Random Key Pre-Distribution with Transitory Master Key for Wireless Sensor Networks

Filippo Gandino
Politecnico di Torino
Dipartimento di Automatica ed Informatica
Corso Duca degli Abruzzi, 24
10129 Torino, Italy
filippo.gandino@polito.it

Bartolomeo Montrucchio
Politecnico di Torino
Dipartimento di Automatica ed Informatica
Corso Duca degli Abruzzi, 24
10129 Torino, Italy
bartolomeo.montrucchio@polito.it

Maurizio Rebaudengo
Politecnico di Torino
Dipartimento di Automatica ed Informatica
Corso Duca degli Abruzzi, 24
10129 Torino, Italy
maurizio.rebaudengo@polito.it

ABSTRACT

Many Wireless Sensor Networks (WSNs) employ security schemes based on the symmetric encryption, which requires the establishment and management of cryptographic keys. This paper presents a new key management scheme based on Random Key Pre-Distribution. The novelty of the proposed approach is the introduction of a keyed transformation with a Transitory Master Key, which is exploited by the nodes in order to increase the number of different keys employed in the network. The presence of a larger pool of keys provides more robustness and connectivity than previously proposed Random Key Distribution schemes. Furthermore, random pre-distribution allows recovering secure communication also when the master key has been compromised.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection (e.g., firewalls)*

General Terms

Security

Keywords

Wireless Sensor Networks, Key Management

1. INTRODUCTION

Effective security systems are required in order to protect the privacy, the integrity, and the accuracy of Wireless Sensor Networks (WSNs). The main effective security protocols are based on symmetric cryptography, where two nodes share a common key which is used to encrypt and to decrypt the messages.

The application of a symmetric encryption scheme in the transmissions between two nodes requires that both the nodes know the employed key. The establishment of the cryptographic keys in the network is called Key Management [2]. This activity affects strongly the security and the computation performance of the network.

CoNEXT Student Workshop'09, December 1, 2009, Rome, Italy.
ACM 978-1-60558-751-6/09/12.

The two basic key management schemes employ only a global key, or a pair-wise key for each possible link in the network. With the former scheme, compromising a key, an opponent can eavesdrop the full network and pass any authenticity check. The latter is more robust, but it requires much more memory, since each node has to record a key for each other node.

An important family of protocols is based on Random Key Distribution, which consists in a random distribution of key material picked from a pool. Eschenauer and Gligor [1] proposed a simple random key pre-distribution scheme, where a large pool of p keys is generated off-line. A ring of r randomly chosen keys is then distributed to each node. After the deployment, each node verifies if some neighbors share a key with it. Random Key Distribution schemes can provide good robustness, but they decrease the connectivity of the network.

In order to reach a good security, without limiting the connectivity, several other key management protocols have been proposed [2, 3]. Some protocols reach better performance using deployment knowledge. However, their application is limited to networks where the deployment is known. Other protocols use a Transitory Master Key in order to generate the pairwise keys. In LEAP+ scheme [4] each couple of neighbor nodes can establish a key transforming their identifications with the master key. An adversary with the master key can eavesdrop all the links and inject new nodes, so the master key is deleted at the end of the initialization phase, in order to make harder compromising it.

We propose a new Random Key Distribution scheme with Transitory Master Key. As in a standard Random Key Distribution scheme, each node receives a ring of keys randomly picked from a pool. During the network initialization the nodes execute the keyed transformation on the keys, increasing their number. Preliminary analysis demonstrates that the larger number of keys does not affect the connectivity, and improves the security. With respect to LEAP+, we exploit random pre-distribution, allowing recovering secure communications also when the master key has been compromised.

The remaining of the paper is organized as follows: in Section 2 the characteristics of the proposed protocol are described. Section 3 analyzes the performance of the protocol. Finally, in Section 4 some conclusions are drawn.

2. KEY MANAGEMENT SCHEME

The proposed Key Management scheme adopts a *keyed transformation*, which is defined by assigning to each element of the domain a different element of the co-domain according to a key. The transformation is iterated on randomly pre-distributed keys a maximum number of times (μ).

A pool of keys, called *starting keys*, and a *master key* for the transformation are initially generated. Each starting key is matched to an identification code (ID).

When a node discovers a new neighbor, they perform the *shared-keys discovery* exchanging the IDs of their starting keys. When the node finds a shared starting key, it arranges with the neighbor a randomly chosen number of iterations between 0 and μ. Both the nodes iterate the transformation, using the master key, on the shared key. The resulting key will be used by those nodes during their future communications.

At the end of the initialization phase, each node performs a random number of iterations of the keyed transformation on each remaining starting key, and it deletes all the secret material apart from the final pairwise keys.

The goal of the scheme is to perform the shared-keys discovery with few keys, in order to increase the connectivity, and then increasing the number of keys used in the network, decreasing the number of nodes that can be compromised by an opponent.

If a node is compromised after the initialization phase, then all the keys owned by that node must be revoked. If a node is compromised during the initialization phase, then all the keys based on the starting keys owned by that node must be revoked.

New nodes can be added to the network, since they can discover shared-keys exchanging the IDs of the starting keys in their ring.

3. PERFORMANCE

The parameters that affect the performance of the network are the number of nodes (n), of keys in the pool (p), of keys in each ring (r), and of maximum iterations of the keyed transformation (μ).

When the ring is too large with respect to the pool, each node can communicate directly with the majority of its neighbors, but the capture of a node compromises a large part of the network. When the ring is too small with respect to the pool, a node could be not able to communicate with any neighbor. Moreover, a large pool decreases the number of links compromised by an adversary that takes only one key. Each compromised key allows establishing links with $\frac{rn}{p(\mu+1)}$ nodes, and each captured node provides r keys. Moreover, an adversary with a key can eavesdrop a link with probability equal to $\frac{1}{p(\mu+1)}$. When a node is compromised during the initialization phase an adversary can iterate the keyed transformation with the master key on the keys in the ring of that node, reaching $r(\mu+1)$ keys.

According to [1], the probability that the network is connected is

$$probability = 1 - \frac{(1-\frac{r}{p})^{2(p-r+\frac{1}{2})}}{(1-\frac{2r}{p})^{(p-2r+\frac{1}{2})}}$$

Observing the two schemes with the same values of r and p, we find that our approach is more robust, since $\mu > 0$. Moreover, the last formula is independent of μ, so with the same r and p, our approach provides the same connectivity as [1], where $\mu = 0$. When an adversary compromises the master key, the proposed protocol provides also the same robustness as [1].

The iteration of the keyed transformation requires additional computation, but the better ratio between robustness and connectivity allows a reduction of r and a lower memory area.

When a node in the initialization phase is compromised, the opponent can inject new nodes and eavesdrop several links. However, by revoking all the keys generated by the starting keys of the compromised node the connectivity is reduced, since $\frac{r}{p}$ links are stopped, but the network becomes again safe.

In LEAP+ [4], each compromised key allows establishing links with 2 nodes, and eavesdropping 1 link. Each captured node allows establishing links with all its neighbors. Therefore, it is more robust than the proposed protocol, because $\frac{rn}{p(\mu+1)} > 2$. However, when the node is compromised during the initialization phase, an adversary can generate all the keys, establishing links with any node, and eavesdropping any link. Therefore, our protocol is more robust against the capture of the master key, because $\frac{rn}{p} < n$. Furthermore, in LEAP+ when the master key is compromised the network security is unrecoverable.

4. CONCLUSIONS AND FUTURE WORKS

The adoption of a keyed transformation together with Random Key Distribution seems a valuable improvement, since it increases the robustness and the connectivity of the network with respect to Random Key Distribution schemes, and it enhances the robustness against master key capture. However, it involves a computational overhead. In order to accurately evaluate the performance of the proposed approach, future works will include precise calculation of the computational, communication, and memory overhead according to the reached connectivity and robustness.

5. ACKNOWLEDGMENTS

This work has been partially supported by the grant "Piattaforma Tecnologica Innovativa per l'Internet of Things" from Regione Piemonte.

6. REFERENCES

[1] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002.

[2] J. Lee, V. Leung, K. Wong, J. Cao, and H. Chan. Key management issues in wireless sensor networks: current proposals and future developments. *Wireless Communications, IEEE*, 14(5):76–84, October 2007.

[3] Y. Zhou, Y. Fang, and Y. Zhang. Securing wireless sensor networks: a survey. *Communications Surveys & Tutorials, IEEE*, 10(3):6–28, Quarter 2008.

[4] S. Zhu, S. Setia, and S. Jajodia. Leap+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Trans. Sen. Netw.*, 2(4):500–528, 2006.