Storage-Optimal Key Sharing with Authentication in Sensor Networks

Jian Wang[1], Z.Y. Xia[1], Lein Harn[2], and Guihai Chen[3]

[1] Nanjing University of Aeronautics and Astronautics, China
Wangjian@nuaa.edu.cn
[2] University of Missouri, Kansas City, USA
[3] State Key Lab of Novel Software Technology, Nanjing University, China

Abstract. Wireless Sensor Networks are edging closer to widespread feasibility with recent research showing promising results in developing and adapting new mechanisms to suit their environment. Secure communication between these distributed wireless devices is a desired characteristic, especially in scenarios where these sensors will be exploited for military and other mission-critical operations. This paper highlights some of the research challenges for extending secure communications over these resource-constrained devices and points out why current protocols do not scale well in this unique application realm. Then, a storage-optimal key sharing scheme with authentication is proposed in this paper. This scheme requires each sensor node to hold one secret key and hash functions only, and is optimal in terms of memory consumption which is more rigorous constraint factor than computation and communication resources limit.

1 Introduction

Recent advances in electronic and computer technologies have paved the way for the proliferation of wireless sensor networks (WSNs). Sensor networks usually consist of a large number of ultra-small autonomous devices. Each device, called a sensor node, is battery powered and equipped with integrated sensors, data processing capabilities, and short-range radio communications. In typical application scenarios, sensor nodes are spread randomly over the terrain under scrutiny so as to collect sensing data. Examples of sensor network projects include SmartDust [1] and WINS.

Sensor networks are being deployed for a wide variety of applications [2], including military sensing and tracking, environment monitoring, patient monitoring and tracking, smart environments, etc. When sensor networks are deployed in a hostile environment, security becomes extremely important, as these networks are prone to different types of malicious attacks. For example, an adversary can easily listen to the traffic, impersonate one of the network nodes, or intentionally inject misleading information to other nodes, and even severe, make DoS attacks, e.g. exhausting the battery power by continuously broadcasting meaningless messages.Additionally, adversary can capture the nodes to get secret information stored in them, including secret keys. Due to the fact that communication among sensors is via wireless links,

G. Chen et al. (Eds.): ISPA Workshops 2005, LNCS 3759, pp. 466 – 474, 2005.

sensor networks are highly vulnerable to security attacks [3,4,5]. To provide security, communication should be encrypted and authenticated. The task of securing WSNs is an open research problem. A solution must strike a tradeoff between the security provided and the consumption of energy, computing and communication resources in the nodes.

Security is an important issue for wireless sensor networks. Similar to the wired networks, there are five main attributes of security for a wireless sensor network, i.e. confidentiality, integrity, authentication, non-repudiation and availability. Confidentiality ensures that information is secretly passed only to authorized members. Integrity is to guarantee that a message is transferred without being corrupted. Authentication is to enable a node to identify the peer node it is communicating with, and is thus a useful property to detect isolated or compromised nodes. Non-repudiation is to ensure that a node cannot deny having sent/received the message. Availability is to ensure that the network services survive despite denial of service attacks which could be invoked at any level of the sensor network, e.g., jamming the frequency to interfere with communication. Key management service could be certainly a target of attack of this kind too.

The key problem of securing WSNs is how to bootstrap secure communications between sensor nodes, i.e. to set up secret pair-wise keys between communicating nodes in the communication range. This problem is known as the *key agreement* or *key sharing* problem, which has been widely studied in general network environments. There are three types of general key agreement schemes: trusted-server schemes, public-key schemes, and key pre-distribution schemes. *Trusted-server* schemes depend on a trusted server for key agreement between nodes; an example is Kerberos. This type of scheme is not suitable for sensor networks because this kind of schemes can't scale well and this central server will also be an attacking target and thus a point of failure. *Public-key* schemes depend on asymmetric cryptography and require some sort of public-key infrastructure to be in place; an example of such schemes is an authenticated key agreement protocol using public-key certificates. However, as pointed out by Perrig, et al.[6], the limited computation and energy resources of sensor nodes often make it undesirable to use public-key algorithms in WSNs. A third way to establish keys pair-wise is via *Key Pre-Distribution* (KPD), where (secret) key information is distributed to all sensor nodes prior to deployment. Such schemes seem most appropriate for WSNs.

Sensor network is usually constrained in memory, computation and communication resources. Thus, key sharing approaches are always measured by overloads of memory, computation and communication. And, operations of communication are more consuming than that of computation. Therefore, most schemes by now have been focusing on reducing the round number of communication. In fact, communication is usually constrained by one-hop so that the overhead of communication is relatively moderate. Memory resources are more rigorous constrained factor, because all employed keys, programs, sensing data, and etc are compulsorily reserved altogether in the memory which is definitely only some hundreds of kilobyte in most cases. Thus, this paper intends to advance a storage-optimal authenticated key sharing scheme.

1.1 Our Main Contribution

1. Highlight definitely the contradiction between connectivity and resilience in KPD schemes. This shows the proposed KPD schemes improper to be applied in real scenario.
2. Propose a novel storage-optimal authenticated key sharing scheme. Each node need to hold a private key and two one-way hash functions only and additionally it can pair-wisely authenticate with other node.

1.2 Structure of This Paper

This paper is organized as following. Key pre-distribution is narrated and analyzed in section 2, and at the same time the contradiction between connectivity and robustness is highlighted there. Section 3 gives some unique characteristics of wireless sensor network and some assumptions employed later. We describe the proposed scheme, and analyze its security and performance in section 4 and 5 respectively. Section 6 discusses previously proposed related works. And finally, section 7 concludes this paper.

2 Key Pre-Distribution (KPD)

Up to now, there are two kinds of KPDs employed in the scenarios of sensor networks, i.e. deterministic KPDs [7,8] and random KPDs[9,10,11,12]. Deterministic KPDs can certainly guarantee the connectivity and the exclusive key pairing, but it requires each node keep more keys (usually $O\left(\sqrt{N}\right)$, N is the number of sensor nodes in a sensor network). On the contrary, random KPDs often emplys smaller key ring assigned to each sensor node compared with deterministic KPDs, but it only assures connectivity in high probability, and even worse, it makes the exclusive key pairing to be a difficult and exhaustive task.

In order to enhance the resistance against node capture in the some degree of connectivity, some types of multi-space schemes are proposed, e.g. Blom's key pre-distribution based schemes, secret-sharing based schemes and hash-chain based schemes. In fact, they gain higher resilience with the moderate cost of memory.

KPD is some sort of resource consumption and security tradeoff. And more public message transmitted which means more information it discloses, more prone to node-capture it is. For example, with disclosed information (e.g. key id), attacker can adaptively choose a node which holds the most unknown keys as next target. In order to avoid uncovering the key ids, only node ids are exchanged in some schemes (e.g. [11]), which necessitates some mapping from node id to key ids it holds so that nodes can know the key ids of other node with its id. The mapping undoubtedly expense extra memory, and if it is known by attacker, he then still is capable to choose the most valuable target.

In addition, the contradiction between connectivity and resilience exists in the random KPDs exists. High connectivity usually requires more keys assigned to each node. Attacker can get more keys upon success in compromising node each time if we assume the attacker is smart enough to choose the most valuable target to attack. In

another extreme, each node keeps few keys so that attacker is required to compromise more significant number of nodes before he can successfully destroy whole network. But, fewer key in each node means lower probability of connectivity.

3 Unique Characteristics of Wireless Sensor Network and Some Assumptions

WSNs had attracted much more attentions and been extensively studied recently. WSNs own some unique characteristics as followings

1. Power constraint. Usually, batteries which usually can not be recharged are utilized to provide power for sensor nodes after sensor networks are deployed.
2. Low volume of computation, communication, and memory capability. Sensor network is comprised of large quantities of sensor nodes, and thus sensor node should be in low cost. In a Wireless Sensor Network (WSN for short), individual sensor nodes, or sensors, are constrained in energy, computing, and communication capabilities. Typical sensor node contains a 4MHz CPU, 200kByte RAM, 64kByte ROM, and 19.2kbps wireless transmitter.
3. No topology infrastructure is known prior to deploy. A large amount of sensor nodes are randomly spread in a sensed region. Thus, it is difficult to determine a node's neighborhood before deployment. Typicallysensors are mass-produced anonymous commodity devices that are initially unaware of their location. Once deployed, sensors should self-organize into a network that works unattended.
4. Vulnerability. Due to the fact that individual sensor nodes are anonymous and that communication among sensors is via wireless links, sensor networks are highly vulnerable to security attacks [4,7,12,13]. Communications between sensor nodes is easily intercepted by adversary. Attacker can also intentionally modify or inject some false data into the message between sensor nodes to mislead the user who uses the sensor network. And furthermore sensor node is also vulnerable to node capture. Due to cost constraint, sensor node usually don't have secure hardware, such as tamper resistant part, thus secret information embedded in captured sensor node may be obtained by attacker. Moreover, adversary can launch a DoS (Denial of Sevices) attack by sending a lot of meaningless packet or jamming the frequency to interfere with communication.

In order to implement key sharing with authentication, we also give some assumptions as

1. No tamper-resistance. Due to cost limit, sensor node doesn't possess tamper-resistance capability, which means secret information embedded in captured sensor node may be obtained by attacker.
2. An attacker can intercept those communications in its transmission range only, but not whole communication.
3. Sensor node has the same range of wireless transmission.
4. A secure window exists in the phase of initial deployment. During such window, sensor node can securely transmit small volume of message in the plaintext, even if it is very short.

4 Proposed Scheme

4.1 Notations

H(x) One-way hash function
$H^i(x)$ Run *i* times of hash function
E(k,m) Secure encryption function which means to encrypt message *m* with key *k*.
$f(k_1,k_2)$ One-way hash function

4.2 Our Proposal

Our scheme is comprised of 3 phases sequentially, i.e. initialization pre-deployment phase, key sharing phase, and authentication phase, described briefly in the following.

Initialization pre-deployment phase
Each node is assigned a private key. And a one way hash function, *H*, is chosen and public. In addition, choose another one way hash function *f* which can guarantee $f\left(k_i, H^2\left(k_j\right)\right) = f\left(k_j, H^2\left(k_i\right)\right)$, $H^2(k)$ means invoking *2* times of *H* in *k*.

Key sharing phase
After deployment, each node firstly broadcast $H^2(k_i)$ (k_i is its private key) as its public key. And after receiving the public key of its neighbor (one hop), say *j*, node *i* can compute a shared key k_{ij} which is used to securely communicate with node *j* as $k_{ij} = f\left(k_i, H^2\left(k_j\right)\right)$. Node *j* can compute the sharing key k_{ij} in the same way.

Authentication phase
After computing k_{ij}, node *i* transmits authentication message $E\left(k_{ij}, H\left(k_i\right)\right)$ to node *j*. After receiving authentication message from *i*, node *j* can decrypt it with k_{ij} to get *h'*. Then, it can check if $H^2(k_i)=H(h')$. If so, it means $H^2(k_i)$ comes from node *i* that possesses k_i. Node *i* can check the authenticity of $H^2\left(k_j\right)$ in the same way.

4.3 Considering the Function *f* and *H*

Function, *H*, must be one-way. Otherwise, adversary can easily get the secret key k_i of a node with intercepting $H^2\left(k_i\right)$. And, function, *f*, must also be one-way. Otherwise, node *j* can easily know the secret key k_i of another neighbor node *i* one-hop away with known sharing key between them. In addition, they must meet the special property, i.e. $f\left(k_i, H^2\left(k_j\right)\right) = f\left(k_j, H^2\left(k_i\right)\right)$. Bilinear mapping can be employed to construct such functions, described as follows.

Suppose a *GF(p)* (*p* is a prime) finite field. Suppose G_1 is a cyclic group of *GF(p)* with primitive element *g* and order *p-1*. And, suppose $k \in G_1$, then $k = g^t \bmod p$

($t\neq 0$) and define $H(k) = k^m = g^{tm} \bmod p \in G_1$, here $m\neq 0$ is a predefined parameter.

Define f a bilinear mapping $f: G_1 \times G_1 \rightarrow G_2$ (G_2 is also a multiplicative group) as,

$$f(g_1, g_2^a) = f(g_1, g_2)^a = f(g_1^a, g_2), \quad g_1, g_2 \in G_1$$

If $k_i = g^{t_i} \bmod p \in G_1$ and $k_j = g^{t_j} \bmod p \in G_1$, then $H^2(k_i) = g^{2t_i m} \bmod p$ and $H^2(k_j) = g^{2t_j m} \bmod p$. We have

$$f(k_i, H^2(k_j)) = f(g^{t_i}, g^{2t_j m}) = f(g,g)^{2t_i t_j m}$$

$$f(k_j, H^2(k_i)) = f(g^{t_j}, g^{2t_i m}) = f(g,g)^{2t_i t_j m}$$

$$\therefore f(k_i, H^2(k_j)) = f(k_j, H^2(k_i))$$

But we know such Bilinear mapping with exponent is exhaustive to compute. If G_1 and G_2 are defined in elliptic curve, the overhead of computation may decrease some orders of magnitude with reserving similar security.

$$k_i = t_i \bullet g \in G_1, \quad k_j = t_j \bullet g \in G_1$$

$$H^2(k_i) = 2t_i m \bullet g, \quad H^2(k_j) = 2t_j m \bullet g$$

$$f(k_i, H^2(k_j)) = f(t_i \bullet g, 2t_j m \bullet g) = f(g,g)^{2t_i t_j m}$$

$$f(k_j, H^2(k_i)) = f(t_j \bullet g, 2t_i m \bullet g) = f(g,g)^{2t_i t_j m}$$

$$\therefore f(k_i, H^2(k_j)) = f(k_j, H^2(k_i))$$

Bilinear mapping based on elliptic curve is still computationally extensive relative to sensor network. Therefore, finding low-computation function f is still future work.

5 Security and Performance Analyses

5.1 Security Analysis

Secrecy of shared key

Due to the one-wayness of function H, attacker can not know k_j with knowing $H^2(k_j)$ transmitted from node j. Furthermore, attacker can not derive the sharing key $k_{ij} = f(k_i, H^2(k_j))$ without knowing the secret key k_i, due to the one-wayness of function f.

Resilience to node capture
Each node keeps one secret key and publishes its hash value of the key, and these secret keys are completely independent. Any node can impossibly know secret keys of nodes owing to one-wayness of function *H*. Thus, even a node is compromised and its secret key is known by attacker, secret keys of other nodes still keep secret. Furthermore, even attacker successfully captures many nodes, survival can still work securely. Our scheme can effectively deter the avalanche effect most proposed KPDs confront.

Resilience to Denial of Service (DoS) attacks
Our scheme can resist DoS attacks by authenticating each other. An attacker can record $H^2(k_j)$ legitimate node sent and then later broadcasts this fake value of $H^2(k_j)$ (in fact, he doesn't hold the corresponding secret key k_j), authenticating phase can find such attacker so as to avoid further-processing.

Resistance to impersonating
Attacker can probably impersonate as a fake node by producing a secret key *k* itself and sending $H^2(k_j)$ publicly. In order to mitigate such risk, it is a feasible way to tightly bind secret key of sensor node with sensor node id. For example, in the predeployment phase, the base station can firstly choose randomly secret key and then invoke the predetermined one-way hash function *g* (determined by the base station) on the 2-time hash value of the key to generate the corresponding node ID, and then assign a pair (k_i, id_i) to a node. During deployed, node can check each other if opponent is a fake node through validating if $g\left[H^2(k_i)\right] = id_i$.

5.2 Performance

In our scheme, each node needs only keep one secret key and two one-way hash functions *H* and *f* and possibly an extra one-way function to combine node id with its secret key, which is small and independent with network scale. And only two rounds of communication are needs for key sharing and authenticating, one for sharing and one for authenticating, which is still moderate and acceptable. And each node need invoking two times of one-way hash function *H* and one times of one-way hash function to produce a shared key between any two nodes, and invoking one symmetric encryption for authentication, which is reasonable.

Compared with deterministic KPDs ($O(\sqrt{n})$ storage requirement in each node, (*n* is the network scale)[8,9,10], our scheme outperforms in the storage requirement. Moreover, our scheme has advantageous over random KPDs [11,12,13] in that it can assure the connectivity with less storage.

In addition, our scheme doesn't need the exclusive testing, which is essential in many random KPD schemes and is an exhausted task. With reasonable resource-consumption, our scheme can guarantee both connectivity and resilience.

6 Related Works

If it is known which nodes will be in the same neighborhood prior to deployment, pair-wise keys can be established between these nodes (and only these nodes) *a priori*. However, most sensor network deployments are random; thus, such *a priori* knowledge about the topology of the network does not exist. A number of key pre-distribution schemes do not rely on prior knowledge of the network topology. A naive solution is to let all nodes store an identical *master* secret key. Any pair of nodes can use this master secret key to securely establish a new pair-wise key. However, this scheme does not exhibit desirable network resilience: if a single node is compromised, the security of the entire sensor network is compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to reduce the risk, but this increases the cost and energy consumption of each sensor. Furthermore, tamper-resistant hardware might not always be safe.

At the other extreme, one might consider a key pre-distribution scheme in which each sensor stores *N-1* keys, each of which is known to only one other sensor node (here, we let *N* denote the total number of nodes in the network). This scheme guarantees perfect resilience because any number of compromised nodes does not affect the security of any *un*compromised pairs of nodes. Unfortunately, this scheme is impractical for sensors with an extremely limited amount of memory because *N* could be large. Moreover, adding new nodes to a pre-existing sensor network is difficult when employing this scheme because the existing nodes do not have the new nodes' keys.

Recently, two key pre-distribution schemes suited for sensor networks have been proposed. Eschenauer and Gligor [13] proposed a random key pre-distribution scheme which may be summarized as follows: before deployment, each sensor node is assigned a random subset of keys from a large key pool; to agree on a key for communication, two nodes find a common key (if any) within their subsets and use that key as their shared secret key. Now, the existence of a shared key between a particular pair of nodes is not certain but is instead guaranteed only with some probability (which can be tuned by adjusting the parameters of the scheme).

Based on this scheme, Chan, Perrig, and Song [12] proposed a generalized "*q*-composite" scheme which improves the resilience of the network (for the same amount of key storage) and requires an attacker to compromise many more nodes in order to compromise any additional communication. The difference between this scheme and the previous scheme is that the *q*-composite scheme requires two nodes to find q (with $q > 1$) keys in common before deriving a shared key and establishing a secure communication link. It is shown that, by increasing the value of q, network resilience against node capture is improved for certain ranges of other parameters [12].

J. Wu and R. Wei [10] indicate the probabilistic method [9] can not yield practical CFF (Cover-free family) for key distribution. At the same time, among all known methods, this probabilistic construction provides the best sufficient conditions to ensure the existence of a CFF. That means based on the currently known methods, we can not construct or prove the existence of CFF that can provide satisfactory performance for ad hoc network key distribution.

7 Conclusion

Sensor network has been attracting much more concerns. Security is of sensor network very important component in many cases, especially when sensor network is employed in hostile environment. Key sharing, also known as bootstrapping, is a key component which is the base what other security functionalities will be implemented on. This paper surveys proposed Key Pre-distribution (KPD) schemes to be employed in the sensor network, and indicates the contradiction between connectivity and resilience to node-capture. A new scheme is proposed in this paper, which can guarantee both connectivity and resilience under the low consumption of memory, computation and communication. Further works include finding low-consuming function f which owns this special property, i.e. $f(k_i,H^2(k_j))=f(k_j,H^2(k_i))$. In addition, we will try to find some one-way function to map a secret key to corresponding node id so as to validate the authenticity of secret key.

References

1. Warneke M., Last B., Leibowitz, and K. Pister, SmartDust: communicating with a cubic-millimeter computer, *IEEE Computer*, vol. 34, no. 1, January 2001.
2. I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, Wireless sensor networks: A survey, *Computer Network*s, Elsevier Science, vol. 38, no.4, 2002.
3. D. Carman, P. Kruus, and B. Matt, "Constraints and Approaches for Distributed Sensor Networks Security," NAI Technical Report #00-010, Sep. 2000.
4. G. Jolly, M. Kuşçu, and P. Kokate, "A Hierarchical Key Management Method for Low-Energy Wireless Sensor Networks ", UMBC Online Document, Nov. 2002.
5. A. Wadaa, S. Olariu, L. Wilson, K. Jones, and Q. Xu, "On Training Wireless Sensor Networks," Proc. 3-rd International Workshop on Wireless, Mobile and Ad Hoc Networks, Nice, France, April 2003.
6. A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," Communications of the ACM, vol. 47, no. 6, pp. 53–57, June 2004.
7. M. Eltoweissy, H. Heydari, L. Morales, and H. Sudborough, "Combinatorial Optimization for Key Management in Secure Multicast Environments," Journal of Network and System Management, Kluwer Pobs, 2004.
8. S. A. Camtepe, B. Yener, Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks, ESORICS 2004: 293-308.
9. Aldar C-F. Chan, Distributed symmetric key management for mobile ad hoc networks, IEEE INFOCOM 2004.
10. J. Wu and R. Wei, Comments on "Distributed Symmetric Key Management for Mobile Ad hoc Networks" from INFOCOM 2004, eprint.iacr.org/2005/008.pdf
11. M. Ramkumar, N. Memon, HARPS-Hashed Random Preloaded Subset Key Distribution, Cryptology ePrint Archive, Report 2003/170, 2003, http://eprint.iacr.org/2003/170.
12. H. Chan, A. Perrig, and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks," Proceedings of IEEE 2003 Symposium on Security and Privacy, Berkeley, CA, May 2003.
13. L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks," Proceedings of the 9th ACM Conference on Computing and Communication Security, Nov 2002.