of total network communications that are compromised by a capture of x nodes *not including* the communications in which the compromised nodes are directly involved in, which can no longer be secret.
- *Resistance against Node Replication*. Whether the adversary can insert additional hostile nodes into the network after obtaining some secret information (e.g. through node capture or infiltration).

- *Revocation*. Whether a detected misbehaving node can be dynamically removed from the system.
- *Supportable network size*. As the number of nodes in the network grows, the security characteristics mentioned above may be weakened. We give a detailed definition of *maximum supportable network size* in Section 4.2.

Each solution we propose in this paper involves several components. An *initialization* procedure is performed to initialize sensor nodes before they are deployed. After sensor nodes are deployed, a *key setup* procedure is performed by the nodes to set up shared secret keys between some of the neighboring nodes. The keys established in the key-setup phase will then be used as basis to bootstrap secure communications between other nodes.

3 Background: Overview of basic random key predistribution scheme

The basic random key-predistribution scheme is due to Eschenauer and Gligor [11]. In the remainder of this paper, we refer to their approach as the *basic scheme*. Let m denote the number of distinct cryptographic keys that can be stored on a sensor node. The basic scheme works as follows. Before sensor nodes are deployed, an *initialization phase* is performed. In the initialization phase, the basic scheme picks a pool of random keys P out of the total possible key space. For each node, m keys are randomly selected from the key pool P and stored into the node's memory. This set of m keys is called the node's *key ring*. The number of keys in the key pool, $|P|$, is chosen such that two random subsets of size m in P will share at least one key with probability p.

After the sensor nodes are deployed, a *key-setup phase* is performed. The nodes first perform a key-discovery to find out with which of their neighbors do they share a key. Such key discovery can be performed by assigning a short identifier to each key prior to deployment, and having each node broadcast its set of identifiers. Nodes which discover that they contain a shared key in their key rings can then verify that their neighbor actually holds the key through simple challenge-response. The shared key then becomes the key for that link.

After key-setup as described above is complete, a connected backbone graph of secure links is formed. Then nodes can set up *path keys* with nodes in their vicinity with whom they did not happen to share keys with in their key rings. If the graph is connected a path can be found from a source node to its neighbor. The source node can then generate a path key and send it securely via the path to the target node.

One needs to pick the right parameters in order that the graph generated during the key-setup phase be connected. Consider a random graph $G(n, p_l)$, a graph of n nodes for which the probability that a link exists between any two nodes is p_l. Erdös and Rényi showed that for monotone properties of a graph $G(n, p_l)$, there exists a value of p_l over which the property transitions very abruptly from "likely false" to "likely true" [21]. In particular, given a desired threshold of the probability that a random graph is connected, denoted as P_c, we can calculate the necessary expected node degree d in terms of the size of the network n:

$$d = \left(\frac{n-1}{n}\right)(\ln(n) - \ln\ln(P_c)) \tag{1}$$

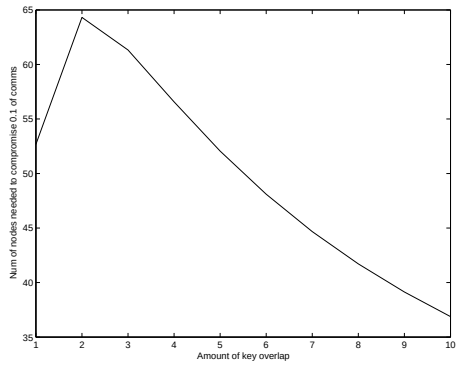
It is observed that $d = O(\log n)$.

For a given density of sensor network deployment, let n' be the expected number of neighbors within communication range of a node. Since the expected node degree must be at least d as calculated above, the required probability p of forming a secure link with some neighbor can be calculated as:

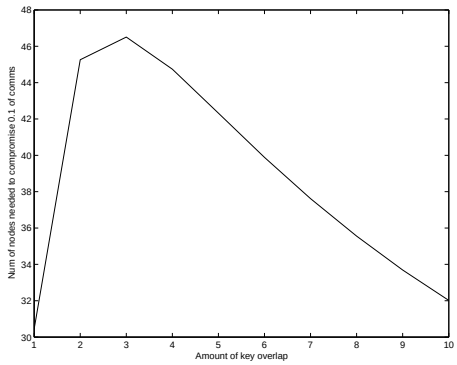
$$p = \frac{d}{n'} \tag{2}$$

4 q-composite random key predistribution scheme

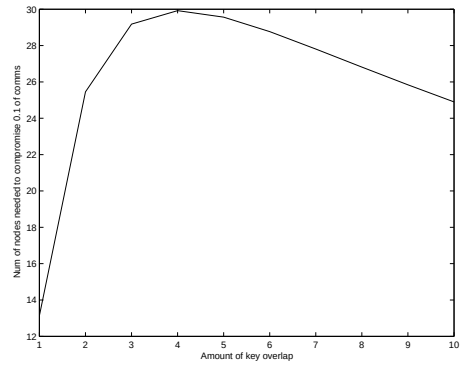
In the basic scheme (see Section 3) two nodes need to find a single common key from their key rings in order to establish communications. We propose a modification to the basic scheme whereby q common keys $(q \geq 1)$



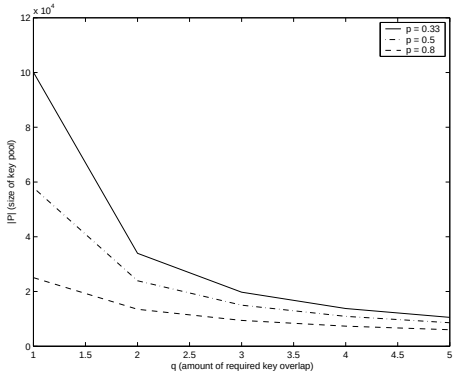
(a) $m = 200$, $p = 0.33$



(b) $m = 200$, $p = 0.5$



(c) $m = 200$, $p = 0.8$



(d) Key pool size $|P|$ vs q for $p = 0.33, 0.5, 0.8$.

Figure 1: Figures (a)–(c) above reflect the number of nodes that the adversary needs to capture before it is able to break any given link with probability 0.1, for various amounts of required key overlap q ($q = 1$ represents the basic scheme). m is the number of keys stored in each node. p is the required probability of any two neighbors being able to set up a link (see Equation 2). Figure (d) reflects the required key pool size (in 10,000s of keys) for various amounts of key overlap, for various p.

are needed, instead of just a single one. By increasing the amount of key overlap required for key-setup, the resilience of the network against node capture and infiltration can be increased.

Figure 1 reflects the motivation for the q-composite keys scheme. As the amount of required key overlap increases, it becomes exponentially harder for an attacker with a given key set to break a link. However, in order to preserve the given probability p of two nodes sharing sufficient keys to establish a secure link, it is necessary to reduce the size of the key pool $|P|$. This allows the attacker to gain a larger sample of P by breaking fewer nodes. The interplay of these two opposing factors results in an optimal amount of key overlap in order to pose the greatest obstacle to an attacker for some desired confidence of breaking a link.

From Figure 1 it can be easily seen that as p increases, the optimal amount of overlap increases. Thus, the q-composite scheme is most suited for scenarios where nodes require a high probability of connecting to any given neighbor.

4.1 Description of the q-composite keys scheme

4.1.1 Initialization and Key Setup

In the initialization phase, we pick an *ordered* set of P random keys out of the total key space, where $|P|$ is computed as described later in Section 4.1.2. For each node, we select m random keys from P and store them into the node's key ring in the order in which they occured in P.

In the key-setup phase, each node must discover all common keys it possesses with each of its neighbors. This can be accomplished with a simple local broadcast of all key identifiers that a node possesses. While broadcast-based key discovery is trivial to implement, it has the disadvantage that a casual eavesdropper can identify the key sets of all the nodes in a network and thus pick an optimal set of nodes to compromise in order to discover a large subset of the key pool P. A more secure, but slower, method of key discovery would be to issue m client puzzles (one for each of the m keys) to each neighbouring node. Any node that responds with the correct answer to the client puzzle is thus identified as knowing the associated key.

After key discovery, each node can identify each neighbor node with which it shares at least q keys. Let the number of actual keys shared be q', where $q' \geq q$. A new communication link key K is generated as the hash of *all* shared keys, e.g., $K = hash(k_1||k_2||\ldots||k_{q'})$. The keys are hashed in some canonical order, for example, based on the order they occur in the original key pool P. Key-setup is not performed between nodes that share fewer than q keys.

4.1.2 Computation of key pool size

Let m be the number of keys that a single node can hold in its key ring. For a chosen overlap parameter q, we wish to generate a pool P of keys, where $|P|$ is chosen such that the probability of any two nodes having at least q keys in common is equal to some given probability p. p is derived from the physical specifications of the network via Equation 2 (see Section 3).

We compute $|P|$ as follows. Let $p(i)$ be the probability that any two nodes have exactly i keys in common. Any given node has $\binom{|P|}{m}$ different ways of picking its m keys from the key pool of size $|P|$. Hence, the total number of ways for both nodes to pick m keys each is $\binom{|P|}{m}^2$. Suppose the two nodes have i keys in common. There are $\binom{|P|}{i}$ ways to pick the i common keys. After the i common keys have been picked, there remain $2(m-i)$ distinct keys in the two key rings that have to be picked from the remaining pool of $|P|-i$ keys. The number of ways to do this is $\binom{|P|-i}{2(m-i)}$. The $2(m-i)$ distinct keys must then be partitioned between the two nodes equally. The number of such equal partitions is $\binom{2(m-i)}{m-i}$. Hence the total number of ways to choose two key rings with i keys in common is the product of the aforementioned terms, i.e., $\binom{|P|}{i}\binom{|P|-i}{2(m-i)}\binom{2(m-i)}{m-i}$. Hence, we have

$$p(i) = \frac{\binom{|P|}{i}\binom{|P|-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{|P|}{m}^2} \tag{3}$$

Let $p_{connect}$ be the probability of any two nodes sharing sufficient keys to form a secure connection. $p_{connect} = 1-$ (probability that the two nodes share insufficient keys to form a connection), hence

$$p_{connect} = 1 - (p(0) + p(1) + \cdots + p(q-1)) \tag{4}$$

For a given key ring size m, minimum key overlap q, and minimum connection probability p, we choose the largest $|P|$ such that $p_{connect} \geq p$.

The variation of $|P|$ with key overlap q and probability of connection p is shown on Figure 1d.

4.2 Evaluation of the q-composite random key distribution scheme

We evaluate the q-composite random key distribution scheme in terms of resilience against node capture and the maximum network size supported. We note that this scheme has no resistance against node replication

since node degree is not constrained and there is no limit on the number of times each key can be used. The scheme can only support node revocation via a trusted base station. Such a revocation scheme is described by Eschenauer and Gligor in their description of the basic scheme [11].

4.2.1 Resilience against node capture in q-composite keys schemes

The q-composite key scheme strengthens the network's resilience against node capture when the number of nodes captured is low. Let the number of compromised nodes be x. Since each node contains m keys, the probability that a given key has not been compromised is $(1 - \frac{m}{|P|})^x$. Hence the expected fraction of total keys compromised is $1 - (1 - \frac{m}{|P|})^x$. For any communication link between two nodes, if its link key was the hash of q_s shared keys, then the probability of that link being compromised is $(1 - (1 - \frac{m}{|P|})^x)^{q_s}$. Hence, we have that the fraction of total communications compromised is

$$\sum_{i=q}^{m} \left(1 - \left(1 - \frac{m}{|P|}\right)^x\right)^i \frac{p(i)}{p_{connect}}$$

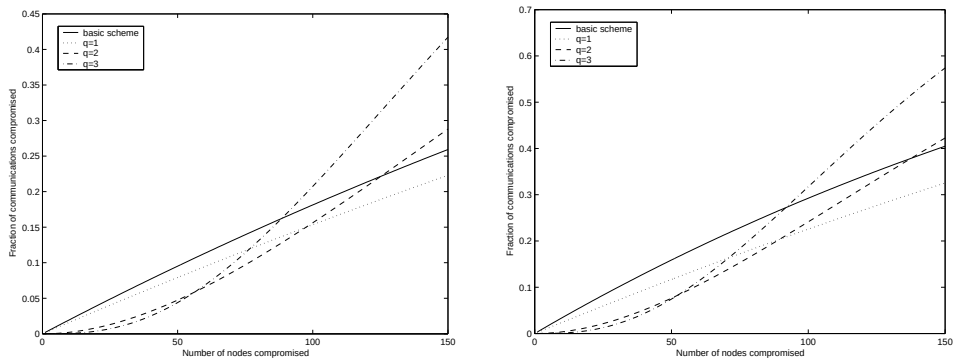
Figure 2 shows the fraction of additional communications (i.e., external communications in the network independent of the captured nodes) that an adversary can compromise based on the information retrieved from x number of captured nodes. It is important to note that the x-axes are absolute numbers of nodes compromised (i.e., independent of the actual total size of the network) while the y-axes are fractions of the total network communications compromised. It is thus immediately clear that the schemes are not infinitely scalable - a compromise of x number of nodes will always reveal y fraction of the total communications in the network regardless of how large the network is. An method to estimate the largest supportable network size of the various schemes is discussed in section 4.2.2.

It can be seen that the q composite keys schemes offer greater resilience against node capture when the number of nodes captured is small. For example, for $q = 2$, the amount of additional communications compromised when 50 nodes have been compromised is 4.74%, as opposed to 9.52% for the basic scheme. However, when large numbers of nodes have been compromised, the q-composite keys schemes tend to reveal larger fractions of the network to the adversary. By increasing q, we make it harder for an adversary to obtain small amounts of initial information from the network via a small number of initial node captures. This comes at the cost of making the network more vulnerable once a large number of nodes have been breached. This is a desirable property because the lowering of initial payoff to the adversary of smaller scale network breaches makes it necessary for them to commit to attacking a significant proportion of the network. This represents a larger commitment of time and resources and effectively deters small scale attacks.

In assessing the effectiveness of the q-composite scheme, we note the intersection points of the lines for q-composite and the basic scheme in Figure 2. For example, in 2a, $q = 2$ is worse than the basic scheme after about 90 nodes have been compromised. The $q = 2$ composite scheme should thus only be implemented in such a network if it is believed that the adversary either does not have the resources or lacks the motivation to exhaustively compromise significantly more than 90 nodes. One potential way of limiting the adversary's motivation to attack is to limit the total number of nodes in the network. This is discussed in the following section 4.2.2. However, there are many scenarios (e.g., safety-related sensors, and some military applications) where even limiting the scale of the sensor network deployment may not be a sufficient to limit an adversary's incentive for mounting a significant attack against the network. In all cases, therefore, a careful study of the application should thus be conducted before the q-composite scheme is selected.

4.2.2 Maximum supportable network sizes for the q-composite keys scheme

Since a fixed number of compromised nodes causes a *fraction* of the remaining network to become insecure, these random-key distribution schemes cannot be used for arbitrarily large networks if resilience against node capture is to be retained. For example, in the basic scheme, the capture of 50 nodes compromises approximately 9.5% of communications in the network. For a network of 10,000 nodes this translates to an approximate



(a) $m = 200$, $p = 0.33$ (b) $m = 200$, $p = 0.5$

Figure 2: The figures show the probability that a specific random communication link between two random nodes A, B can be decrypted by the adversary when the adversary has captured some set of x nodes that does not include A or B. m is the number of keys stored in each node. p is the probability of any two neighbors being able to set up a secure link.

payoff of 10% of communications compromised for a cost to the attacker of capturing just 0.5% of total nodes, representing a huge invitation to attack for potential adversaries.

We can estimate a network's maximum supported size by framing the following requirement:

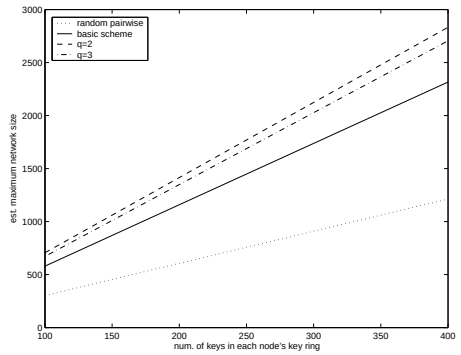
> *Limited global payoff requirement*: Suppose the adversary has captured some nodes, but is only able to break some fraction $f_c \leq f_{threshold}$ of all communications. We require that each subsequent node that is compromised to the enemy allows them to break as many links in the rest of the network, on expectation, as the average connectivity degree of a single node.

In other words, given that the network is still mostly secure ($f_c \leq f_{threshold}$), we would like that, on average, after capturing some node, the adversary does not learn more about the rest of the network than they learn about the communications of the node itself. Via this requirement, smaller scale attacks on a network must be mainly economically justified by the direct communications value of the individual nodes compromised rather than the amount of information that the captured keys can reveal in the rest of the network, thus limiting the incentive of an adversary to begin an attack.

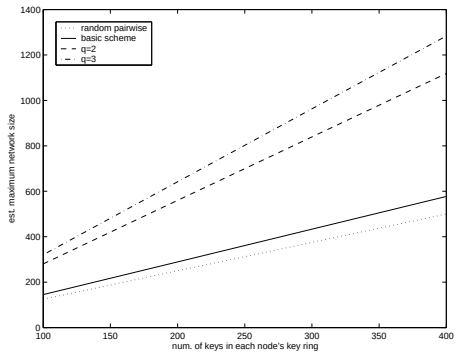
We can thus estimate the maximum allowable sizes for the networks in order that our requirement holds true. Let x be the number of nodes compromised such that some fraction $f_{threshold}$ of the total communications in the network has been compromised. Let the average connectivity degree of a single node be d. The adversary thus holds an expected xd connections in which the compromised nodes are directly involved. We require that the number of *additional* links compromised elsewhere in the network be less than this number of directly compromised links. There are $\frac{nd}{2}$ total links in the network. Hence, the requirement is that $(\frac{nd}{2} - xd) f_{threshold} \leq xd$. Simplifying,

$$n \leq 2x(1 + \frac{1}{f_{threshold}}) \tag{5}$$

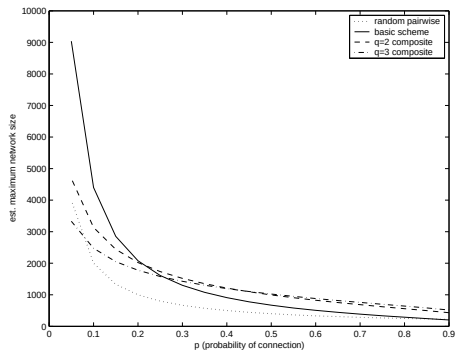
Figure 3 shows the estimated maximum network sizes for the basic random keys scheme as well as for several parameters of the q-composite keys scheme. We note that the maximum network sizes scale linearly with key ring size m. Also, as connection probability between two nodes p increases, the maximum supportable network size for the q-composite keys schemes with larger q perform increasingly well against the basic scheme. This follows from our observation that the q-composite keys schemes performs comparatively better against the basic scheme when p is large.



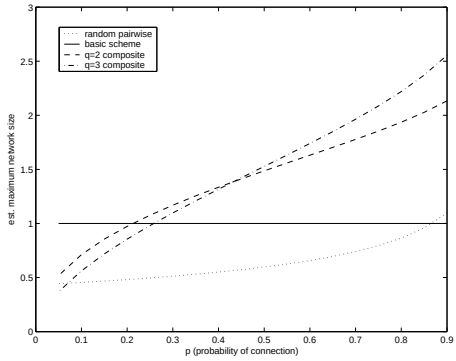
(a) $p = 0.33$, $f_{threshold} = 0.1$



(b) $p = 0.8$, $f_{threshold} = 0.1$



(c) $m = 200$, $f_{threshold} = 0.1$



(d) Maximum network sizes of the q-composite scheme as a fraction of the basic scheme, $m = 200$, $f_{threshold} = 0.1$, various p

Figure 3: Maximum network sizes for various network parameters

5 Multipath Key Reinforcement

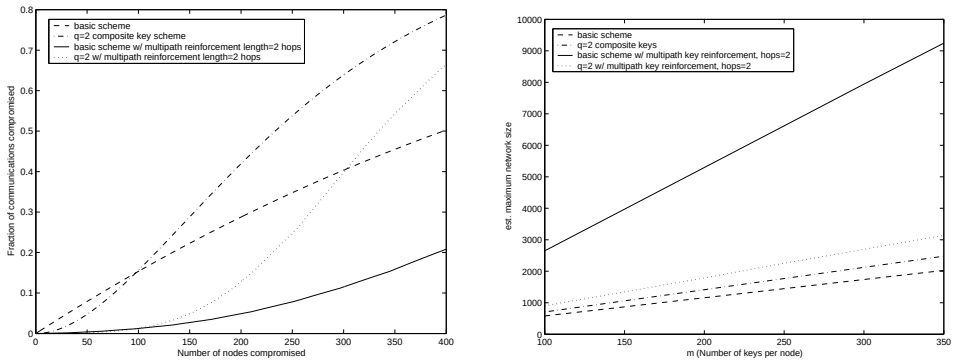
5.1 Description of multipath key reinforcement

We present a scheme to strengthen the security of an established link key by establishing the link key through multiple paths.

We assume that initial key-setup has been completed, i.e., there are now many secure links formed through the common keys in the various nodes' key rings. Suppose A wishes to set up a link key with B (B may not share a secure link with A and may not be within direct communication radius). Assume that during the setup phase enough routing information was exchanged such that A knows all paths to B created during initial key-setup, that are h hops or less. Note that we say $A, N_1, N_2, \ldots, N_i, B$ is a path between created during the initial key-setup if and only if each link $(A, N_1), (N_1, N_2), \ldots, (N_{i-1}, N_i), (N_i, B)$ has established a link key during the initial key-setup using the common keys in the nodes' key rings. Let j be the number of such paths that are *disjoint* (do not have any links in common). A then generates j random keys $k_1, \ldots, k_j$. A then routes each key along a different path to B. When B has received all j keys, then the link key can be computed by both A and B as:

$$k = k_1 \otimes k_2 \otimes \ldots \otimes k_j$$

The secrecy of the link key k is protected by all j random keys. Unless the adversary successfully manages



(a) Resistance against node capture

(b) Maximum network sizes

Figure 4: Multipath key reinforcement results ($m = 200, p = 0.33$)

to eavesdrop on all j paths, they will not know sufficient parts of the link key to reconstruct it. Note that for any given path, the probability that the adversary can eavesdrop on the path increases as the length of the path increases. Also, the probability that the adversary can eavesdrop on all j disjoint paths decreases as j increases. Therefore, in the multipath scheme, we would like to use as many short disjoint paths as possible to improve the security of the established link key.

5.2 Evaluation of multipath key reinforcement

The effectiveness of multipath key reinforcement was investigated in simulation with probability of connection between any 2 nodes $p = 0.33$. Nodes were assumed to know all possible paths of h hops to the target node with which key establishment is being attempted.

Figure 4a indicates the amount of communications compromised vs number of nodes compromised, with and without key reinforcement various schemes. Successfully implementing multipath key reinforcement on a $q = 1$ basic scheme enables it to outperform the q-composite scheme for $q \geq 2$ even when it is supplemented by key reinforcement. This means that if multipath key reinforcement is feasible, it should be implemented on the basic scheme and the q-composite scheme should not be considered. Figure 4b shows the maximum network size of the basic scheme with multipath key reinforcement. It is clear that multipath key reinforcement gives a significant boost to network size performance when implemented on the basic scheme, but has little effect with the q-composite scheme.

Another important effect of multipath key reinforcement is that it hinders the ability of an adversary to specifically attack nodes of high degree in the initial key-setup graph in the hope of catching a maximum of cross traffic. In order to perform such selective attacks successfully when the network has multipath key reinforcement, the attacker has to commit to capturing a long segment of the network, which would be far costlier than attacking a single node.

The cost of the improved security due to multipath key reinforcement is an added overhead in path discovery and key establishment traffic. This cost is likely to be variable based on the underlying routing protocols between the nodes of the sensor network.

6 Random-pairwise keys scheme

In the basic scheme and q-composite keys scheme, there is no capability for node-to-node authentication. All that any given node A knows about a given neighbor B is that A and B share some set of common keys. There is no concept of a unique identity for B. This is because there is no limit to the number of times a key could be

picked for various key rings in different nodes.

In this section, we propose a scheme called the *random pairwise scheme* to address this drawback. The random pairwise scheme has the following properties:

Perfect resilience against node capture Any node that is captured reveals no information about links that it is not directly involved in.

Node-to-node identity authentication Nodes are able to verify the identities of the neighbor nodes with whom they are communicating. An adversary is unable to impersonate the identity of any node B unless B has already been totally compromised.

Distributed Node Revocation without base stations With some added overhead in key storage, misbehaving nodes can be revoked from the network without involving a base station.

Resistance to node replication and generation The scheme can reduce the opportunity of node replication at some cost to node memory and communication setup overhead.

Comparable maximum supportable network sizes vs other schemes without authentication The scheme can support a maximum number of nodes that is comparable to the number of nodes supportable by the basic scheme and q-composite schemes under the limited global payoff requirement framed in section 4.2.2.

6.1 Description of the scheme

Suppose a sensor network has a maximum of n nodes. A simple solution to the key-predistribution problem is the *pair-wise* keys scheme where each node contains $n-1$ communication keys each being pair-wise privately shared with one other node in the network.

The *random pairwise* keys scheme is a modification of the simple pair-wise keys scheme based on the observation that not all $n-1$ keys need to be stored in the node's key ring in order to have a connected random graph with high probability. Erdös and Rényi's formula allows us to calculate the smallest probability p of any two nodes being connected in order that the entire graph is connected with high probability P_c. In order to achieve this probability p in a network with n nodes, each node need only store a random set of np pair-wise keys instead of exhaustively storing all $n-1$. Reversing the calculation, if a node can store m keys, then the maximum supportable network size is

$$n = \frac{m}{p} \tag{6}$$

The maximum supportable network size is thus linear with m, and can be quite large if the nodes are densely packed or the effective communication radius of each node is large. Later we will discuss how this scheme allows us to expand the effective communications radius of the nodes, allowing us to further increase effective network size.

The use of pairwise keys instead of purely random keys chosen from a given pool gives us authentication properties since it allows each node to know the identity of the node that is on the other end of the link. This allows us the option of distributing some security functions away from the base station and onto the nodes themselves.

6.1.1 Initialization and Key-setup in the random pairwise keys scheme

The random pairwise keys scheme proceeds as follows:

- In the initialization phase, a total of $n = \frac{m}{p}$ unique node identities are generated. The actual size of the network may be smaller than n. Unused node identities will be used if additional nodes are added in the network in the future. Each node identity is matched up with m other randomly selected distinct node IDs and a pairwise key is generated for each pair of nodes. The key is stored in both nodes' key rings.

- In the key-setup phase, each node first broadcasts their node IDs into the network to their neighbors. Then a cryptographic handshake is performed between neighbor nodes who wish to mutually prove that they do indeed share a pairwise key from their key rings.
- Once initial key-setup is complete, optional multi-path key reinforcement can be performed to establish link keys between all neighbors and further increase the network's security.

6.1.2 Multi-hop range extension

Because the node ID is just a few bytes, key discovery involves much less network traffic and computational overhead in the nodes than standard random-key predistribution. Hence the effective communication range of nodes for key setup can be extended beyond physical communication range by having neighboring nodes re-broadcast the node ID for a certain number of hops. For example, if the node ID is re-broadcast for 2 hops, then the effective communications range is multiplied by 3. This has a profound impact on the maximum supportable network size n. Recall from Equation 2 that connection probability $p = \frac{d}{n'}$ where n' is the number of neighbours and d was computed via the required probability of graph connectivity. From Equation 6 we have that maximum network size $n = \frac{m}{p}$ where m is the key ring size. Hence

$$n = \frac{mn'}{d} \tag{7}$$

By increasing the effective communications radius x times, we increase the number of neighbours n' by a factor of x^2, hence the maximum supportable network size n also increases by a factor of x^2.

6.1.3 Support for distributed node revocation

In this subsection, we assume the existence of a mechanism in each sensor node that enables it to detect node compromise or node misbehavior in a communicating neighbor. A very crude example here might be interpreting a prolonged lack of communication from a neighbor node to indicate that it is under physical attack. Intrusion detection in sensor networks is a highly challenging problem, one that currently lacks a satisfactory solution. However, in order for any such scheme to function effectively, the underlying key management system for the sensor network must support some method to revoke the secret information of any nodes that have been identified as compromised. We present such a revocation scheme in the hope that the problem of intrusion detection may be satisfactorily addressed in the near future.

In the random pairwise scheme, node revocation can be supported via base stations as described by Eschenauer and Gligor [11]. This has the disadvantages mentioned in section 2.1, i.e., the base station acts as a single-point-of-failure, and also may slow the node revocation process due to the potential high latency between the nodes and the base-station. In revocation, fast response is particularly crucial since a detected attack must be sealed off from the network before it can do significant harm.

As an alternative to using base stations, a distributed node revocation scheme can be implemented for the random pairwise scheme by having neighboring nodes broadcast 'public votes' against a misbehaving node (we use the term *public vote* since the identity of the voter in this case need not be kept secret). If any node B observes more than some threshold number t of public votes against some node A, then B breaks off all communications with A. By listening on the network (like any other sensor node), the base station can relay the votes back to the physically secure location where the undeployed nodes are stored. There, any as-yet undeployed node identities react appropriately by erasing any pairwise keys associated with A from the undeployed nodes' key rings. This has the effect of permanently severing node A from the network.

Designing a compact and efficient distributed public vote counting mechanism for sensor nodes is non-trivial. We require the voting scheme to have the following properties:

1. Each voting member must be able to cast a valid vote, while no voting member is able to forge another member's vote.

2. Each voting member must be able to verify the validity of a broadcasted public vote.

3. Broadcasted public votes from one voting member reveal no information that would allow listeners to generate additional public votes.

4. Broadcasted public votes have no replay value.

A straightforward scheme can be as follows: Consider a node A, which, like every other node in the network, has m keys in its key ring. Since all the keys are issued to exactly 2 nodes and no 2 keys are issued to the same pair of nodes, we have exactly m nodes that share a pairwise key with node A. We call this set of m nodes the set of *voting members* of A. Each of these m voting members are assigned a random voting key k_i. It also knows the respective hashes of the voting keys of all the $m-1$ other voting members, i.e., $hash(k_1), hash(k_2), \ldots, hash(k_{i-1}), hash(k_{i+1}), \ldots, hash(k_m)$. To cast a public vote against A, the node broadcasts k_i. All other voting members can verify the vote by computing $hash(k_i)$. Once k_i is verified, voting members can replace $hash(k_i)$ with k_i and a flag reflecting the fact that this vote has already been heard on the network.

One problem with this scheme is that each entry on the key ring now stores not only the pairwise key but also $m-1$ hash values and a voting key. Hence, if m pairwise keys are stored on the node, the memory requirement is $O(m^2)$.

An alternative method we propose to reduce the memory requirement is to use Merkle trees [17] to compactly represent the entire array of m hash values. In doing so, only a single verifying hash value (the root value of the Merkle tree) needs to be stored, but the voting information is now size $O(\log m)$, since in order to vote, each node needs to reveal not just its secret voting key but also the hash values of the $\log m$ internal nodes in the Merkle tree that will allow listeners to verify the validity of the vote.

Choice of t parameter. t is the minimum number of votes needed to revoke a node. t is chosen low enough such that it is unlikely that any node has degree $< t$ in the network, but high enough such that rogue nodes cannot cause the revocation of innocent nodes. For any of the m keys in a node's key ring, the probability that it is used is the probability that the other node which has this key is within communication radius. This probability is $\frac{n'}{n}$ since there are n' neighbors out of n total nodes, that will be within communication radius. The distribution of the degree of a node is hence binomial $(m, \frac{n'}{n})$. Since $n = \frac{mn'}{d}$(from Equation7), $\frac{n'}{n}$ simplifies to $\frac{d}{m}$. Hence we have that the degree of a node is binomial$(m, \frac{d}{m})$, the average is d and the variance is $d(1-\frac{d}{m})$. For key ring sizes sufficient to support a reasonably sized network, $\frac{d}{m}$ will be small. Hence the variance is close to the average d, i.e., the distribution is heavily skewed to the left.

The expected degree of a node d increases very slowly with network size n (from Equation 1, $d = O(\log n)$). Hence t should remain small (≤ 5) even for large networks. Since t is small, we note that memorising previously cast votes to prevent replay is not a significant memory cost.

One consequence of implementing such a voting scheme is that it cannot be allowed that any node have less than t neighbors, otherwise it cannot be revoked. Since t was chosen such that it is unlikely that any node has degree $< t$ in the network, the scheme can be modified such that any node that is unable to form at least t connections on the network after the key-setup phase must be revoked. Such low-degree nodes can be detected via the degree-counting mechanism described in the following subsection 6.1.4.

Resisting revocation attack. One possible weakness associated with this node revocation scheme is that each node holds the potential to cast a vote against m other nodes. Since the total number of nodes $n = \frac{m}{p}$, this represents a significant fraction of the node population. Hence only a fixed number of nodes need to be compromised without detection in order for them to revoke a significant proportion of the network, regardless of the network size.

To prevent widespread release of revocation keys by compromised nodes, we require that only nodes that are actually in direct communication with some node B have the ability to revoke node B.

We do this by distributing the revocation keys to the potential neighbours of B in a deactivated form, i.e. each neighbour i stores its revocation key k_i masked (XORed) with some secret S_i. This deactivated key will

not hash to the correct verifying value and is thus useless for voting. During the key discovery and setup phase, if node i wishes to complete key setup with node B, it requires node B to transmit its activation secret S_i (and vice-versa). Once node i has received S_i it unmasks k_i using S_i, and verifies that it was given the correct unmasking secret by performing vote verification on the unmasked k_i to see if it is a valid revocation key. Storage of m masking factors on node B takes only $O(m)$ space and is insignificant compared to the total $O(m \log m)$ space needed to store the voting and verification apparatus.

Such a policy of need-to-know key activation ensures that the majority of revocation keys recovered through node capture are in an unusable masked state. In order to use these revocation keys to revoke some node A the adversary now has to physically communicate with A and complete key-setup for up to t new connections.

Via this mechanism the adversary's ability to attempt sabotage via this course of action is seriously limited through the implementation of schemes to limit node replication and node generation (see next Section 6.1.4). In general, since resistance against node replication imposes an upper limit d_{max} on the degree of a node, the number of revocation keys issuable by each compromised node is limited to d_{max}.

Even if we do not assume implementation of schemes for resisting node replication, the requirement that the adversary establish physical (1-hop) communication with a target node is a strong disincentive to mount a DoS attack via revocation. For example, if disruption rather than subversion of the network is all that is desired by the adversary and the adversary has the ability to physically communicate with the target nodes, then a radio jamming attack is probably cheaper and more productive than a revocation attack.

It is important to note that the vote-activation mechanism presented above merely limits the damage an adversary can inflict by broadcasting node revocations – it does not completely eliminate the potential for such an attack.

It should also be noted that threshold-based node revocation might be countered if several nodes have been compromised without detection. In such a case, several compromised nodes can surround a misbehaving node and effectively shield it from being revoked, since only communicating neighbors have revocation rights. In such a case, proper revocation is still possible depending on the sensitivity and accuracy of the detection mechanism. However, designing an intrusion detection mechanism that has both high sensitivity and accuracy is an extremely challenging problem.

6.1.4 Resistance against node replication and node generation

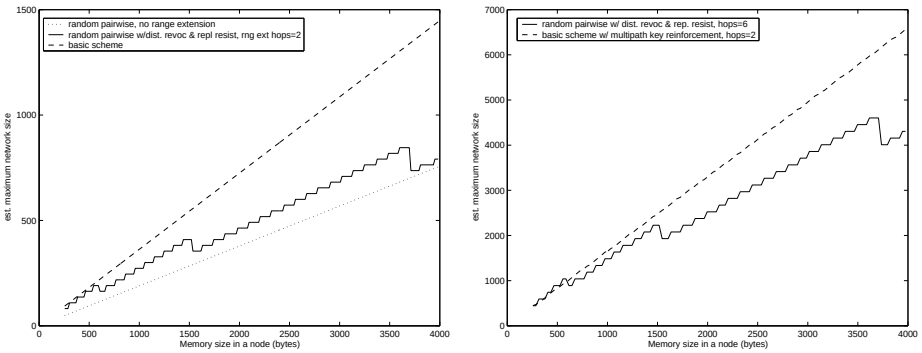
In the event that node capture goes undetected by the network, it is desirable that the network be resistant against the addition of infiltrator nodes derived from captured nodes, especially considering that resistance may be required to prevent revocation attack on the network (see section 6.1.3)

In order to prevent unconstrained node replication on the network, the degree of any node can be limited. We know that the degree of a node on the network is approximately binomially $(m, \frac{d}{m})$ with expectation d and variance close to d (see Section 6.1.3 for derivation). Hence very few nodes should have degree $\geq 3d$, for example. This implies that we can limit the degree of nodes to d_{max} where d_{max} is some small multiple of d, without disrupting network connectivity.

It is notable that since $d = O(\log n)$ (Equation 1 and $n = \frac{m}{p}$ (Equation 6), $d = O(\log m)$ and hence $d_{max} = O(\log m)$. Hence d_{max} will be very small compared with the total *potential* connectivity m.

Since the random-pairwise scheme allows us to have a notion of authenticated node identity, a method for node-degree counting for the random-pairwise scheme may be implemented with the public-vote counting scheme presented in section 6.1.3. The operation of the degree-counting scheme is exactly identical. Each node contains a voting key and some way to verify valid voting keys. Each time a given node A forms a connection with some node B, A broadcasts its voting key for B and vice-versa. Each node can thus track the degree of all m of its potential neighbors, and refuse to form new connections if the degree becomes too large.

One concern in this case is that we now need to memorize d_{max} number of cast votes instead of a small number t. However, because $d_{max} = O(\log m)$ this problem does not increase storage complexity since the voting mechanism will dominate the storage requirement in any case. Various sparse-storage directory



(a) (b)

Figure 5: Network sizes for random pairwise key setup compared against the basic scheme with and without multipath key reinforcement. Link keys are 128bits, hash values are 80bits in this simulation. $p = 0.33, f_{threshold} = 0.1$

mechanisms could also be implemented to reduce the storage requirement, such as the Coarse Vector [14] and Tristate [1] protocols.

6.2 Evaluation of the random keys scheme

Perfect resilience vs node capture. Since each pairwise key is unique, capture of any node does not allow the adversary to decrypt any additional communications in the network besides the ones that the compromised node is directly involved in. This would be represented in Figure 2 as the line y=0.

Maximum supported network size. The limited global payoff requirement of Section 4.2.2 cannot be used to compute the maximum network size of the random pairwise keys scheme because global information revealed from local node capture is always 0. Rather, the maximum network size of a random pairwise keys scheme is fixed at design time by Equation 6.

The maximum supportable network size for a random pairwise key scheme without distributed node revocation or range extension through repeated broadcast is shown in Figure 3. We note that the random pairwise keys scheme performs comparatively better when the probability of node-node link-forming p is low. Figure 5a reflects the network sizes for random pairwise scheme with all the features mentioned earlier including range extension. It can be seen that with the range extension of just two hops, we can get network sizes comparable to the other schemes in this case. Also, the $\log m$ cost of including distributed node revocation does not significantly impact maximum network size. Figure 5b compares the random pairwise keys scheme with the basic scheme combined with multipath key reinforcement[1]. It can be seen that by extending the range by 6 hops, the random pairwise scheme can achieve comparable network sizes against the basic scheme with multipath key reinforcement. Given that the random pairwise scheme also has perfect resilience against node capture and authentication features, this is an excellent result.

Resistance to revocation attack of distributed scheme. If resistance against node replication is implemented, then the theoretical number of nodes an attacker can revoke per successful node captured is $\frac{d_{max}}{t}$ which is $O(d)$. Since $d = O(\log n)$, the effectiveness of revocation attack scales only slowly with $\log n$ as network size n increases. This makes it unlikely that an attacker would find it economically worthwhile to launch a revocation attack on the network, especially considering that they must physically establish communications with every node that they wish to revoke.

[1]Applying multipath key reinforcement on the random pairwise scheme has no effect on maximum network size. This configuration is hence not evaluated here.

7 Related Work

We first review work in establishing shared keys in mobile computing, then review work in sensor network key establishment.

Tatebayashi, Matsuzaki, and Newman consider key distribution for resource-starved devices in a mobile environment [23]. Park et al. [18] point out weaknesses and improvements. Beller and Yacobi further develop key agreement and authentication protocols [4]. Boyd and Mathuria survey the previous work on key distribution and authentication for resource-starved devices in mobile environments [6]. The majority of these approaches rely on asymmetric cryptography. Bergstrom, Driscoll, and Kimball consider the problem of secure remote control of resource-starved devices in a home [5].

Stajano and Anderson discuss the issues of bootstrapping security devices [22]. Their solution requires physical contact of the new device with a master device to imprint the trusted and secret information.

Carman, Kruus, and Matt analyze a wide variety of approaches for key agreement and key distribution in sensor networks [8]. They analyze the overhead of these protocols on a variety of hardware platforms.

Wong and Chan propose a key exchange for low-power computing devices [24]. However, their approach assumes an asymmetry in computation power, that is, one of the participants is a more powerful server.

Perrig et al. propose SPINS, a security architecture specifically designed for sensor networks [19]. In SPINS, each sensor node shares a secret key with the base station. To establish a new key, two nodes use the base station as a trusted third party to set up the new key.

We review the related work by Eschenauer and Gligor [11] in Section 3. Anderson and Perrig propose a key establishment mechanism for sensor networks based on initially exchanging keys in the clear [2], their key infection approach is secure as long as an attacker arrives later and did not eavesdrop on the initial key exchange.

Zhou and Haas propose to secure ad hoc networks using asymmetric cryptography [25]. Kong et al. propose localized public-key infrastructure mechanisms, based on secret sharing and multiparty computation techniques [15]. Such approaches are expensive in terms of computation and communication overhead.

Broadcast encryption by Fiat and Naor [12] is another model for distributing a shared key to a group of receivers. However, this model assumes a single sender, and that the sender knows the key pools of all receivers. Subsequent papers further develop this approach [16, 13, 3].

8 Conclusion

Efficient key distribution for sensor networks is of critical importance for the deployment of secure sensor network applications. Local processing of sensor data requires secure neighbor node to neighbor node communication. We present several efficient random key predistribution schemes for solving the key-setup problem in resource-constrained sensor networks. The differing performance of the schemes we have described for various values of probability of inter-node linkage p suggests a natural partition of the application space.

When p is large, i.e., the density of neighbors is low relative to the effective communication radius, then the q-composite keys scheme is advantageous. Such networks would typically be characterized by small, cheap sensors with highly limited communications capabilities deployed around a base station.

When p is small, i.e., there are many neighbors within a relatively larger effective communication radius, and when base stations cannot be reliably secured against attack or when node-to-node authentication is important, then the random pairwise scheme is advantageous. Such networks would probably be characterized by somewhat more sophisticated nodes with larger memories and longer communications ranges, and multiple redundant base stations that act as data collection points. An example of such a sensor network would be a military sensor net deployed to detect chemical or biological agents.

Finally, whenever the routing protocol allows path discovery to be performed at low cost, then multi-path key reinforcement should be implemented regardless of the underlying initial key-setup scheme. This technique greatly strengthens the security of any scheme against node compromise.

A Notation

We list the symbols used in the paper below:

d	the expected degree of a node – i.e., the expected number of links a node can establish during key-setup.
$f_{threshold}$	if the enemy compromises a fraction of communications below $f_{threshold}$, then the limited global payoff requirement must hold. Used for computing network size via Equation 5
m	number of keys in a node's key ring
n	network size, in nodes
n'	number of neighbor nodes of a node within communication radius
p	probability of 2 neighbor nodes being able to set up a secure link between them during the key-setup phase.
P_c	desired confidence level (probability) that the sensor network is connected.
P	key pool (set of keys randomly chosen from the total key space)
$\|P\|$	size of key pool
q	for the q-composite scheme, required amount of key overlap
t	threshold number of votes after which a node will be revoked.

References

[1] Anant Agarwal, Richard Simoni, Mark Horowitz, and John Hennessy. An evaluation of directory schemes for cache coherence. In *Proceedings of the 15th Annual International Symposium on Computer Architecture*, pages 280–289, 1988.

[2] Ross Anderson and Adrian Perrig. Key infection: Smart trust for smart dust. Unpublished Manuscript, personal communication, November 2001.

[3] Dirk Balfanz, Drew Dean, Matt Franklin, Sara Miner, and Jessica Staddon. Self-healing key distribution with revocation. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 241–257, Oakland, CA, May 2002. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press.

[4] M. Beller and Y. Yacobi. Fully-fledged two-way public key authentication and key agreement for low-cost terminals. *Electronics Letters*, 29(11):999–1001, May 1993.

[5] Peter Bergstrom, Kevin Driscoll, and John Kimball. Making home automation communications secure. *IEEE Computer*, 34(10):50–56, Oct 2001.

[6] Colin Boyd and Anish Mathuria. Key establishment protocols for secure mobile communications: A selective survey. In *Australasian Conference on Information Security and Privacy*, pages 344–355, 1998.

[7] Michael Brown, Donny Cheung, Darrel Hankerson, Julio Lopez Hernandez, Michael Kirkup, and Alfred Menezes. PGP in constrained wireless devices. In *9th USENIX Security Symposium*, August 2000.

[8] David W. Carman, Peter S. Kruus, and Brian J. Matt. Constraints and approaches for distributed sensor network security. *NAI Labs Technical Report #00-010*, September 2000.

[9] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, November 1976.

[10] John R. Douceur. The sybil attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, March 2002.

[11] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communication Security*, November 2002.

[12] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology: CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*. Springer, 1994.

[13] J. Garay, J. Staddon, and A. Wool. Long-lived broadcast encryption. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO '2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 333–352. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 2000.

[14] Anoop Gupta, Wolf-Dietrich Weber, and Todd Mowry. Reducing memory and traffic requirements for scalable directory-based cache coherence schemes. In *Proceedings of the 1990 International Conference on Parallel Processing (Vol. I Architecture)*, pages 312–321, 1990.

[15] Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, and Lixia Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *9th International Conference on Network Protocols (ICNP'01)*, 2001.

[16] M. Luby and J. Staddon. Combinatorial bounds for broadcast encryption. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 512–526. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1998.

[17] Ralph Merkle. Protocols for public key cryptosystems. In *1980 IEEE Symposium on Security and Privacy*, 1980.

[18] C. Park, K. Kurosawa, T. Okamoto, and S. Tsujii. On key distribution and authentication in mobile radio networks. In *Advances in Cryptology - EuroCrypt '93*, pages 461–465, 1993. Lecture Notes in Computer Science Volume 765.

[19] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Seventh Annual ACM International Conference on Mobile Computing and Networks (MobiCom 2001)*, Rome, Italy, July 2001.

[20] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[21] J. Spencer. *The Strange Logic of Random Graphs*. Number 22 in Algorithms and Combinatorics. Springer-Verlag, 2000.

[22] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In B. Christianson, B. Crispo, and M. Roe, editors, *Security Protocols, 7th International Workshop*. Springer Verlag, 1999.

[23] M. Tatebayashi, N. Matsuzaki, and D. B. Jr. Newman. Key distribution protocol for digital mobile communication systems. In *Advances in Cryptology - Crypto '89*, pages 324–334, 1989. Lecture Notes in Computer Science Volume 435.

[24] Duncan S. Wong and Agnes H. Chan. Efficient and mutually authenticated key exchange for low power computing devices. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT '2001*, Lecture Notes in Computer Science, Gold Coast, Australia, 2001. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany.

[25] Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13(6):24–30, November/December 1999.