The Effects of Probabilistic Key Management on Secure Routing in Sensor Networks

Patrick Traynor, Guohong Cao and Tom La Porta
Networking and Security Research Center
Department of Computer Science and Engineering
The Pennsylvania State University
Email: {traynor,gcao,tlp}@cse.psu.edu

***Abstract*— Secure data dissemination in wireless ad hoc and sensor networks has recently received a great deal of attention. A variety of protocols have been proposed in order to ensure secure data delivery across these systems; however, the majority of these schemes assume the presence of public or pre-established symmetric keys. Accordingly, the cost of key management has not been incorporated into secure routing mechanisms in this setting. This paper considers the expenses incurred by sensor networks implementing secure routing schemes on top of probabilistic symmetric key management schemes. Specifically, we examine the overhead observed from proactive and reactive key establishment mechanisms for networks using a balanced method of key management. Through extensive simulation, we quantify more realistic costs for the application of secure hop-by-hop routing in sensor networks.**

Keywords: Sensor Networks, Probabilistic Key Management, Secure Routing

I. Introduction

Wireless sensor networks are rapidly becoming invaluable tools for the distributed aggregation and processing of data. From monitoring inventory and environmental conditions to vehicle tracking, these systems allow their administrators to observe remote and or dangerous locations in a cost-effective fashion. One of the most important challenges facing sensor networks is that of secure data dissemination. The focus of a large pool of literature in recent years, secure data dissemination is absolutely essential to network fidelity as nodes are completely reliant upon their neighbors to forward packets to their intended destinations. In a setting where individuals must be skeptical of the activities of their neighbors, simply trusting that an adjacent node is performing the expected operations on our data would be naïve.

Providing security for wireless sensor networks is a particularly difficult task. Whereas nodes in wired networks are able to leverage considerable resources in terms of power, processing ability and positive human/administrator interaction along with centralized, trusted servers, sensor nodes typically lack all of the aforementioned means. Accordingly, security solutions for sensor networks must typically be distributed, robust and as efficient as possible.

While a number of schemes have been suggested to make the interactions between neighbors more secure in MANETs, the majority of works have failed to truly translate to a sensor network environment as the key management strategies implemented by those methods are infeasible in this setting.

A well received solution to the issue of key management in sensor networks is to distribute a certain number of randomly selected keys in each of the nodes throughout the network. This method has been extended by a number of researchers [3], [4], [5], [6], [9], [13], [18]. Using this approach, one can achieve a known probability of connectivity within a network while minimizing the resources necessary to implement the security system. The specific consequences of this or any other approach when applied to routing and data dissemination, however, have not yet been explored.

In this paper, we make the following contributions:

- **Performance Evaluation:** Through extensive simulation, we explore the full costs of hop-by-hop secure routing in sensor networks implementing probabilistic key management.
- **Keying Strategies:** We examine how the timing and frequency of re-keying requests directly affects performance and overhead.

The remainder of the paper is organized as follows: Section II discusses specific related works in the fields of both secure routing and probabilistic key management. Section III presents the protocol and keying strategies used to establish pair-wise secure relationships. Section IV presents and analyzes performance data for a number of scenarios. Finally, concluding remarks and future directions are offered in Section V.

II. Related Work

Some of the work done in secure routing for MANETs has suggested that a cryptographic interaction between the endpoints of communication is necessary to implement secure data exchange. For example, in Papadimitratos, et al. [10], the source and destination nodes share each others' public keys. When the DSR route request (RREQ) launched by the source arrives at the destination, the target node signs the list of nodes used to reach it with its private key. Upon receiving the response, the sender can be sure that the RREQ indeed did reach the desired node and secure interaction can begin.

A number of other papers in this domain have proposed the use of neighbor-based authentication. When an intermediary node in the route receives an RREQ in the ARAN protocol [12], it examines the certificate and contents encrypted by the private key of the previous hop. If the decrypted value correctly corresponds to the hash value of the packet, the current node then removes the previous certificate and adds its own before encrypting and forwarding the packet on to its next hop.

Ariadne [7], [8] combines the previous two approaches by having both the target and intermediary participants take part in the authentication of routing data. Before sending the RREQ, the source computes an HMAC using a key k_{SD} shared between itself and destination. During route discovery, each node along the source-destination path authenticates routing information with its Tesla [11] key. The destination then buffers the packet until the requisite amount of time passes for each intermediary to release its Tesla key before processing the packet.

The above solutions, and a number of additional approaches [1], [2], [17], make simplifying assumptions when the issue of key management is raised. The majority of these schemes default to the presence and capability of nodes to use public-key cryptography to accomplish their goals; however, such approaches are not possible in sensor networks due to the previously discussed limitations of node capabilities. Those papers not specifically requiring the non-repudiation characteristics inherent to public-key methods instead typically state that symmetric keys already exist between nodes but do not discuss the method in which those keys have been established.

The probabilistic distribution of keys, which allocates a given number of randomly selected keys in each node, makes it possible to attain a known probability of connectivity within a network. The seminal work in this field was presented by Eschenauer and Gligor [6]. In this scheme, a large pool of P keys is generated, from which k are randomly selected, without replacement[1], for each sensor node. Two nodes may communicate to directly establish a session key if they have a key match. Further extensions to the balanced probabilistic distribution of keys scheme have been proposed by a number of researchers [3], [4], [5], [9], [18], [13].

The advantages of such probabilistic approaches are numerous. Most importantly, they allow for secure communications to be possible in resource constrained platforms while using very little memory for key storage. In a setting with exceedingly limited resources, where hop-by-hop authentication for secure routing is a desirable goal, such a keying scheme is particularly attractive.

Because probabilistic methods of keying are so well suited for sensor networks, we investigate the costs incurred by secure routing solutions built upon the balanced [6] method. We examine two approaches for establishing shared keys between neighbors. In the first, nodes proactively establish keys with their direct neighbors; we call this proactive keying (PK). In the second, nodes only establish keys when they are required; we call this method reactive keying (RK).

The details of these and the protocols required to establish pairwise keys between neighbors are discussed in the following section.

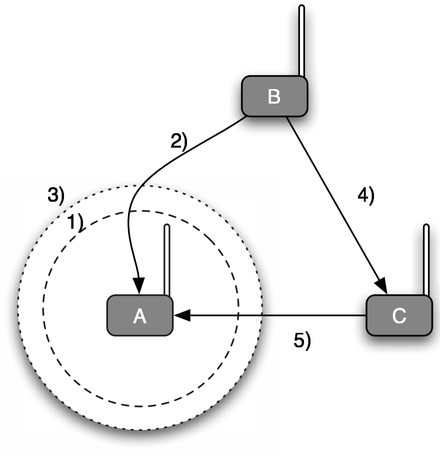

Fig. 1. Node A, attempting to establish keys with its neighbors, 1) broadcasts its key identifiers. Node B, which has a key that corresponds to one of the identifiers, 2) responds to A's challenge-response broadcast. After waiting for some time, 3) A launches a request for assistance message so that it can establish a key with C. B, already sharing a key with A, 4) establishes a session key between A and C on A's behalf. B then 5) contacts A and gives it a copy of the session key encrypted by B with the key it shares with A.

III. Overview

A. Protocol Specification

We now offer a short overview of the protocol used for key distribution. For reasons of space, a full description of this protocol is available in Traynor, et al [14]. Figure 1 also provides a short tutorial.

Before deployment, all nodes receive k randomly selected keys from a pool P as described in Section II [6]. Nodes attempting to establish a secure relationship with their neighbors broadcast a set of key identifiers corresponding to their pre-deployed keys. The source node then waits for challenge-responses from its neighbors with which keys are shared. During the challenge-response, session keys are established between nodes with a key match.

To establish keys with the nodes without pre-deployed matching keys, an indirect key match phase commences. After the expiration of the direct-phase timer, a node desiring a secure relationship with its remaining neighboring nodes launches a request for assistance, which includes the node IDs with which it desires to establish keys. Nodes that already have a secure relationship with the source can use their secure relationships with other nodes on the transmitted list to help establish a session key. If a node attempting to establish keys with all of its neighbors should still lack a key with any of its neighbors, it could simply launch the indirect key match phase using the newly keyed nodes as new hops for indirection.

B. Keying Strategies

Determining when to initiate the protocol discussed above to establish session keys has important effects on performance. We define both the proactive keying (PK) and reactive keying (RK) with direct neighbors. With PK, upon detection of a previously unseen node, a node proactively establishes a key

[1] The keys of individual nodes are chosen without replacement to prevent multiple copies of keys being stored on a node. Each node is able to select its keys from P.

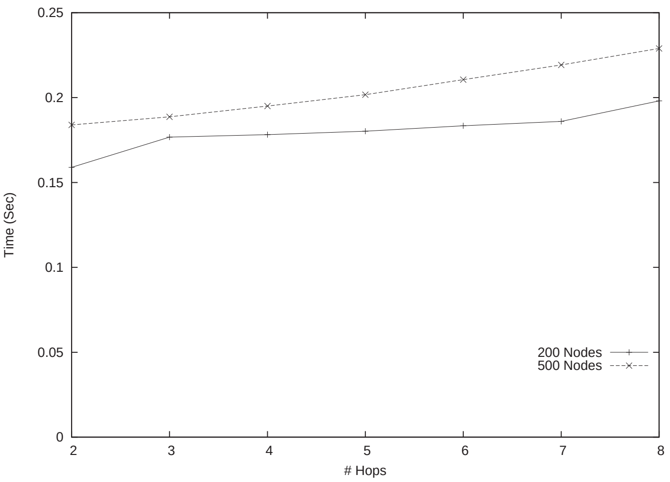

Fig. 2. The average delay (sec) to establish a route between two nodes under the PK keying scheme. Because key establishment occurs proactively, it does not directly affect message delivery.

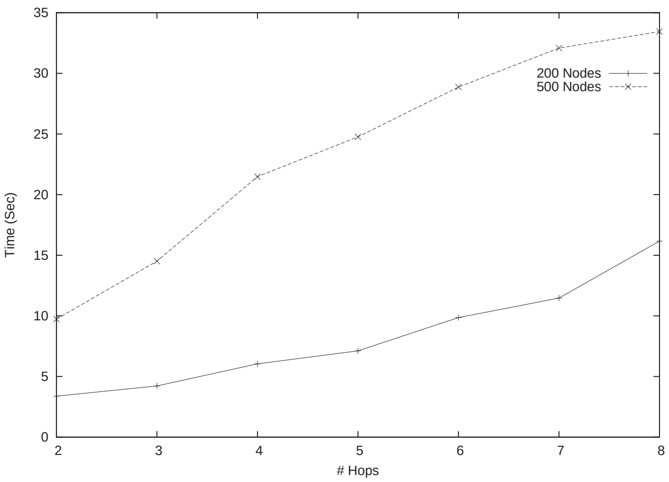

Fig. 3. The average initial delay (sec) to establish a route between two nodes under the RK keying scheme. Delivery latency is greatly increase because keying does not occur until data needs to be delivered.

TABLE I

THE AVERAGE LATENCY TO ESTABLISH KEYS

Scenario	Latency (sec)	
	200 Nodes	500 Nodes
Single Node w/ Neighbors	1.6119	4.7757
All Nodes w/ Neighbors	99.1952	251.6091
All Nodes w/ All Nodes	473.880	1408.232

so that this node is a viable first hop over which a message can be sent. In effect, nodes always maintain keys with their neighbors. PK has the benefit of faster route establishments and faster local route repairs at the expense of possibly wasteful key matching exchanges. Under RK, nodes only establish secure relationships when there exists a need to do so. This will lead to higher latencies in establishing routes, but will limit unnecessary overhead.

We first evaluate the performance in an environment in which nodes are not mobile. This provides insight into the overhead of the basic key matching protocol. We then evaluate the performance impact of PK and RK on a secure version of AODV using a protocol based on ARAN [19] for the random-waypoint mobility model.

We consider cases in which session keys have an infinite lifetime and in which keys expire after a period of time. With PK, when neighbors have a key that is about to expire, they perform a simple point-to-point handshake to renew the session key. Using RK, nodes that are actively communicating (i.e., have exchanged data within 100 seconds) also renew expiring keys with a simple handshake. Keys established with nodes that are not in active communication expire and are re-established only if required by the key matching protocol described above.

IV. SIMULATION AND DISCUSSION

We evaluated the effects of probabilistic keying on secure routing using the Network Simulator (ns-2) version 2.27. Each simulation was run over two different field configurations: 200 and 500 nodes in a 500m x 500m test-bed. Nodes were given a maximum transmission range of 75m and used an 802.11 MAC layer.

Routing was accomplished via the built-in implementation of the AODV protocol. `HELLO` messages were launched every two seconds for PK and once every 100 seconds for RK. The timer for transmitting a request for assistance was set to 0.5 seconds. In both the 200 and 500 node density cases, all nodes responded within 0.1 seconds of the initial broadcast of key identifiers. By allowing this extra time, it was our hope that we would allow as many nodes as legitimately possible to establish keys directly.

The exchange of keying information is designed to allow the secure routing and delivery application data. Every 100 seconds, some number of nodes transmit 50, 500-byte packets over the course of 0.5 seconds. The selection of source and destination for these exchanges of data are random.

Initial node placements and their subsequent movements were generated by ns-2's included scenario generation utility. Where applicable, mobile nodes move at between zero and five meters per second according to the random-waypoint mobility model.

The keys stored in each of the nodes were generated a priori using the library function `random()`. Each key scenario generation was also seeded with the current time. The key pool, P, was of size 10,000 and each node received $k = 83$ keys [6].

A. Static Network Evaluation

In this scenario we determine the average amount of time required to create secure relationships between nodes in a static network using the key matching protocol described in Section III.

We consider two general cases: all nodes establishing keys with all other nodes in the network and all nodes establishing keys with their neighbors. Although the case in which each node establishes keys with every other node in the network is not feasible for reasons of scalability, we have included a

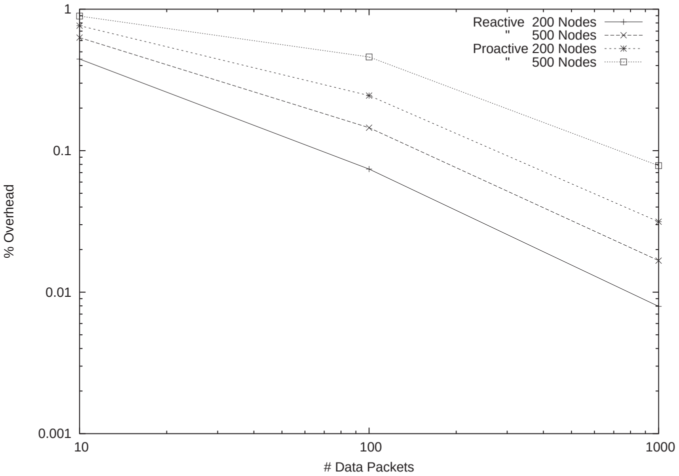

Fig. 4. The overhead created by additional packet transmissions for key establishment.

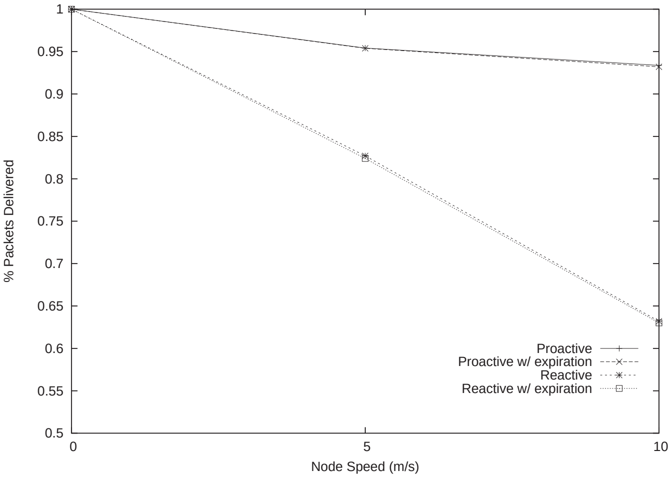

Fig. 5. The observed packet loss for PK and RK schemes for individual mobility. Each run consisted of 500 nodes. There is negligible difference between the scenarios with infinite versus 500 second key lifetimes.

simulation to demonstrate its complications and understand its characteristics.

The results of network initialization testing is shown in Table I. In a network as small as 200 nodes with an average of approximately four and worst case of 11 hops per message, the network requires almost eight minutes to perform complete $n - 1$ pair-wise keying. The more dense case of 500 nodes, with the same average and worst case numbers for hops per message, takes almost 23.5 minutes to complete. It is clear that it is not possible to use complete pair-wise keying if network initialization time is a concern. Such latencies for key establishment are extremely high and are the result of contention for the air interface.

There are benefits to implementing this scheme on a small scale. Once initialization completes, a wide range of secure data dissemination and routing protocols using end-to-end cryptographic operations could be conducted in parallel with hop-based measures. Additionally, by placing the expense of keying before the mission-oriented operations of the network take place, the cost of delivering messages end-to-end is equivalent to the examples given in the previously cited literature. The addition of key expirations consequently, would destroy any benefits yielded by such an approach.

When bootstrapping the network using the PK strategy, each node attempts to begin keying as soon as they become active. In the case where there are 200 nodes this process can take just under two minutes. The scenario containing 500 nodes requires over four minutes. Like the previous $n-1$ case, the dominant factor contributing to such long key-establishment times is contention for the air interface. This competition naturally worsens with increasing node density. With RK, however, there is no bootstrapping phase as keying occurs on demand.

The routing latency trade-off for PK versus RK for a static network is demonstrated in Figures 2 and 3. The advantage of PK, as shown in Figure 2 is that once the bootstrapping phase is complete, hop-by-hop secure routing protocols perform well. Figure 3 shows the initial cost of route requests when using RK. The on-demand nature of RK route discovery takes significantly longer than under the PK approach, although the two schemes eventually converge given enough traffic. The decreased latency under PK is the result of placing its costs up front and allowing the key establishment to be amortized over the lifetime of the network.

The static nature of these networks influences keying overhead. For a system initialized using PK, no new key establishments are necessary after the bootstrapping period. Under RK, new keys will be established until all nodes have keys with all neighbors. Once keys have been established for all nodes, the performance of systems running RK versus PK converge.

B. Mobile Network Evaluation

Because mobile sensor networks are being increasing studied [15], [16], we examine the effects of probabilistic keying on secure routing in dynamic environments. While route discovery times eventually converge in static networks, latency remains at approximately the levels shown in Figures 2 and 3 for both the PK and RK approaches throughout the mobile simulations. Performance is dampened because intermediary nodes are required to establish keys with new neighbors for almost every route request transmission. It is therefore crucial to understand the characteristics of a network of mobile nodes as it moves towards stead-state.

Figure 4 shows the observed overhead per packet and packet loss for PK and RK for two different node densities. As expected, PK has significantly higher overhead than RK because nodes establish keys even if no data is ultimately sent. The simulations demonstrated that apart from the multiple nodes added during the first iteration of the key matching protocol, new nodes are typically added one at a time. Accordingly, each new neighbor encountered requires a given node to launch the key establishment protocol. Such an extremely high overhead is the result of sending such a small amount of data between nodes. The benefit of this elevated overhead is decreased routing latency. Low-intensity, latency critical applications such as reporting the presence or identity of a newly acquired target can be expected to demonstrate such characteristics. If the nature of the communication is critical,

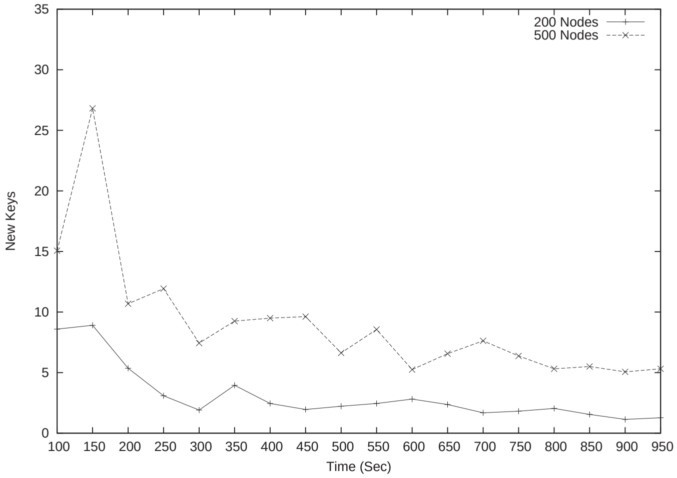

Fig. 6. The average number of new keys established per node per 50 seconds with PK key management and infinite key lifetimes.

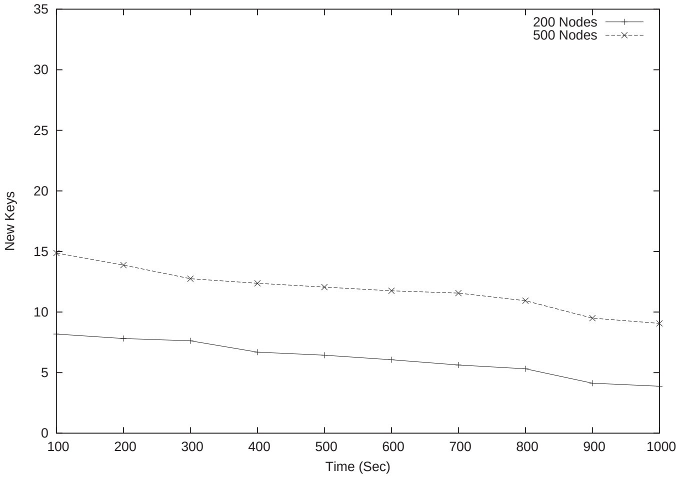

Fig. 8. The average number of new keys established per node with RK given transmissions every 100 seconds and infinite key lifetimes.

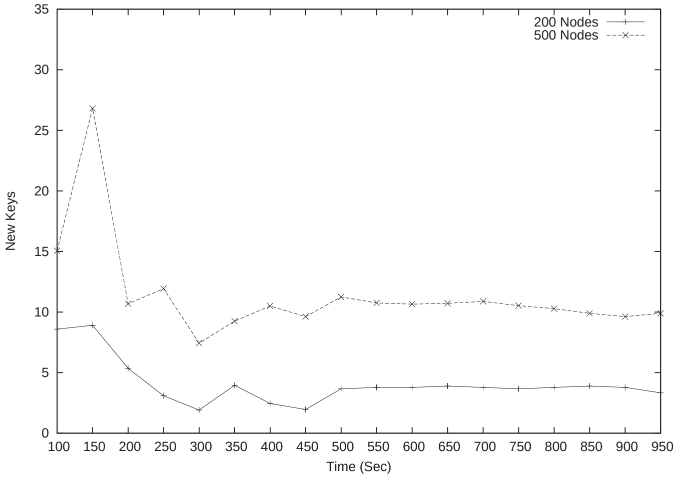

Fig. 7. The average number of new keys established per node per 50 sec using PK with 500 second key lifetime.

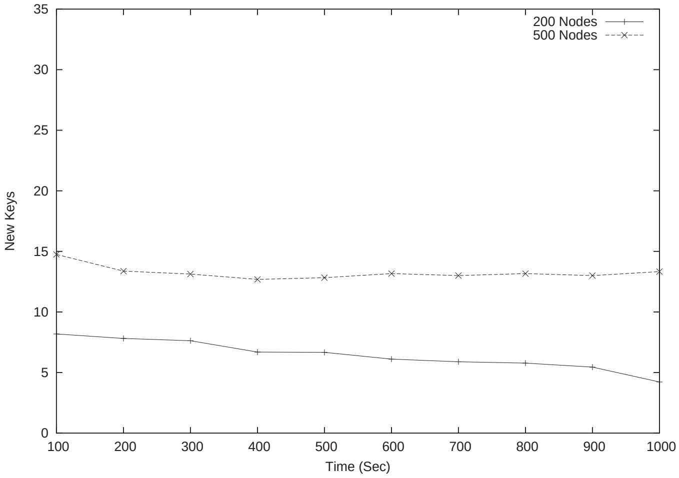

Fig. 9. The average number of new keys established per node with RK with transmissions every 100 seconds and a key lifetime of 500 seconds.

the steep cost of overhead is more acceptable; however, non-critical communication with low intensity is expensive enough that the overall life of the network may be reduced.

The relationship between traffic intensity and overhead, as demonstrated in Figure 4, exhibits an important attribute of probabilistic keying. The correspondence between these two elements is linear, demonstrating that overhead is not a function of traffic intensity. As the amount of mobility increases during a given period, however, so too does the overhead. This result is intuitive as nodes encounter more new neighbors as their mobility increases. Mobility being the dominant factor, which was verified by additional testing with increased node velocities, is intuitive as the very nature of PK requires key establishment with all newly encountered nodes.

Because the RK approach establishes keys only at transmission time, it incurs overhead at almost a per RREQ rate. This data alone may be enough to be the deciding factor when selecting a keying strategy for a network of mobile sensors. Should the mobility of individual nodes be high and the number of destinations for and frequency of data delivery be low, the RK approach may be more beneficial; however, should mobility be low and the number RREQs become large, an implementation of PK could prove more advantageous to the operations of the network. Before any judgment is made, it is necessary to consider the consequences of these approaches on packet loss.

Figure 5 depicts the effects of keying on the packet loss ratio. In the static network simulations, none of the simulations exhibited any dropped UDP packets. This result is not totally unexpected as the frequency of keying in both the PK and RK cases dies off quickly as neighbors are unchanging. When nodes become mobile, we see that the drop rates for the RK scheme exceed that of PK. There are two main reasons for these losses. The first, and potentially more crucial, is the result of routes being broken due to the latency associated with keying RK keying. The route established by the RREQ is often no longer valid by the time data transmission occurs. The second and more obvious is collisions. While the PK case was almost constantly attempting to establish secure relationships with new neighbors, the effects of high medium access were effectively amortized as the keying traffic at the time of data transmission was less than in the RK scenario. Figure 5 also demonstrates that the expiration of keys has minimal effect packet loss.

The effect of key expirations is shown in Figures 6 - 9. Figure 6 shows the number of new keys established in an average node (recorded every 50 seconds) when using PK and infinite key lifetimes. The trending in this figure demonstrates that nodes retaining shared keys with every node encountered allows the network to eventually reach a keying steady-state of zero. The ability to implement a sensor network based on such a practice, for reasons of freshness, security, and memory constraints, however, is not necessarily realistic. Figure 7 shows the same measurement with unused key lifetimes set to 500 seconds. The higher steady-state value for key establishments per quanta (≈ 10 for high density scenarios) reinforces that the overhead associated with keying will always be present in real systems.

RK with and without elapsing keys is demonstrated in Figures 8 and 9, respectively. The RK method approaches steady-state keying much more slowly than PK. The steady-state values for the RK method with 500 second key lifetimes are also much higher than the PK method. While the number of keys established per broadcast closely approaches the average value attained PK, RK is not quite able to reduce this number to that of its competitor during the 1,000 second simulation. Like PK, however, RK exhibits a constant overhead (≈ 14 for high density scenarios) of key establishments per data burst. Accordingly, any protocol implementing either the PK or RK approach as the basis of their key establishment mechanism must recognize the significant overhead for implementing such a scheme.

V. Conclusion

Secure routing is an extremely important element of safeguarding sensor networks; however, the majority of previous works in this area neglect to specifically address the difficult task of key management. Although a number of papers have explored probabilistic keying schemes, none have applied them to an application such as secure routing and observed the issues inherent to its implementation.

We have shown how the use of a probabilistic key management approach affects latency, overhead and the packet loss ratio in environments where static and independently mobile sensors are present. If keying occurs proactively such that all new nodes are automatically keyed regardless of the transience of the pair relationship, the latency of data delivery is greatly reduced, but at a great increase in overhead. With reactive keying, the number of unnecessarily established secure relationships is minimized, but only at the cost of tremendously high latency. This latency is often the reason for packet loss when the mobility of individual nodes becomes high. Furthermore, we have demonstrated that the overhead associated with these methods of key management is a function of the mobility of the nodes and not of the traffic intensity.

If an emergency were to arise, it may take the reactive approach too much time to establish a route for its data to be useful. However, because sensor nodes are absolutely constrained by power restrictions, the reactive scheme may be the only practical scheme that can be implemented in a network that is meant to be long lived. We therefore recommend that a hybrid method of key management be investigated such that the positive attributes of the two above schemes can be combined to make secure routing in sensor networks as robust as possible.

References

[1] B. Awerbach, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *Proceedings of the ACM Workshop on Wireless Security WiSe*, September 2002.

[2] S. Capkun and J. Hubaux. BISS: Building Secure Routing out of an Incomplete Set of Security Associations. In *Proceedings of the ACM Workshop on Wireless Security WiSe*, September 2003.

[3] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2003.

[4] W. Du, J. Deng, S. Han, and P.K. Varshney. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2003.

[5] W. Du, J. Deng, Y.S. Han, S. Chen, and P.K. Varshney. A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In *Proceedings of IEEE INFOCOM*, 2004.

[6] L. Eschenauer and V. Gligor. A key management scheme for distributed sensor networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2002.

[7] Y. Hu and A. Perrig. A Survey of Secure Wireless Ad Hoc Routing. May 2004.

[8] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[9] D. Liu and P. Neng. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2003.

[10] P. Papadimitratos and Z.J. Haas. Secure Routing for Mobile Ad Hoc Networks. In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2002.

[11] A. Perrig, R. Canetti, D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. In *RSA CryptoBytes*, volume 5(2), pages 2–13, 2002.

[12] K. Sanzgiri, Bridget Dahill, B. Levine, C. Shields, and E. Belding-Royer. A Secure routing Protocol for Ad Hoc Networks. In *Proceedings of the IEEE International Conference on Network Protocols*, 2002.

[13] P. Traynor, H Choi, G Cao, S Zhu, and T La Porta. Establishing Pair-Wise Keys in Heterogeneous Sensor Networks. Technical Report NAS-TR-0001-2004, Network and Security Research Center, Department of Computer Science, Pennsylvania State University, November 2004.

[14] P. Traynor, R. Kumar, H. Bin Saad, G. Cao, and T. La Porta. LIGER: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. Technical Report NAS-TR-0008-2005, Network and Security Research Center, Department of Computer Science, Pennsylvania State University, July 2005.

[15] G. Wang, T. La Porta, and G. Cao. Movement-Assisted Sensor Deployment. In *Proceedings of IEEE INFOCOM*, 2004.

[16] G. Wang, T. La Porta, G. Cao, and W. Zhang. Sensor Relocation in Mobile Sensor Networks. In *Proceedings of IEEE INFOCOM*, 2005.

[17] S. Yi, P. Naldurg, and R. Kravets. Security-Aware Ad Hoc Routing for Wireless Networks. In *Proceeding of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2001.

[18] S. Zhu, S. Zu, S. Setia, and S. Jajodia. Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach. In *Proceedings of the IEEE International Conference on Network Protocols*, 2003.