A Routing-Driven Public-Key Cryptosystem Based Key Management Scheme for a Sensor Network

Satya Venkatesh Kadali[#1], O.Srinivasa Rao[*2], Dr MHM Krishna Prasad[#3]

[#]*Dept of CSE, JNTUK_UCEV*
Vizianagaram, Andhrapradesh, India

**Associate Professor, Dept of IT, JNTUK_UCEV*
Vizianagaram, Andhrapradesh, India

***Abstract*— In homogeneous sensor networks, many-to-one traffic pattern dominates and hence a sensor may only communicate with a small portion of its neighbors. So, the homogeneous sensor networks have poor performance and scalability. Most existing key management schemes try to establish shared keys for all pairs of neighbor sensors, no matter whether these will communicate with each other or not, and this causes large overhead. This project proposes a novel routing-driven key management scheme, which establishes shared keys only for those neighboring sensors that communicate with each other by using RSA public-key algorithm along with Quantum Key Distribution Protocols (QKDPs). Quantum cryptography easily resists replay and passive attacks. Classical cryptography enables efficient key verification and user authentication. This work integrates the advantages of these two techniques along with QKDPs so that the eavesdroppers can be detected, replay attacks can be easily avoided. This project also proposes how user authentication and session key verification can be done in a single step process at the receiving end.**

***Keywords*—Security, key management, sensor networks, Quantum Key Distribution Protocols (QKDPs).**

I. INTRODUCTION

As the wireless technologies has become the need of an hour, Securing sensor networks has received much attention in the last few years and as so many research works are going on in order to achieve stronger security and to reduce overhead to the maximum possible extent on wireless networks created a strong interest in me to do some work concerning security issues on wireless sensor networks. Wireless sensor networks have applications in many areas, such as military, homeland security, health care, environment, agriculture, manufacturing, and so on. In the past several years, sensor networks have been a very active research area. Most previous research efforts consider homogeneous sensor networks, where all sensor nodes have the same capabilities. However, a homogeneous ad hoc network suffers from poor fundamental limits and performance.

Security is critical to sensor networks deployed in hostile environments, such as military battlefield and security monitoring. A number of literatures have studied security issues in homogeneous sensor networks, e.g., [6], [7]. Key management is an essential cryptographic primitive up on which other security primitives are built. Due to resource constraints, achieving such key agreement in wireless sensor networks is non-trivial. In [6], Eschenauer and Gligor first present a key management scheme for sensor networks based on probabilistic key pre distribution. Several other key pre-distribution schemes (e.g., [7]) have been proposed. In this paper, we present an efficient key management scheme that only needs small storage space. The scheme achieves significant storage saving by utilizing 1) the fact that most sensor nodes only communicate with a small portion of their neighbors; 2) efficient public-key cryptography.

Definition: c-neighbor: A neighbor sensor node v is referred to as a communication neighbor (c-neighbor) of sensor node u if v is in a route from u to the sink. Based on the above observation, we propose a novel idea for efficient key management in sensor networks. A key management scheme only needs to set up shared keys for each sensor and its c-neighbors, i.e., it does not need to set up shared keys for each pair of neighbor sensors. The new scheme can significantly reduce the overhead of key establishment in sensor networks. For example, suppose that a sensor node u has 30 neighbors but only sends packets to 2 neighbors (e.g., one primary next-hop node and one backup). Using traditional key management schemes, 30 pair wise of keys need to be established for u, one key for each neighbor. Using c-neighbor concept, only 2 pair wise keys need to be set up for u, one for each c-neighbor. thus, the new scheme can significantly reduce communication and computation overheads, and hence reduce sensor energy consumption.

Public-key cryptography has been considered too expensive for small sensor nodes, because traditional public-key algorithms (such as RSA) require extensive computations and are not suitable for tiny sensors. However, the recent progress on Elliptic Curve Cryptography (ECC) [10] provides new opportunities to utilize public-key cryptography in sensor networks. The recent implementation of 160-bit ECC on Atmel ATmega128, a CPU of 8Hz and 8 bits, shows that an ECC point multiplication takes less than one second [11], which demonstrates that the ECC public-key cryptography is feasible for sensor networks. Compared with symmetric key cryptography, public-key cryptography provides a more flexible and simple interface, requiring no key pre-distribution, no pair-wise key sharing, and no complicated one-way key chain scheme.

ECC can be combined with Diffie-Hellman approach to provide key exchange scheme for two communication parties.

ECC can also be utilized for generating digital signature, data encryption and decryption. The Elliptic Curve Digital Signature Algorithm (ECDSA) utilizes ECC to generate digital signature for authentication and other security purposes [12], [13]. Several approaches for encryption and decryption using ECC have been proposed [10], [12]. Please refer to references [10], [12], [13] for the details. In this paper, we present an efficient key management scheme for HSNs. The scheme utilizes the c-neighbor concept and ECC public-key cryptography. Typical sensor nodes are unreliable devices and may fail overtime. Our key management scheme considers topology change caused by node failures. That is, the scheme set up pair wise keys for each sensor with more than one neighbor. In case the primary next hop node fails, a backup node is used for communications. In addition, if there is a need for two neighbor sensor nodes to set up shared keys later (e.g., in case all backup nodes fail); they can do this with the help from other neighbors [6]. The contributions of this paper are three folds. First, we observed the fact that a sensor only communicates with a small portion of its neighbors and utilized it to reduce the overhead of key management. Second, we designed an effective key management scheme for HSNs by taking advantage of powerful H-sensors. Third, we utilized a public key algorithm - ECC for efficient key establishment among sensor nodes. The rest of the paper is organized as follows.

II. THE ROUTING STRUCTURE IN HSNS

In this Section, we present an efficient key management scheme for HSNs which utilizes the special communication pattern in sensor networks and ECC. The scheme is referred to as ECC-based key management scheme. We consider an HSN consisting of two types of sensors: a small number of high-end Sensors (H-sensors) and a large number of low-end sensors L-sensors). Both H-sensors and L-sensors are powered by batteries and have limited energy supply. Clusters are formed in an HSN. For an HSN, it is natural to let powerful H-sensors serve as cluster heads and form clusters around them. First, we list the assumptions of HSNs below.

• Due to cost constraints, L-sensors are NOT equipped with tamper-resistant hardware. Assume that if an adversary compromises an L-sensor, she can extract all key material, data, and code stored on that node.

• H-sensors are equipped with tamper-resistant hardware. It is reasonable to assume that powerful H-sensors are equipped with the technology. In addition, the number of H-sensors in an HSN is small (e.g., 20 H-sensors and 1,000 L-sensors in an HSN). Hence, the total cost of tamper-resistant hardware in an HSN is low.

• Each L-sensor (and H-sensor) is static and aware of its own location. Sensor nodes can use a secure location service such as [14] to estimate their locations, and no GPS receiver is required at each node.

• Each L-sensor (and H-sensor) has a unique node ID.

• The sink is trusted.

A. The Cluster Formation:

After sensor deployment, clusters are formed in an SN (Sensor Network) and designed an efficient clustering scheme for SNs [9]. For the simplicity of discussion, assume that each H-sensor can communicate directly with its neighbor H-sensors (if not, then relay via L-sensors). All H-sensors form a backbone in an SN. After cluster formation, a SN is divided into multiple clusters, where H-sensors serve as the cluster heads. An illustration of the cluster formation is shown in Fig:1, where the small squares are L-sensors, large rectangular nodes are H-sensors, and the large square at the bottom-left corner is the sink. For the ease of execution, I considered all H-sensor, L-sensors in a single host machine and confined all nodes to communicate in a single cluster where each H-sensor can directly communicate with any of its L-sensors(if the node is not a neighbor, then it can relay via other L-sensors).

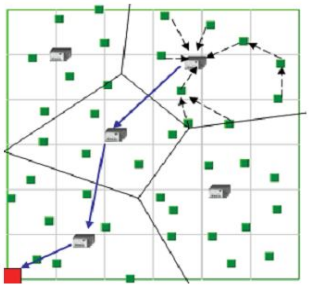


Fig 1: Cluster Formation in an HSN.

B. Routing in SNs:

In a SN, the sink, H-sensors and L-sensors form hierarchical network architecture. Clusters are formed in the network and H-sensors serve as cluster heads. All H-sensors form a communication backbone in the network. Powerful H-sensors have sufficient energy supply, long transmission range, high date rate, and thus provide many advantages for designing more efficient routing protocols [6]. Routing in a SN consists of two phases: 1) Intra-cluster routing – each L-sensor sends data to its cluster head via multi-hops of other L-sensors; and 2) Inter-cluster routing - a cluster head (an H-sensor) aggregates data from multiple L-sensors and then sends the data to the sink via the H-sensor backbone. The routing structure in an SN is illustrated in Fig:1. An intra-cluster routing scheme determines how to route packets from an L-sensor to its cluster head. The basic idea is to let all L-sensors (in a cluster) form a tree rooted at the cluster head H. (1) If complete data fusion is conducted at intermediate nodes, then a minimum spanning tree (MST) consumes the least total energy in the cluster. (2) If there is no data fusion within the cluster, then a shortest-path tree (SPT) can be constructed using either a centralized or distributed algorithm. It consumes the least total energy.

III. THE ROUTING-DRIVEN KEY MANAGEMENT SCHEME

In this paper we discuss about the uses of RSA public-key Cryptosystem for key generation and integrated QKDP's for key distribution.

A. Key Generation:

The first step in RSA encryption is to generate a key pair. Two keys are generated of which one is used as the public key and the other is used as the private key. The keys are generated as follows:

- Generate two large random primes p and q.
- Compute n which is equal to product of those two prime numbers, n = pq
- Compute $\varphi(n) = (p-1)(q-1)$.
- Choose an integer e, $1 < e < \varphi(n)$, such that $\gcd(e, \varphi(n)) = 1$.
- Compute the secret exponent d, $1 < d < \varphi(n)$, such that $ed \equiv 1 \pmod{\varphi(n)}$.
- The public key is (n, e) and the private key is (n, d). The values of p, q, and $\varphi(n)$ should also be kept secret.
    - n is known as the modulus.
    - e is known as the public exponent or encryption exponent.
    - d is known as the secret exponent or decryption exponent.

B. Key Distributions

For key distribution process, QKDP's were used with RSA in order to distribute the keys to neighboring nodes by the cluster header (which acts as a Trusted Center).

1. Quantum Cryptography:

Quantum cryptography is only used to produce and distribute a key, not to transmit any message data. With the use of Quantum cryptography, the two communicating parties can be able to detect the presence of any third party trying to gain knowledge of the key. For secure communications, Quantum key distribution protocols (QKDP's) are used. It enables two parties (sensors) to produce a shared random bit string known only to them, which can be used as key to encrypt and decrypt the messages. Quantum cryptography easily resists replay and passive attacks. An unique property of quantum cryptography is providing the ability to the both communicating users to detect the presence of any third party trying to gain knowledge of the key by using quantum super positions or quantum entanglement and transmitting information in quantum states, by this eavesdroppers can be detected.

2. *Key Management Scheme:*

This technique involves encoding information in quantum states (Qu-bits) as opposed to classical communications use of bits. Usually, photons are used for these quantum states. QKD divided into two main categories depending on which property they exploit.

- Prepare and measure protocols (Calculate the amount of information that has been intercepted).
- Entanglement based protocols (Two quantum states of two (or more) separate objects can become linked together in such a way that they must be described by combined quantum states, not as individual objects).

IV. EXPERIMENTAL EVALUATION

TC (Trusted Center)-Cluster header and participant synchronize their polarization basis according to pre-shared secret key. During session key distribution, the pre-shared secret keys together with random string are used to produce another encryption key to encipher the session key. By this, a receiver will not receive the same polarization qu-bits even if identical session key is retransmitted. Hence, the secrecy of pre-shared secret key can be preserved and thus this secret key can be long term and repeatedly used between TC and participant. Due to combined use of classical cryptographic techniques over quantum channel, a receiver can authenticate user identity, verify the correctness and freshness of the session key and detect the presence of eavesdroppers.

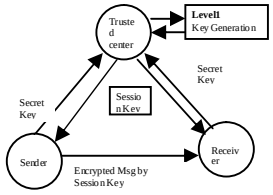


Fig 2: Distribution of Session Key with Quantum Cryptography.

The below figure depict the formation of cluster with the neighboring nodes in a network. Among the existing nodes, any node can be a cluster header and the remaining nodes have to register with the particular header in order to participate in the communication process.

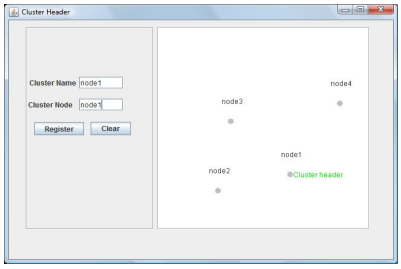


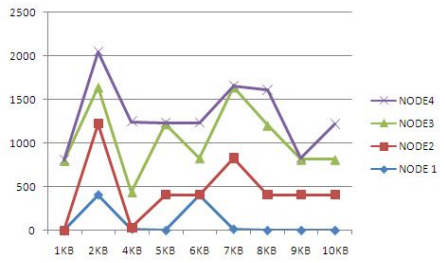
Fig 3: Formation of a Cluster

In this paper, a model was implemented which provides internal security in a network and also an efficient key management scheme has been proposed for a sensor network. This scheme utilizes the fact that a node communicates with only a small portion of its neighbors and thus greatly reduces the communication and computation overheads of key setup. A public-key algorithm RSA is used along with QKDP's to further improve the key management scheme for generation and distribution of secret keys. These

keys were used to encrypt, transmit and decrypt sensitive data being shared among nodes within a network.

This work can be extended in real world heterogeneous sensor networks by making use of Elliptic Curve Cryptography (ECC) algorithm to achieve stronger information security. Authentication would be still provided in an easier manner by making use of ECC algorithm on wireless sensor networks. By making use of ECC, further reduction in storage space, computational overheads, power consumption could be achieved because of its shorter key length.

In the below graph, we compare the total energy consumption of using the centralized ECC key management scheme and the E-G scheme. The energy consumption reported here only includes the energy used to set up security keys, but does not include the energy for data communications. In the simulation, the number of L-sensors varies from 200 to 1200, with an increase of 200. The number of H-sensors under the ECC scheme is always 20. For the E-G scheme, the key pool size is P = 16,000 and the number of pre-loaded keys in each sensor is m = 150, thus, the key-sharing probability is about 90%. Under the ECC scheme, a sensor only establishes shared key with communication neighbors. Denote the number of communication neighbors as n. We measure the energy consumption of the ECC scheme for different values of n, including 2, 6 and 12, where 12 mean that a sensor sets up keys with every neighbor. The simulation results are reported in below graph. The graph shows that the ECC key management scheme consumes much less energy than the E-G scheme (including the case when n = 12), and the ECC scheme achieves more energy saving for larger networks. We obtain similar results for the distributed ECC key management scheme.



V. Conclusions

In this Paper this model was implemented which provides internal security in a network and also an efficient key management scheme has been proposed for a sensor network. This scheme utilizes the fact that a node communicates with only a small portion of its neighbors and thus greatly reduces the communication and computation overheads of key setup. A public-key algorithm RSA is used along with QKDP's to further improve the key management scheme for generation and distribution of secret keys. These keys were used to encrypt, transmit and decrypt sensitive data being shared among nodes within a network.

References

[1] WATRO R, et al. TinyPK: secuting sensor networks with public key technology. Proceedings of the 2nd ACM Workshop on Secutity of Ad hoc and Sensor Networks. New York, 2005. 135-142

[2] Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney, Jonathan Katz, Aram Khalili, "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks", ACM Journal Name, Vol. V, No. N, Month 20YY, 2005.

[3] David J. Malan, "Toward PKI for Sensor Networks" Division of Engineering and Applied Sciences. Harvard University malan@eecs.harvard.edu. 8 November 2004.

[4] Arjan Durresi, Vijay Bulusu, Vamsi Paruchuri, Mimoza Durresi, Raj Jain, "Key Distribution in Mobile Heterogeneous Sensor Networks" direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings-(2006).

[5] Jeremy Brown, Xiaojiang Du, Kendall Nygard, "An Efficient Public-Key-Based Heterogeneous Sensor Network Key Distribution Scheme" Nygard, "Global Telecommunications Conference, GLOBECOM '07. IEEE 26 December 2007.

[6] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," in Proc. 9th ACM Conference on Computerand Communication Security, pp. 41-47, Nov. 2002.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proc. 2003 IEEE Symposium on Security and Privacy, May 2003, pp. 197-213.

[8] Yong Ma, Siddharth Dala1, Majd Alwan, James Aylor, "ROP: A Resource Oriented Protocol for Heterogeneous Sensor Networks" Wireless Communications, vol. 6, no. 9, pp. 3395–3401, in 2007.

[9] K. Whitehouse, C. Sharp, E. Brewer, and D. Culler, "Hood: a neighborhood abstraction for sensor networks," in Proc. ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '04), Boston, MA, June, 2004.

[10] N. Koblitz, "Elliptic curve cryptosystems," Mathematics of omputation, vol. 48, pp. 203-209, 1987.

[11] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in Proc. 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston, MA, Aug. 2004.

[12] N. Koblitz, A Course in Number Theory and Cryptography, 2nd ed. Graduate Texts in Mathematics, vol. 114, Springer, 1994.

[13] I. Blake, G. Seroussi, and N. Smart, Elliptic Curves in Cryptography, London Mathematical Society, Lecture Note Series 265, Cambridge University Press, 1999.

[14] L. Lazos and R. Poovendran, "SeRLoc: secure range-independent localization for wireless sensor networks," in Proc. 2004 ACM workshop on Wireless security (ACM WiSe 2004), Philadelphia, PA..

[15] Qing Yang, Qiaoliang Li, Sujun Li, "An Efficient Key Management Scheme for Heterogeneous Sensor Networks" Networks, ICON 2008. 16th IEEE International Conference on Dec 2008.

[16] F. Amin, A. H. Jahangir, and H. Rasifard, "Analysis of Public-Key Cryptography for Wireless Sensor Networks Security" Embedded End-to-End Wireless Security with ECDH Key Exchange, the 46th IEEE in 2008

[17] A.S.Poornima, B.B.Amberker, "Tree-based Key Management Scheme for Heterogeneous Sensor Networks" This paper appears in: Networks, 2008. ICON 2008. 16th IEEE International Conference Dec. 2008..

[18] T.Kavitha, D.Sridharan, "Security vulnerabilities in Wireless Sensor Networks: A Survey" Mobile - Wireless Communications, Security Management in 2009.

[19] P. Mackenzie, "More efficient password authenticated key exchange" CT-RSA, pages 361 – 377, 2001.

[20] RSA Laboratories. "Frequently Asked Questions About Today's Cryptography".2005. Bedford: RSA Laboratories. < http://www.rsasecurity.com/rsalabs/node.asp?id=2152>

[21] *Fang Liu, Maiou Jose "Manny" Rivera, Xiuzhen Cheng. "Location aware Key Establishment in Wireless Sensor Networks", IWCMC'06,2006.*

[22] Atul Kahate, Cryptography and Network Security, TMH.

[23] William Stallings, "Network Security Essentials and Standards", Person Education, 2000.Cryptography and Network Security, TMH.