韓國 컴퓨터情報學會 論文誌
第13卷 第5號, 2008. 9.

2008-13-5-3-14

다양한 무선네트워크 하에서 TCG/TPM을 이용한 정보보호 및 프라이버시 매커니즘

이 기 만*, 조 내 현*, 권 환 우**, 서 창 호***

Security and Privacy Mechanism using TCG/TPM to various WSN

Ki-man Lee *, Nae-Hyun Cho *, Hawn-woo Kwon **, Chang-Ho Seo ***

요 약

본 논문에서는 무선 센서 네트워크 보안 강화의 효율성을 높이기 위한 클러스터된 이기종(heterogeneous) 무선 센서네트워크 구조를 제안하였다. 본 논문에서 제안된 무선 센서 네트워크 구조는 리소스의 제한이 있는 센서 노드와 클러스터 헤드의 역활을 하는 다수의 강력한 하이엔드 장치들로 구성된다. 하이엔드 클러스터 헤드는 센서 노드보다 계산량, 저장공간, 파워 공급, 무선 송신 범위가 뛰어나기 때문에 센서 노드가 겪는 자원의 부족으로 인한 문제점이 발생하지 않는다. 제안된 이기종 무선 센서 네트워크의 구조는 클러스터 헤더에 신뢰 컴퓨팅 기술이 접목되어 있는 것을 특징으로 하며, 특히 각 클러스터 헤더에 신뢰 컴퓨팅 그룹에서 정의한 표준을 따르는 신뢰 플랫폼 모듈이 포함되어 있다. 신뢰 컴퓨팅 그룹에서 정의한 표준에 의하면, 신뢰 플랫폼 모듈은 암호 연산을 수행할 수 있으며 외부 공격으로부터 내부 데이터를 보호할 수 있는 하나의 독립적인 프로세서이다. 또한 호스트에 포함된 신뢰 플랫폼 모듈은 데이터를 안전하게 저장하는 기능과 호스트의 상태를 측정하고 이를 보고하는 기능을 제공함으로써 신뢰 컴퓨팅이 가능하도록 한다.

Abstract

In this paper, To improve the effectiveness of security enforcement, the first contribution in this work is that we present a clustered heterogeneous WSN(Wireless Sensor Network) architecture, composed of not only resource constrained sensor nodes, but also a number of more powerful high-end devices acting as cluster heads. Compared to sensor nodes, a high-end cluster head has higher computation capability, larger storage, 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. A distinct feature of our heterogeneous architecture is that cluster heads are equipped with TC(trusted computing) technology, and in particular a TCG(Trusted Computing Group) compliant TPM (Trusted Platform Module) is embedded into each cluster head. According the TCG specifications, TPM is a tamper-resistant, self-contained secure coprocessor, capable of performing cryptographic functions. A TPM attached to a host establishes a trusted computing platform that provides

• 제1저자 : 이기만
• 접수일 : 2008. 9. 3, 심사일 : 2008. 9. 11, 심사완료일 : 2008. 9. 25.
* 공주대학교 일반대학원 군사정보과학(정보보호전공) 박사과정
** 공주대학교 일반대학원 바이오정보학과(정보보호전공) 박사수료
***공주대학교 바이오정보학과 및 군사정보과학 부교수
※이 논문은 2007년도 한국과학재단 특정기초사업의 지원에 의하여 연구되었음(R01-2007-000-20291-0)

sealed storage, and measures and reports the integrity state of the platform.

▸ Keyword : WSN, Ad-hoc, TCG/TPM, Network Security

Ⅰ. Introduction

A WSN is an ad-hoc network composed of small sensor nodes deployed in large numbers. Sensor nodes are usually severly resource limited and power constrained.

Emerging as an important new technology, WSNs have a wide range of potential applications, especially in the realtime monitoring scenarios, such as battle-eld surveil-

lance, wildlife tracking, healthcare monitoring, emergency response and earthquake monitoring. A WSN consists of a large number of sensor nodes collecting environmental data. The sensor nodes communicate wirelessly and self-organize after being deployed in an ad hoc manner. The nodes are usually severely constrained in computation, storage, communication and power resources.

When deployed in critical applications, mechanisms must be in place to secure a WSN. Security issues associated with WSNs can be categorized into two broad classes content-related security, and contextual security/privacy, Content-related security deals with security issues related to the content of data traversing the sensor network such as data secrecy, integrity, and key exchange[1].

Numerous efforts have recently been dedicated to content-related security issues, such as secure routing [2], key management and establishment [3, 4], access control [5], and data aggregation [6]. In many cases, it does not suffice to just address the content-related security issues . Suppose a sensitive event triggers a packet being sent over the network: while the content of the packet is encrypted, knowing which node sends the packet reveals the location where the event occurs. Contextual security/privacy is thus concerned with protecting such contextual information associated with data collection and transmission.

It is commonly acknowledged that the resource-constrained nature of sensor nodes makes security enforcement in WSNs a challenging task. The majority of the above mentioned efforts attempted to solve security issues in homogeneous WSNs where all sensor nodes have the same capabilities. However, both theoretical and empirical studies have concluded that homogeneous WSNs are not scalable.

In this paper, we propose a clustered heterogeneous architecture for WSNs, where high-end cluster heads are incorporated, and the cluster heads are further equipped with trusted computing technology. As such, the cluster heads act as online trusted parties, helping to effectively address privacy issues in WSNs. We present a scheme for achieving user query privacy and another scheme for achieving source location privacy in the proposed WSN. We are probably the first to apply trusted computing technology to securing generic WSNs, where sensor nodes are too low cost to be equipped with trusted computing hardware.

Ⅱ. Preliminaries: Overview of TCG/TPM

The latest effort in trusted computing is represented by the Trusted Computing Platform specifications defined by TCG [7]. The specifications aim to provide hardware based roots of trust through a tamper resistant coprocessor, TPM. A TPM is attached to a host machine and acts as the root of trust of the host platform, given its tamper resistance property. TPM is capable of performing cryptographic functions such as random number generation, SHA1 hash function, and RSA encryption and digital signature.

A core functionality provided by TPM is integrity measurement and storage, and reporting of the state of the host platform. Integrity measurement and storage are achieved through a set of PCRs(Platform Con-guration Registers), internal to TPM. Each PCR value is a 20-byte SHA1 hash digest 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 loading of the protected applications. A update to a PCR value is through what is termed extending the PCR, which is described as

$$PCR[i] \leftarrow SHA1(PCR[i] \,||\, wnewly\ measured\ value)$$

where i is the index of the PCR being updated. Since a PCR value is a digest of the platform state, it is meaningless by itself. The data that complements PCRs in providing semantics is SML(Stored Measurement Log). 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 state of the host platform.

Upon request, TPM can report the state of its underlying platform to a remote challenging entity through attestation. In particular, TPM has a number of key pairs called AIKs(Attestation Identity Keys), which are used as aliases of the unique EK(Endorsement Key). 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 AIK certi-cate to the challenging entity. (5) The challenging entity veri-es the replied data - the AIK certi-cate is validated, the measurement digest is computed from the SML entries and compared with the signed PCR values.

Another security function provided by TPM is Sealed Storage, which encrypts sensitive data with integrity measurement values. In particular, the data to be protected is encrypted/sealed together with one or more PCR values. Subsequently, TPM releases an encrypted data only if the current PCR values match those stored during encryption. 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 SRK(Storage Root Key) internal to TPM, or by a key protected by the SRK.

To end this section, we highlight an essential distinction between the TCG trusted computing technology and other trusted computing initiatives such as IBM PCIXCC [8]. The trusted hardware for the former is used to ensure the state of a protected application which executes in the host machine and uses the ample resources of the host. By contrast, under the latter model a protected application runs within the trusted hardware, and thus is severely limited by the constraint of the trusted hardware.

III. A TC-enabled Heterogeneous Architecture for WSNs

3.1 The Architecture

We partition a WSN into a number of clusters. A high-end device is placed into each cluster, acting as the cluster head. In contrast to sensor nodes, high-end cluster heads have relatively higher

computation capability, larger storage size, and longer radio range. They also have longer power supply, and in some circumstances they can even be line-powered, e.g., when a WSN is deployed to monitor a building, the cluster heads can easily tap on the electricity lines to get power supply. Therefore unlike sensor nodes, cluster heads do not suffer from the resource scarceness problem. The introduction of high-end cluster heads into a WSN makes the once homogeneous network heterogeneous. The general heterogeneous architecture is depicted in Fig1.

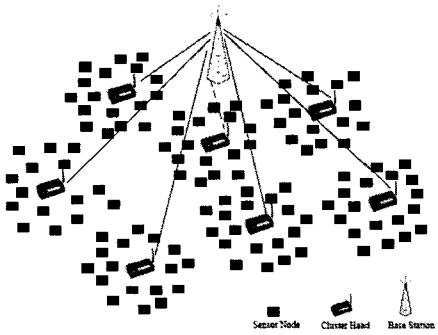

그림 1. 이질적인 무선 센서 네트워크
Fig 1. Heterogeneous Wireless Sensor Network

Downlink communication (from base station to sensor nodes) and uplink communication (from senor nodes to base station) in the architecture are asymmetric. Messages broadcast by the base station can directly reach sensor nodes, whereas messages sent by a sensor node need to be forwarded by its corresponding cluster head. As a result, uplink communication follows a hierarchical manner and consists of intra-cluster and inter-cluster communications, respectively.

3.2 Configuration of Cluster Head

Depending on application scenarios, hardware capabilities of cluster head 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 MTM(Mobile Trusted Module), similar to TPM for PCs [9]. Prototype implementation of MTM were already available (e.g.,). Hence, there exists no technical barrier to implement our envisioned TC-enabled cluster head.

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 choose to design the cluster head as a restricted trusted computing platform due to its specialized functionality and application in WSNs. A reference platform configuration of cluster head is shown in Fig 2. The platform runs the sole ClusterH application. At the application layer, the ClusterH 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 mechanisms. The security module is dedicated to implementing the designated WSN security mechanisms (e.g., the algorithms to implement user query privacy and source location privacy). The sensor agent is the communication interface with sensor nodes, while the base station agent is the interface with the base station or other cluster heads. The OS layer implements the secure kernel, bridging between the application layer and the hardware layer. The hardware layer includes a TCG-compliant TPM/MTM, providing hardware based root of trust.

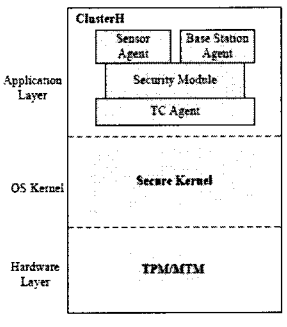

그림 2. 클러스터 헤드 플랫폼 구성 참고
Fig 2. A Reference Cluster Head Platform Configuration

IV. A Scheme Achieving User Query Privacy

Equipped with TPM/MTM, the secure kernel and the ClusterH software, cluster heads act as online trusted parties. To show the effect of the trusted cluster heads on security enforcement, in this section we present solutions to two important contextual security/privacy problems in WSNs: user query privacy and source location privacy. Compared to existing solutions in [1], our schemes achieve better privacy and higher efficiency.

4.1 Problem Statement

We WSNs are often deployed to provide services to other users than the network owner [10]. Users are allowed to query a network to get sensed data from particular areas. In such a scenario, a user may wish to protect her "areas of interest" from being disclosed to other users or even the network

owner. User query privacy is thus concerned with the following problem: suppose a user queries the network, intending to get the sensed data in cluster c_i, a user query privacy scheme ensures that the user ends up getting the desired data, but the adversary does not learn c_i by observing the communication.

4.2 Our Scheme

4.2.1 Network Model

We support roaming users querying a wireless sensor network. The network follows the heterogeneous architecture proposed earlier: the whole network is partitioned into a set of n clusters, $c_1, c_2, \cdots, c_n$ where c_i is the identifier of the ith cluster; each cluster is grouped around a TC-enabled cluster head and we denote chi the cluster head in c_i. A user who desires to query the network frist contacts the nearest cluster head within her proximity, through which she will issue queries. This cluster head is called access point. Taking Fig 3. as an example, ch_1 of c_1 is the access point for the user.

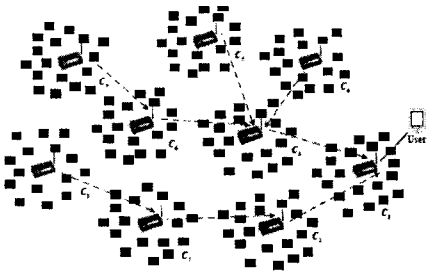

그림 3. 사용자 이용 Her 액세스 포인트로서의 c_1의 클러스터 헤드

Fig 3. A User Uses the Cluster Head of c_1 as Her Access Point

4.2.2 Assumption

To obtain services from a WSN, a user is assumed to have a certain means to authenticate to the WSN. Also, a TC-enabled cluster head can authenticate to users using the AIK of its embedded TPM/MTM. Therefore, the access point and the querying user can accomplish mutual authentication, based on which we assume the two entities share a secret key for semantic secure symmetric encryption. Further, it is also easy for each cluster head to get this secret key from the access point, since they can clearly authenticate each other with the help of their respective AIKs. We denote $\varepsilon_{ch_i}(.)$ and $\vartheta_{ch_i}()$ the encryption and decryption, respectively, by ch_1 using this secret key. We also assume each cluster head shares a secret cluster key (for semantic secure symmetric encryption) with all sensor nodes in its cluster. Several well studied key exchange schemes [11] can achieve this objective. $c\varepsilon_{c_i}(.)$ and $c\vartheta_{c_i}()$ denote the encryption and decryption, respectively, using the cluster key of c_i.

4.3 Overview of the Scheme

A straightforward way to achieve query privacy is that every cluster head sends encrypted data to the access point, who then forwards only the data desired by the user. This however unnecessarily wastes communication bandwidth among cluster heads. Even for this straightforward method, we should still be very cautious not to leak information about the queried cluster from the size of the data returned to the user. More specifically, clusters normally have different number of sensor nodes, so data from different clusters are likely to have different lengthes.

Let us suppose the data from each sensor node forms a packet for simplicity. The total number of packets from a cluster equals the number of sensor nodes. Without privacy treatment, the number of packets eventually returned to the user by the access point would clearly indicate the cluster from which the data originates. A method to fix this problem is that regardless of which cluster is queried, the access point returns a fixed-number of packets, corresponding to the biggest cluster size. We use l to denote this number thereafter. For a cluster whose size is smaller than l, dummy packets are generated.

In our approach, every cluster head sends out l packets. Due to the semantic security of encryption, re-encryptions of the same data are not distinguishable. Therefore, the adversary watching the network cannot tell if the l packets sent out by a cluster head originate from the cluster head itself or from its dependent nodes

4.3.1 Scheme Details

A complete description of the scheme in pseudo code is shown in Algorithm 1,

where u denotes the querying user and ap denotes the access point.

To start, the user contacts the access point by sending a hello message, including a nounce that will be used in the ensuing attestation process (Step

Algorithm 1 The Scheme Achieving User Query Privacy

```
1  u → ap: hello
2  ap → {ch_i}: establish routing paths.
3  u → ap: attestation
4  u → ap: e_u = E_k(c) /* c is the identifier of the target cluster*/
5  ap → {ch_i}: e_u
6  for EACH ch_i do
7    c = D_k(e_u)
8    {sensor} = ch_i: data_ch_i = {sense_i(sensed_data)}
9    {dependent_node} = ch_i: {packets} /*ch_i waits until gets packets from all its dependent nodes*/
10   if ch_i = c then
11     /*ch_i is cluster head of the target cluster c */
12     packets = GD_k(data_ch_i)
13     packets = header||content = E_c(c)||E^l_ck(packets)
14   else
15     foundquerriedcluster = FALSE
16     for packets from EACH dependent_node do
17       c' = D_ck(head)
18       if c' = c then
19         content' = D_ck(content)
20         packets = head||content = E_ck(c)||E^l_ck(content')
21         foundqueriedcluster = TRUE
22         break
23       end if
24     end for
25     if foundquerriedcluster == FALSE then
26       packets = head||content = E_ck(c,1,...,l)E^l_ck(dummy,data)
27     end if
28   end if
29   ch_i → forward_node: packets
30 end for
31 ap → u: packets
```

1). The access point then informs all the other cluster heads to form routing pathes using the method described earlier (Step 2). Before sending a query, the user must have assurance of the trustfulness of the cluster heads. This is achieved by means of attestation (Step 3). Note that it is unnecessary for the user to check the status of all cluster heads, which is quite expensive; it suffices to adopt the strategy of "chained attestation" along the established routing pathes. In particular, referring to Fig 3, u only verifies the access point; ch_9 is veri-ed by ch_6; ch_4, ch_5 and ch_6 are verified by ch_3; ch_8 is verified by ch_7 who is in return verified by ch_2; ch_2 and ch_3 are veri-ed by the access point.

Once attestation is successful, the user sends to the access point the query e_u, which is the encryption of the identifier c of the target cluster using the shared secret key (Step 4). The access point broadcasts the query to all other cluster heads (Step 5), each decrypting the query and knowing which cluster the user is querying (Step 7). Each cluster head then collects sensed data (encrypted using the cluster key) from the sensor nodes of its cluster (Step 8). Before sending out l packets of data to its forwarding node (Step 29), a cluster head must wait until it receives packets from all its dependent nodes (Step 9). Afterwards, if the cluster itself is the target cluster (Step 10-13), the cluster head simply ignores the packets from its dependent nodes, and encrypts the sensed data from its cluster. Note that every set of l packets consists of head and

content, where the head is used to inform cluster heads enroute the origin of the l packets while without decrypting the content. For a cluster that is not the target one (Step 14-27), the cluster head checks whether one of its dependent nodes sends in the data of the target cluster. If yes, the cluster head re-encrypts the data (Step 16-22); otherwise, the cluster head generates l dummy packets (Step 25-27). Eventually, the access point passes the l packets of the target cluster to the querying user (Step 31).

4.3.2 Security Analysis

We argue security of our scheme from two aspects: communication data and communication patterns. For the first aspect, we define view of a query to be all data communicated across the network to answer the query. It is not difficult to prove that for any query q, any PPT adversary A, there exists a PPT simulator A^* such that $|\Pr[A(view) \neq f(q)] - \Pr[A^*(struc) = f(q)]|$ is negligible, as long as the encryption scheme is a pseudo-r andom permutation, where $f(q)$ is any

function on the result of query q, and struc is the structure of the underlying sensor network. This suggests that the data communicated do not divulge query privacy. For the second aspect, it is clear that every query results in the same communication pattern, i.e., every cluster head reads sensed data from the sensor nodes in its cluster; every cluster head sends out l packets to its forwarding node after receiving data from all its dependent nodes.

Altogether, observing communication in the network does not in any way help the adversary to figure out which cluster is the query target.

4.3.3 Improvement

In the above scheme, to answer a user query all the sensor nodes are asked to send data to their respective cluster heads. This may shorten the lifetime of the network because of excess energy consumption. To mitigate this problem, We can alternatively trade off data freshness for energy efficiency, especially when queries come in at a high rate. In particular, sensor nodes periodically provide the sensed data to their respective cluster heads, who cache the data. The cluster heads then handle user queries using the cached data rather than collecting realtime data from the sensor nodes.

V. Conclusion and Future Work

Due to stringent resource limitations of sensor nodes, security enforcement is extremely challenging in wireless sensor networks. To solve this problem, we proposed to render a wireless sensor network heterogeneous, by incorporating TC-equipped high-end devices into clusters of the network, acting as cluster heads. We demonstrated how the TC-enabled cluster heads can effectively address privacy issues in WSNs.

This study is still in the preliminary stage. We are preparing to implement proof-of-the-concept TC-enabled WSN architecture, and further experiment with the architecture in certain real world wireless sensor network settings.

참고문헌

[1] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing Source-Location Privacy in Sensor Network Routing, Proc. 25th

[2] 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.

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

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

[5] H. Wang, and Q. Li. Distributed User Access Control in Sensor Networks, Proc. Distributed Com-puting in Sensor Systems, LNCS 4026, pp. 305-320, 2006.
[6] B. Przydatek, D. Song, and A. Perrig. SIA: Secure Information Aggregation in Sensor Networks,Proc. ACM SenSys, 2003.
[7] Trusted Computing Group. https://www.trustedcomputinggroup.org.
[8] T. Arnold, and L. V. Doorn. The IBM PCIXCC: A New Cryptographic Coprocessor for the IBM EServer. IBM Journal of Research and Development 48 (May 2004).
[9] TCG Mobile Phone Work Group. https://www.trustedcomputinggroup.org /groups/mobile.
[10] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, Query Privacy in Wireless Sensor Networks, Proc. 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON '07, pp. 203-212, 2007.
[11] D. Liu, P. Ning, and K. Sun. E±cient Self-Healing Group Key Distribution with revocation Capability,Proc. ACM Conference on Computer and Communication Security, CCS'03, 2003

저 자 소 개

이 기 만(Ki-man Lee)

1986년 : 광주대학교 전자계산학과 (학사)
1990년 : 연세대학교 산업대학원 (석사)
2006년 : 공주대학교 일반대학원 군사정보과학(정보보호전공) 박사수료

조 내 현(Nae-Hyun Cho)

1987년 : 국방대학원 운영분석(석사)
2007년 : 공주대학교 일반대학원 군사정보과학(정보보호전공) 박사과정

권 환 우 (Hawn-woo Kwon)

1987년 : 영남대학교 전자공학과(학사)
2004년 : 공주대학교 일반대학교 수학과(이학석사)
2005년 : 공주대학교 일반대학원 바이오정보학과(정보보호전공) 박사수료

서 창 호 (Changho Seo)

1990년 : 고려대학교 수학과(학사)
1992년 : 고려대학교 일반대학원 수학과 (이학석사)
1996년 : 고려대학교 일반대학원 수학과 (이학박사)
2001년~현재 : 공주대학교 바이오정보학과 및 군사정보과학 부교수