10-2008

Using Trusted Computing Technology to Facilitate Security Enforcement in Wireless Sensor Networks

Yanjiang YANG
Singapore Management University, yjyang@smu.edu.sg

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Feng BAO
Singapore Management University, fbao@smu.edu.sg

Jianying ZHOU
Singapore Management University, jyzhou@smu.edu.sg

DOI: https://doi.org/10.1109/APTC.2008.13

Citation

YANG, Yanjiang; DENG, Robert H.; BAO, Feng; and ZHOU, Jianying. Using Trusted Computing Technology to Facilitate Security Enforcement in Wireless Sensor Networks. (2008). *APTC 2008: Proceedings of the 3rd Asia-Pacific Trusted Infrastructure Technologies Conference: 14-17 October, Wuhan*. 43-52. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/439

Using Trusted Computing Technology to Facilitate Security Enforcement in Wireless Sensor Networks

Yanjiang Yang
Institute for Infocomm Research
Singapore 138632
yyang@i2r.a-star.edu.sg

Robert H. Deng
Singapore Management University
Singapore 178902
robertdeng@smu.edu.sg

Feng Bao, Jianying Zhou
Institute for Infocomm Research
Singapore 138632
{baofeng,jyzhou}@i2r.a-star.edu.sg

Abstract

Security enforcement in wireless sensor networks is by no means an easy task, due to the inherent resource-constrained nature of sensor nodes. To facilitate security enforcement, we propose to incorporate more powerful high-end Security Enforcement Facilitators (SEFs) into wireless sensor networks. In particular, the SEFs are equipped with TCG-compliant Trusted Platform Modules (TPMs) to protect cryptographic secrets, perform authenticated booting and attest their platform state to a remote base station. As such, the SEFs act as online trusted third parties to effectively monitor the states of sensor nodes, help in key management, simplify secure routing, and facilitate access control.

1 Introduction

Wireless sensor networks (WSNs) are an emerging technology with a wide range of applications, such as battlefield surveillance, healthcare monitoring, wildlife tracking, emergency response, and disaster recovery. Sensor nodes in WSNs are extremely constrained in hardware due to requirements such as low cost and maximizing lifetime. As a result, sensor nodes normally have very limited computation capability, storage capacity, power supply, and radio transmission range. For example, a typical Smart Dust [24] sensor is configured with a 4MHz 8-bit CPU, 8Kb program RAM and 0.5KB data memory.

When deployed in mission-critical applications, securing WSNs is compulsory. Unfortunately, the resource-constrained nature of sensor nodes makes security enforcement in WSNs a challenging task. Recently lots of efforts have been spent in exploring security enforcement in WSNs, and proposals on various security services and mechanisms have been proposed, ranging from intrusion detection [2, 16, 23, 30, 43], secure routing [4, 17, 18, 25, 35], key management and establishment [7, 9, 10, 12, 26, 27, 28, 29, 37, 44], access control [41, 45], to data aggregation [8, 15, 31, 36]. Most of these efforts are on homogeneous WSNs where all sensor nodes have the same capabilities. However, homogeneous WSNs are not scalable. Indeed, both theoretical and empirical studies have found out that the larger number of sensor nodes adversely affects the throughput of individual nodes to a substantial extent, and that as the traffic becomes heavy, the control overhead due to the underlying routing protocols alone will consume a large portion of the available bandwidth [11, 14].

To improve security enforcement and performance, in this paper we propose a heterogeneous WSN which comprises of the resource constraint sensor nodes as well as more powerful high-end Security Enforcement Facilitators (SEFs). Compared to a sensor node, a SEF has higher computation capability, larger storage size, longer power supply, and longer radio transmission range, and it thus does not suffer from the resource scarceness problem as much as a sensor node does. More importantly, a SEF is equipped with trusted computing technologies, and in particular a TCG-compliant TPM (Trusted Platform Module) [39] is attached to each SEF. According the TCG specifications, the TPM is a tamper-resistant, self-contained secure coprocessor attached to a computing platform, capable of performing cryptographic functions. A TPM enabled trusted computing platform provides *sealed storage* for storing secret data, measures and reports the integrity state of the platform. As a result, the TPM functions as a hardware based root of trust and has a set of mechanisms to propagate trust across multiple platforms. More discussions on TCG/TPM can be found in Section 2. SEFs are special type of TPM-enabled trusted computing platforms which act as online *trusted parties* within a WSN; therefore security enforcement is expected to be enormously simplified and improved. In this paper, we focus on illustrating how SEFs help to facilitate enforcing security mechanisms in WSNs, thereby enhancing the efficacy of security enforcement.

The remainder of the paper is organized as follows. In section 2, we give a brief overview of the TPM as preliminary knowledge. We then discuss in detail the SEF-enabled WSN architecture in Section 3. We illustrate how the introduction of SEFs into WSNs can facilitate security enforcement by examples on various security services in Section 4. We review related work in Section 5, and Section 6 concludes the paper.

2 Preliminaries: Overview of TPM

The latest effort in trusted computing is represented by the Trusted Computing Platform specifications defined by Trusted Computing Group (TCG) [39]. The specifications aim to provide hardware based root of trust through a tamper resistant coprocessor, Trusted Platform Module (TPM). The TPM is attached to a host machine, and acts as the *root of trust* of the host platform given its tamper resistance feature. The TPM is designed to be capable of performing cryptographic functions such as random number generation, SHA1 hash function, and RSA encryption and digital signature.

A core functionality provided by the TPM is integrity measuring and storage, and reporting of the host platform. The integrity measuring and storage are achieved through a set of Platform Configuration Registers (PCRs), internal to the TPM. Each PCR value is a 20-byte cumulative hash digest (SHA1 value) of a number of measured platform integrity metrics. Altogether the PCRs record the integrity status of the host platform from booting to OS loading to applications loading. A update to a PCR value is through what is termed *extending the PCR*, which is described as

$$PCR[i] \leftarrow SHA1(PCR[i]||newlymeasuredvalue) \tag{1}$$

where i is the number of the PCR being updated. Since a PRC value is a digest of the platform state (which results from a series of state altering events), it is meaningless by itself. The data that complements PCRs in providing semantics is Stored Measurement Log (SML). The SML stores the complete event history for all the PCRs, and each PCR has corresponding entries in the SML that records the series of events leading to the current PCR value . The SML is stored unprotected outside the TPM. This however does not compromise integrity as the corresponding digests are stored in PCRs, and "extending a PCR" can only be performed by TPM protected capabilities. The PCR values, together with the corresponding entries of the SML, are used as evidence to attest to the current platform state of the host.

Upon request, the TPM can report the state of its underlying platform to a remote challenging entity through *attestation*. In particular, the TPM has a number of key pairs called Attestation Identity Keys (AIKs), which are used as aliases of the unique Endorsement Key (EK). The attestation protocol proceeds as follows. (1) The challenging entity issues a challenge message, indicating that it wants to inspect one or more PCR values. (2) A Platform Agent collects the related SML entries corresponding to the requested PCR values. (3) TPM sends the Platform Agent the requested PCR values signed by the private key of an AIK. (4) The Platform Agent sends the signed PCR values, together with the relevant SML entries and the related credentials such as the certificates to the challenging entity. (5) The challenging entity verifies the replied data: the measurement digest is computed from the SML entries and compared with the signed PCR values; the credential of AIK is validated.

Another security function provided by the TPM is *Sealed Storage*, which encrypts sensitive data with integrity measurement values. In particular, the sensitive data together with one or more PCR values are encrypted/sealed. Subsequently, the TPM releases an encrypted data only if the current PCR values match those stored during en-

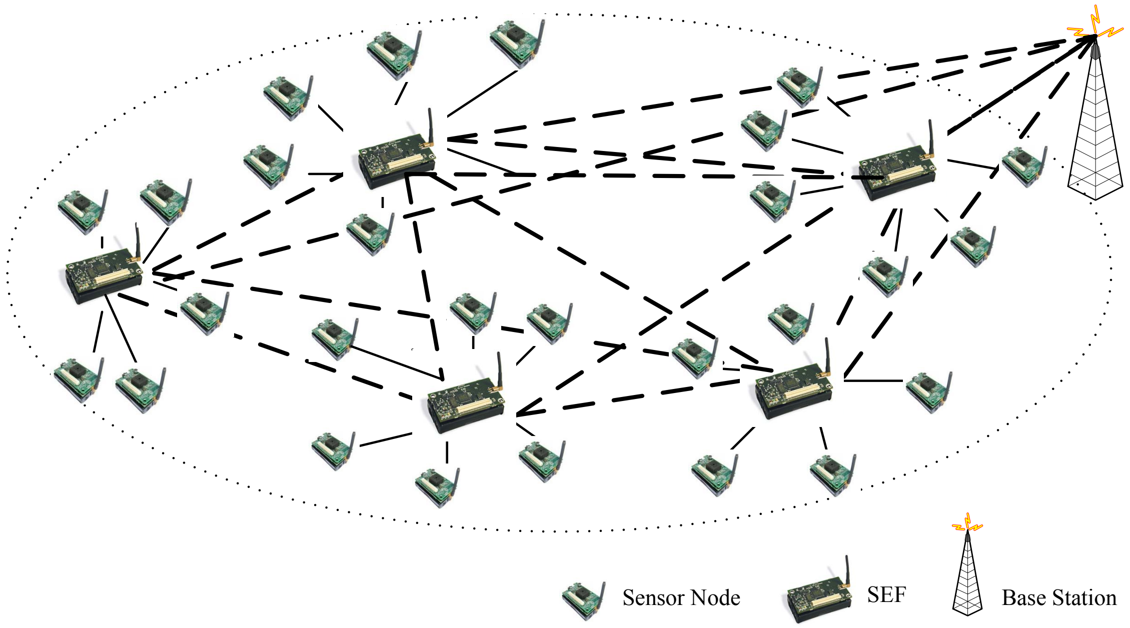

Figure 1. Heterogeneous SEF-enabled Wireless Sensor Network

cryption. In other words, if the state of a platform is modified, the encrypted data in the sealed storage under that state will not be decrypted/unsealed. The encryption key is protected either by the Storage Root Key (SRK) internal to the TPM, or by a key protected by the SRK.

3 Incorporating SEFs in Wireless Sensor Networks

3.1 SEF-enabled Architecture

We partition a WSN into a number of *clusters*. A SEF is placed into each cluster, acting as the *cluster head*. In contrast to the sensor nodes, the SEFs have relatively higher computation capability, larger storage size, and longer radio transmission range. They also have longer power supply, and in some circumstances, the SEFs can even be line-powered, e.g., in the case that the wireless sensor network is established to monitor a physical facility, it will be easy to tap on the electricity lines to supply power to the SEFs. Therefore unlike the sensor nodes, the SEFs do not have the resource scarceness problem. The introduction of the high-end SEFs into a WSN actually makes the once homogeneous network *heterogeneous*, as shown in Figure 1. Within a cluster, the SEF acts as a gateway, and a sensor node can reach the SEF directly, or via *few* other sensor nodes. A fundamental criteria for cluster partition is that the data traversal path from the originating node to the SEF must be short (i.e., the number of intermediary nodes is small). Inter-cluster communication or communication between a sensor node and the base station is through the respective cluster heads. Since SEFs are not constrained by resources and power supply, communication at the level of SEFs and the base station does not suffer from the limitations as sensor nodes do, and it can utilize more advanced communication infrastructure, e.g. 802.11 or even wired network. As a result, SEFs-enabled architecture can be expected to enormously improves the overall system performance and the lifetime of the network.

More importantly, we equip each SEF with a TPM, so as to facilitate and simplify security enforcement in the SEF-enabled WSNs. The rationale is that under the auspices of TPM, SEFs can act as online *trusted parties* on behalf of the base station[1] in enforcing security mechanisms. The base station ascertains the trustfulness of SEFs by means of attestation (see Section 2). It is important to note that sensor nodes do not challenge the SEFs for attestation (in fact they can not afford to do so), and they simply trust the SEFs. It is the sole responsibility of the base station to check

[1] Without loss of generality, we assume that the base station is the control center responsible for implementing security mechanisms.

the status of the SEFs. The base station can either periodically challenge all the SEFs on fixed intervals, or choose some SEFs for attestation at random. Misbehavior of the SEFs can be readily detected by the base station in the course of attestation. Without detecting compromises, the base station can entrust the task of security enforcement to individual SEFs.

Our SEF-enabled architecture has a number of advantages: (1) As discussed earlier, the SEFs help to reduce the amount of data that must traverse the network, thereby enhancing the over system throughput. (2) Partitioning a network into clusters makes management of the network scalable. Incorporating SEFs simplifies management of sensor nodes, as the base station can delegate the management and adminstration of the sensor nodes to the respective cluster heads. For example, the SEF is aware of the structure of the cluster where it resides, thus it takes charge of routing information within the cluster. The sensor nodes no longer need to discover routing paths by themselves, which is an energy-consuming process; they simply depend on the SEF for data transmission. (3) The SEF-enabled network allows for easy node dynamics such as node join and node departure. The reason is that node dynamics within a cluster are managed by the corresponding cluster head, which is closer to the sensor nodes. (4) The SEFs amortize the workload of the base station in security enforcement, preventing the base station from becoming the system bottleneck. Further, SEFs avoid single point of vulnerability at the base station. (5) The SEFs can work together to provide better resilient security mechanisms, using for example threshold cryptographic techniques.

3.2 SEF Configuration

Depending on applications, hardware capabilities of a SEF may vary from that comparable to a bluetooth device to that of a high end PDA. The TCG is currently working on the specifications for Trusted Mobile Platforms, whose core element is a Mobile Trusted Module (MTM), similar to the TPM for PCs [40]. Attempts for prototype implementation of MTM were also already available (e.g., [38]). Hence, there exists no technical barrier to implement TPM/MTM-enabled SEFs.

A trusted computing platform can be implemented as a restricted system or an open system. The former runs a small set of protected applications, while the latter runs both protected and unprotected applications. We chose to design the SEF as a restricted trusted computing platform due to its specialized functionality and application in WSNs. A reference platform configuration of SEF is shown in Figure 2. The platform runs the sole SEF application. At the application layer, the SEF program includes four main components. The *trusted computing agent* (or TC agent) is the interface that accesses the functionalities provided by the underlying TPM/MTM such as sealed storage and integrity reporting mechanism. The *security module* is dedicated to implementing the designated WSN security mechanisms. The *sensor agent* is the communication interface with sensor nodes, while the *base station agent* is the interface with the base station or other SEFs. The OS layer implements the SEF core, bridging between the application layer and the hardware layer. The hardware layer includes a TCG-compliant TPM/MTM, providing hardware based root of trust.

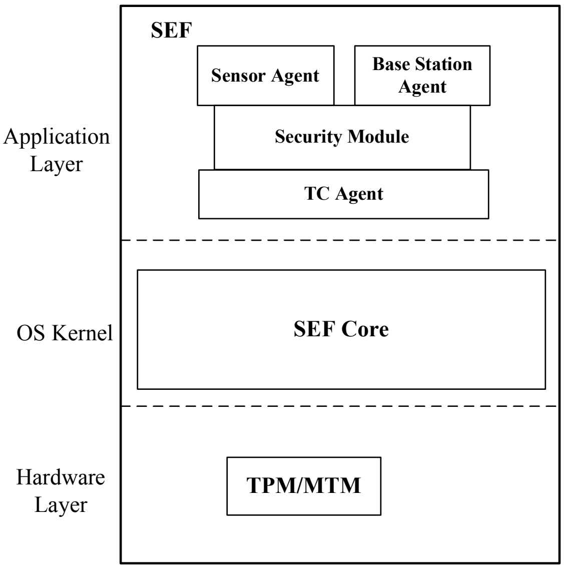

Figure 2. A Reference SEF Platform Configuration

Finally, we stress that platform attestation in our case allows the SEF to attest to the base station the platform state from booting, to loading of the SEF core, and to loading of the SEF program. Referring to the general attestation protocol reviewed in Section 2, the concrete attestation process between the SEF platform and the base station can be briefly outlined as follows: the base station sends the challenging message to the base station agent, who then invokes the TC agent to obtain the requested PCR values signed by the

TPM/MTM. The base station agent collects the pertinent SML entries and credentials, and returns these data, along with the signed PCR values, to the base station, who then carries out the necessary verification process.

4 How SEFs Facilitate Security Enforcement

In this section, we illustrate how SEFs help to enforce security mechanisms efficiently in WSNs through examples of intrusion detection, security routing, key management and establishment, access control, and secure data aggregation, respectively.

4.1 Intrusion Detection/Detection of Node Compromises

Wireless sensor networks are often deployed in hostile environments. The unattended sensor nodes thus easily fall victim to attacks whereby an adversary physically invades the sensor nodes. Since sensor nodes are simple in design and cannot afford tamper resistance, an adversary who captures a sensor node can easily extract secrete information from them. As physical safety is fundamental to all security services, it is crucial to timely detect compromises of sensor nodes.

An annoying fact is that the well-established intrusion detection techniques for high end computers or networks are in general not applicable to WSNs. A WSN entails tailored intrusion detection solutions. Since a sensor node by itself does not afford implementing intrusion detection mechanism, existing solutions for general Ad Hoc wireless networks or sensor networks such as [2, 16, 23, 30, 43] often enlist cooperation among multiple nodes to detect intrusion, and in particular, each node is responsible for watching the activities of its neighboring nodes. Cooperative intrusion detection of this kind is quite expensive in both computation and power consumption in a WSN, since sensor nodes are required to be on alert all the time. Worse yet, collusion of sensor nodes is even harder to detect, since each node only has knowledge of its *local situation*.

In contrast, intrusion detection mechanism in a SEF-enabled WSN can be readily performed at two levels, the cluster level and the network level. At the cluster level, each SEF monitors the sensor nodes within the cluster. Since all sensor nodes communicate with their designated SEF, intrusion detection enforced by the SEF is bound to be effective. Detection of collusion among sensor nodes is also possible, as the SEF monitors the behaviors of all sensors in the cluster. Instead of invading a sensor node, an adversary may try to deploy spurious sensor nodes in a cluster in order to generate bogus traffic. This kind of attacks can also be readily addressed, with the SEF in place to detect the presence of illegitimate nodes through key management and access control.

At the network wide level, SEFs and the base station form the second layer of defense. SEFs can watch each other's behaviors to detect intrusion to SEFs. The base station can also check the state of SEFs by performing remote attestation periodically. Compromising a SEF via hardware attack is much harder to succeed than hardware attack to a sensor node, since the former is equipped with a tamper-resistant TPM while the latter is not.

By and large, the introduction of SEFs into a WSN helps to alleviate one of the fundamental threats to WSNs, i.e., compromises of sensor nodes, thereby laying a solid foundation to implement other security mechanisms and services.

4.2 Secure Routing

Secure routing ensures secure data transmission to the target node. In particular, the need for secure routing stems from the fact that data transmitted from one sensor node to another node (or to the base station) requires many intermediary nodes for routing. This leaves the chances for the routing nodes enroute to alter the data to be delivered. Existing proposals to secure routing, e.g., [4, 17, 18, 25, 35], rely on cooperation among sensor nodes across the network. As the number of sensor nodes increases in a WSN, the average routing path becomes longer. This results in two adversary effects. First, the control overhead due to routing protocols would consume most of the available bandwidth [11], and second, the throughput of each node deteriorates rapidly as the number of sensor nodes becomes large [14].

In a SEF-enabled WSN, however, data transmission is carried out in a hierarchical manner. Within a cluster, all traffic from sensor nodes are sent to their SEF either directly or via a short traversal path. SEFs then deliver traffic directly to the base station. This greatly reduces routing protocol overhead and hence significantly improves overall network throughput.

4.3 Key Management and Establishment

Public key cryptosystems are in general too expensive for WSNs, so symmetric key primitives such as secret key encryption or cryptographic hash function are often preferred in implementing security mechanisms such as authentication and data secrecy. As such, the issue of key management and establishment in WSNs boils down to sharing of secret keys between two sensor nodes (or between the base station and sensor nodes). To establish shared keys among sensor nodes, a commonly accepted approach is to pre-load a set of key values inside sensor nodes before their deployment. These pre-loaded keys are later used either directly as shared keys for data encryption and authentication, e.g, [7, 10, 12, 26, 27, 28, 37, 44], or as a basis to establish new keys, e.g., [9, 29]. Either way, the introduction of SEFs into a WSN considerably simplifies key management.

Sensor nodes in a WSN either use a common group key [9, 28, 29] or pair-wise keys [7, 10, 12, 26, 27, 37, 44] to achieve security. In the former case, all sensor nodes in the WSN share a common key which is used to protect control messages. Sharing a common key on a large scale is apparently risky, since compromise of a single node would jeopardize the security of the entire network. The SEF-enabled architecture assuages this situation by partitioning a WSN into smaller clusters. Each cluster shares a distinct group key, instead of sharing a network wide group key. The effect of compromise of a cluster key is therefore limited to the cluster only, with no adverse effect on the other clusters. The SEFs can also assist a common group key-based scheme with two more advantages. First, in a WSN without SEFs, the establishment and update of a common key is normally dependent on the base station to broadcast key management messages to all sensor nodes. Performance of broadcast key management schemes deteriorates rapidly as the group size increases. This is especially true for WSNs where the wireless channels subject to both random as well as burst errors and key broadcast will suffer from the so called ”ACK explosion” problem. In a SEF-enabled WSN, key broadcast is confined to individual clusters and hence effectively avoids most of the pitfalls associated with key broadcast in large scale networks. Second, SEFs enable detection of compromised sensor nodes in a more timely manner than in a WSN without SEFs.

Pair-wise key sharing is required to protect communications between two communicating sensor nodes. In a SEF-enabled network, the SEFs serve as on-line trusted key distribution centers for their respective clusters. Key establishment via trusted key distribution center is a matured research area where many highly efficient protocols have been proposed in the literature.

Key management and establishment is a security bootstrapping service, without which many other security mechanisms are impossible. Introducing SEFs into a WSN make key management and establishment much more simple and efficient and hence more practical.

4.4 Access Control

Access control in a WSN is used to restrict outside entities from access sensor nodes. In a plain WSN without SEFs, access control is enforced by the base station which regulates access requests from the outside entities. The base station first determines the validity of the requesting entity, and if it’s valid, passes the key used by the target node to the requesting entity. SEFs add flexibility to access control in a SEF-enabled WSN: each SEF controls accesses to the nodes within the cluster it oversees. Alternatively, the SEFs can collaborate to enforce threshold based access control, which has greater resilience.

We note that with the SEFs in place to enforce access control, it is possible to readily address DoS attacks, whereby one more adversaries intentionally send bogus messages or replay old messages to a WSN in order to consume the power of the sensor nodes and bandwidth of the network. First of all, SEFs afford to implement relatively sophisticated countermeasures to DoS attacks. On top of that, with all outside communication passing through SEFs, the DoS messages can be effectively blocked from reaching the sensor nodes.

4.5 Secure Data Aggregation

It is well known that data transmission consumes more power than data processing for a sensor node. To reduce the amount of data traversed across a network, data aggregation has been proposed in certain applications wherein statistical values of data from multiple sensor nodes rather than individual sensor data are delivered. Secure aggregation, e.g., [8, 15, 31, 36], ensures that data are correctly aggregated en route to the destination point (e.g., the base station). The basic approach for secure data aggregation in WSNs is that some sensor nodes are selected, acting as aggregators, each of which is responsible for aggregating data from its neighboring nodes; the aggregators then forward the intermediate aggregated values to the destination point where the final aggregated value is calculated. Of course security measures must be in place upon the aggregator nodes to ensure that they act honestly. In a SEF-enabled WSN, SEFs are natural candidates for aggregators, aggregating data from their respective clusters. Safe guarding the correct operation of a SEF is much more straightforward than safe guarding that of a sensor node, by virtue of the TPM attached to the SEF.

5 Related Work

Clustering a wireless sensor network into separate clusters were proposed by several authors for achieving scalability and better performance (see for example [1, 3, 6, 13, 21]). Normally, some nodes within a cluster are chosen as cluster head. SEFs in our proposal provide a natural way to materialize the concept of cluster head. Network clustering should be common practice in managing large WSNs.

Given the limited capabilities of sensor nodes, several studies from both the research community and the industry sector have tried to enhance network performance by incorporating a number of more powerful nodes in WSNs. A detailed theoretical analysis on the effect of adding powerful nodes to WSNs was given in [42]. It is concluded that only a modest number of reliable, long-range backhaul links and line-powered nodes are required to have a significant effect, and if properly deployed, heterogeneity can triple the average delivery rate and a 5-fold increase in the lifetime of a large battery-powered sensor networks. Intel has an on-going experimental effort [20] to incorporate Intel XScale® based nodes into WSNs. The experiment indicated that data traversing across a network are routed biased towards the XScale® nodes over simple sensor nodes, thereby indeed enhancing the overall system performance.

Our study of SEF-enabled WSNs is motivated by the above idea of networking clustering and TCG's trusted computing platform. Our SEF-enabled heterogeneous WSN differs from the Intel heterogeneous networks [20] in several aspects. The former is SEF-aware such that all nodes within a cluster is aware of the existence of the SEF, while the latter intends to achieve non high-end node awareness. Sensor nodes in the Intel networks need to dynamically discover routing pathes, while the sensor nodes in the SEF-enabled networks are designed to get routing information from the corresponding SEF. As a result, WSNs empowered by SEFs are expected to have lower control overhead and higher throughput than the Intel heterogeneous networks. Moreover, our SEF-enabled WSNs not only improves network performance but also greatly facilitate security enforcement, as we have demonstrated in the paper.

The introduction of the TPM-equipped SEFs into wireless sensor networks represent yet another application of trusted computing technology in network applications. Trusted computing has already been exploited in wireless networks where wireless devices having moderate capabilities, in which case the TCG-compliant TPMs can be directly embedded in the devices, e.g., [5, 33, 34]. In contrast, the sensor nodes in a WSN we are considering by themselves cannot house TPMs. Trusted computing has also been proposed to protect privacy in RFID systems (the RFID tags forming a RFID system are also quite weak in capabilities), where a TPM is attached to the RFID reader [32] in order to make the reader act according to the established privacy policies. Another work to protect RFID privacy is [22] which proposes to incorporate powerful devices into a RFID system, and the devices are designed to simulate the behavior of the RFID tags and the tags themselves are made dormant. The devices are simply assumed to be trusted, but how to make the assumption practically true is unclear. Anyway, a RFID system has quite different working mechanisms from a general WSN, thus the security concerns, and in turn the challenges in enforcing security, in the two are significantly different.

6 Conclusion and Future Work

Sensor nodes are severely constrained by their scarce resources in terms of computation capability, storage capacity, power supply, and radio range. Security enforcement is thus extremely challenging in wireless sensor networks. To solve this problem, we proposed to render a wireless sensor network heterogeneous, by incorporating TPM-equipped SEFs into clusters of the network. We gave various examples, to illustrate how SEFs help to facilitate and simplify security enforcement in WSNs.

This study is still in the preliminary stage. Our future work will first focus on the design and implementation of proof-of-the-concept TPM equipped SEFs. We will then study and experiment of incorporating SEFs to real world wireless sensor networks, e.g., the sensor networks deployed to oversee an energy distribution system.

References

[1] M. Aboelaze, and F. Aloul. *Current and Future Trends in Sensor Networks: A Survey*, Proc. IFIP International Conference on Wireless and Optical Communications Networks, WOCN'05, pp. 551-555, 2005.

[2] S. Buchegger, and J. Le Boudec. *Performance Analysis of the CONFIDANT Protocol Cooperation of Nodes - Fairness in Dynamic Ad-hoc Networks*, Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'02, pp. 226-236, 2002.

[3] S. Banerjee, and S. Khuller. *A Clustering Scheme for Hierarchical Control in Multihop Wireless Networks*, Proc. IEEE INFOCOM'01.

[4] P. Bryan, M. Luk, E. Gaustad, and A. Perrig. *Secure Sensor Network Routing: A Clean-Slate Approach*, Proc. 2nd Conference on Future Networking Technologies, CoNEXT'06, 2006.

[5] S. Balfe, A. D. Lakhani, and K. G. Paterson. *Trusted Computing: Providing Security for Peer-to-Peer Networks*, Proc. IEEE International Conference on Peer-to-Peer Computing, P2P'05, 2005.

[6] S. Basagni. *Distributed Clustering Algorithm for Ad-Hoc Networks*, Proc. International Symposium on Parallel Architectures, Algorithms, and Networks, 1999.

[7] H. Chan, A. Perrig, and D. Song. *Random Key Pre-distribution Schemes for Sensor Networks*, Proc. IEEE Symposium on Security and Privacy, pp. 197-213, 2003.

[8] H. Chan, A. Perrig, and D. Song. *Secure Hierarchical In-Network Aggregation in Sensor Networks*, Proc. ACM Conference on Computer and Communications Security, CCS'06, 2006.

[9] R. Dutta, E. C. Change, and S. Mukhopadhyay. *Efficient Self-healing Key Distribution with Revocation for Wireless Sensor Networks Using One Way Key Chains*, Proc. Applied Cryptography and Netwrok Security, ACNS'07, pp. 385-400, 2007.

[10] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. *A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks*, Proc. ACM Conference on Computer and Communication Security, CCS'03, pp. 42-51, 2003.

[11] S. Das, C. Perkins and E. Royer. *Performance Comparison of Two On-demand Routing Procotols for Ad Hoc Networks*, Proceedings of Infocom 2000, Vol 1, pp. 3-12, IEEE Press, 2000.

[12] L. Eschenauer, and V. D. Gligor. *A Key-Management Scheme for Distributed Sensor Networks*, Proc. ACM Conference on Computer and Communication Security, CCS'02.

[13] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. *Next Century Challenges: Scalable Coordination in Sensor Networks*, Proc. ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'99.

[14] P. Gupta and P.Kumar. *The Capacity of Wireless Networks*, IEEE Transactions on Information Theory, Vol 46(2), pp. 388-404, 2000.

[15] L. Hu, and D. Evans. *Secure Aggregation for Wireless Networks*, Proc. 2003 Symposium on Applications and the Internet Workshops, SAINT'03, pp. 384-394, 2003.

[16] Y. Huang, W. Fan, and W. Lee. *Cross-feature Analysis for Detecting Ad-hoc Routing Anomalies*, Proc. 23rd International Conference on Distributed Computing System, 2003.

[17] Y. Hu, D. Johnson, and A. Perrig. *SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks*, Ad Hoc Networks Journal, Vol. 1(1), pp. 175-192, 2003.

[18] Y. Hu, A. Perrig, and D. Johnson. *Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks*, Wireless Networks Journal, Vol. 11(1), 2005.

[19] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. *Impact of Network Density on Data Aggregation in Wireless Sensor Networks*, Proc. 1st International Conference on Distributed Computing Systems, 2001.

[20] http://www.intel.com/research/exploratory/heterogeneous.htm.

[21] N. Jain, and D. P. Agrawal. *Current Trends in Wireless Sensor Network Design*, International Journal of Distributed Sensor Networks, Vol 1(1), pp. 101-122, 2005.

[22] A. Juels, P. Syverson, and D. Bailey. *High-Power Proxies for Enhancing RFID Privacy and Utility*, Proc. Privacy Enhancing Technologies (PET) Workshop, pp. 210-226. 2005.

[23] O. Kachirski, and R. Guha. *Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks*, Proc. 36th Annual Hawaii International Conference on System Sciences, HICSS'03, 2003.

[24] J. M. Kahn, R. H. Katz, and K. S. J. Pister. *Next Century Challenges: Mobile Networking for "smart dust"*, Proc. International Conference on Mobile Computing and Network, MOBICOM'99, pp. 271-278, 1999.

[25] C. Karlof, and D. Wagner. *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasurements*, Proc. 1st IEEE International Workshop on Sensor Network Protocols and Applications, 2003.

[26] D. Liu, and P. Ning. *Location-based Pairwise Key Establishement for Relatively Static Sensor Networks*, Proc. ACM Workshop on Security of Ad hoc and Sensor Networks, 2003.

[27] D. Liu, and P. Ning. *Improving Key Predistribution wih Deployment Knowledge in Static Sensor Networks*, ACM Transactions on Sensor Networks, 2005.

[28] D. Liu, P. Ning, and W. Du. *Group-based Key Pre-distribution in Wireless Sensor Networks*, Proc. ACM Workshop on Wireless Security, 2005.

[29] D. Liu, P. Ning, and K. Sun. *Efficient Self-Healing Group Key Distribution with revocation Capability*, Proc. ACM Conference on Computer and Communication Security, CCS'03, 2003.

[30] N. Marchang, and R. Datta. *Collaborative Techniques for Intrusion Detection in Mobile Ad-hoc Networks*, Ad Hoc Network, Vol. 6, pp. 508-523, Elsvier, 2008.

[31] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. *TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks*, Proc. 5th Annul Symposium on Operating Systems Design and Implementation, OSDI'02, 2002.

[32] D. Molnar, A. Soppera, and D. Wagner. *Privacy for RFID Through Trusted Computing*, Proc. ACM workshop on Privacy in the Electronic Society, WPES'05, pp. 31-34, 2005.

[33] W. Mao, F. Yan, and C. Chen. *Daonity - Grid Security with Behaviour Conformaity from Trusted Computing*, Proc. ACM Workshop on Scalable Trusted Computing, pp. 43-46, 2006.

[34] A. Perrig. *Trusted Computing Technologies for Embedded Systems and Sensor Networks*, presentation slides, http://moss.csc.ncsu.edu/ mueller/esns07/perrig.ppt.

[35] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis. *Secure Routing and Intrusion Detection in Ad Hoc Networks*, Proc. 3rd International Conference on Pervasive Computing and Communications, IEEE, 2005.

[36] B. Przydatek, D. Song, and A. Perrig. *SIA: Secure Information Aggregation in Sensor Networks*, Proc. ACM SenSys, 2003.

[37] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. *SPINS: Security Protocols for Sensor Networks*, Wireless Networks Journal (WINE), September 2002.

[38] A. U. Schmidt, N. Kuntze, and M. Kasper. *On the deployment of Mobile Trusted Modules*, Proc. Wireless Communications

and Networking Conference, WCNC'08, 2008.

[39] Trusted Computing Group. *www.trustedcomputinggroup.org*.

[40] TCG Mobile Phone Work Group. *https://www.trustedcomputinggroup.org/groups/mobile*.

[41] H. Wang, and Q. Li. *Distributed User Access Control in Sensor Networks*, Proc. Distributed Computing in Sensor Systems, LNCS 4026, pp. 305-320, 2006.

[42] M. Yarvis, et al. *Exploiting Heterogeneity in Sensor Networks*, Proc. IEEE INFOCOM'05.

[43] Y. Zhang, and W. Lee. *Intrusion Detection in Wireless Ad-hoc Network*, Proc. Mobicom, 2000.

[44] S. Zhu, S Setia, and S. Jajodia. *LEAP: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks*, Proc. ACM Conferenc on Computer and Communication Security, CCS'03, pp. 62-72, 2003.

[45] Y. Zhou, Y. Zhanga, and Y. Fang. *Access control in wireless sensor networks*, Ad Hoc Networks Vol. 5(1), pp. 3-13, Elsvier, 2007.