国立交通大學

資訊科學與工程研究所

博士論文

無線感測網路之安全資料聚集與資料搜尋方法設計

The Design of Secure Data Aggregation and Data Searching for Wireless Sensor Networks

研究生：黃士一

指導教授：謝續平 博士

中華民國九十九年一月

The Design of Secure Data Aggregation and Data Searching for Wireless Sensor Networks

Student: Shih-I Huang

Advisor: Shiuhpyng Shieh

A Dissertation Submitted to
Institute of Computer Science and Engineering
College of Computer Science
National Chiao-Tung University
In Partial Fulfillment of the Requirements
of the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

Jan, 2010

Hsinchu, Taiwan, R.O.C.

無線感測網路之安全資料聚集與資料搜尋方法設計

研究生：黃士一　　　　　　　　　　　指導教授：謝續平

國立交通大學資訊科學與工程研究所

摘要

無線感測網路中感測器(Sensor Node)之間的通訊方式則是採用無線通訊方式，每個感測器持續的傳送感測器的讀值(Reading)，並將讀值傳到無線資料收集器上再加以處理，因此在同一時間內會有大量的資料在無線感測網路中傳送，造成網路雍塞及感測器耗損大量的電力，並進而減損整個網路的使用時效(Lifetime)。為了克服這個問題，有許多研究便以「資料聚集」(Data Aggregation)的方式來減少資料的傳遞量，但由於Sensor是隨意地散佈於環境四周，因此資料在聚集或傳送時極易遭到監聽，入侵或修改，因此如何在Sensor彼此之間建立起認證的管道，以確保資料是正確的傳遞到接收端便是一個重要的議題。

本篇論文所探討的主題為無線感測網路裡資料聚集的安全機制。第一章中我們會深入的介紹目前常見的資料聚集方法，並針對不同的資料聚集方法在安全上的漏洞加以整理分析，並針對目前的防護方式加以整理與討論。第二章著重於無線網路的資料聚集方法，在第二章我們提出了一個安全的資料聚集方法，資料可以在不被第三者知悉且資料是加密的情形下，將重複的資料剔除。本論文所提出之方法可適用於低成本且低算能力的無線感測器上，並只需要$O(n)$的金鑰空間，可實作在無線感測器上，進而強化無線感測器的安全能力。

第三章中我們提出了在可感知RFID之無線感測網路中認證及輕量(lightweight)的資料搜尋方法。我們提出一個混合RFID及感測器所構成的無線感測網路架構(ARIES)，以及在這個架構下的相互認證方法(Mutual Authentication)，及一個提供使用者可以在資料是加密

且不須解密的情形下，可以搜尋密文中是否有特殊的字串的搜尋方法。本章所提出的可感知RFID之無線感測網路結合了RFID與無線感測網路，可解決無線感測網路中的距離限制問題。本章所提出的認證方法可以適用於無線感測網路並減少重新認證的次數。本章所提出的資料搜尋方法可以在資料不須解密的情形下搜尋特定字串，藉此，可以避免資料在無線網路中傳送遭到竊取或破壞。

而受制於無線感測器的硬體限制，可以儲存金鑰的空間極少，因此第四章中我們提出了一個金鑰建佈(Key Distribution)的方法，我們利用了Hash Chain建立Pair-wise金鑰，使得每個Node所需要的金鑰儲存空間更少，但仍保存相同的Network Connectivity，及藉此建立起點與點的金鑰，達到點與點的安全性。

藉由本論文提出的方法，可以建立起無線感測網路的安全防禦機制。首先，藉由我們所提出的金鑰建佈方法，可以建立起點與點的安全溝通管道(Secure Communication Channel)且只需要較少的儲存空間。而第三章提出的資料搜尋方法，提供了一個在加密資料的搜尋機制，除了確保資料安全外，更加上搜尋的功能。而第二章提出的資料聚集方法，除了維持資料的安全性及完整性(Integrity)外，讓感測器可以濾掉由不同的感測器且每個感測器有不同的加密金鑰的情形下，濾掉重複的資料，除了可以避免資料遭到破壞或竄改外，更可以延長無線感測網路的整體平均壽命。本論文除了考慮安全性外，更考慮了省電性，且相關設計都以可以在無線感測上實現為優先，可以做為無線感測網路的安全基礎建設(Security Infrastructure)。

The Design of Secure Data Aggregation and Data Searching for Wireless Sensor Networks

Student: Shih-I Huang Advisor: Shiuhpyng Shieh

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

Wireless Sensor Networks (WSNs) are formed by a set of small devices, called nodes, with limited computing power, storage space, and wireless communication capabilities. Most of these sensor nodes are deployed within a specific area to collect data or monitor a physical phenomenon. Data collected by each sensor node needs to be delivered and integrated to derive the whole picture of sensing phenomenon. To deliver data without being compromised, WSN services rely on secure communication and efficient key distribution. This paper focuses mainly on establishing security protection in WSNs.

The first part of the paper proposes a secure encrypted-data aggregation scheme for wireless sensor networks. Our design for data aggregation eliminates redundant sensor readings without using encryption and maintains data secrecy and privacy during transmission. Conventional aggregation functions operate when readings are received in plaintext. If readings are encrypted, aggregation requires decryption creating extra overhead and key management issues. In contrast to conventional schemes, our proposed scheme provides security and privacy, and duplicate instances of original readings will be aggregated into a single packet. Our scheme is resilient to known-plaintext attacks, chosen-plaintext attacks, ciphertext-only attacks and man-in-the-middle attacks. Our experiments show that our proposed aggregation method significantly reduces communication overhead and can be practically implemented in on-the-shelf sensor platforms.

The second part of the paper investigates authentication and secure data retrieval issues in RFID-aware wireless sensor networks. To cope with the problems, we proposes a network architecture (*ARIES*) consisting of RFIDs and wireless sensor nodes, a mutual authentication protocol (*AMULET*), and a secret search protocol (*SSP*). ARIES utilizes RFID-aware sensor nodes to alleviate the distance limitation problem commonly seen in RFID systems. AMULET performs mutual authentication and reduces the cost of re-authentication. SSP solves the privacy problem by offering a lightweight secret search mechanism over encrypted data, thereby preventing data disclosure during communication and query processes. The proposed scheme only uses symmetric cryptosystems, and does not need to decrypt encrypted data files while searching for specific data. In this way, fewer decryption and encryption operations are needed, and the performance of secret search and data retrieval is greatly improved.

In last part, we proposed two key distribution schemes for WSNs, which require less memory than existing schemes for the storage of keys. The Adaptive Random Pre-distributed scheme (ARP) is able to authenticate group membership and minimize the storage requirement for the resource limited sensor nodes. The Uniquely Assigned One-way Hash Function scheme (UAO) extends ARP to mutually authenticate the identity of individual sensors, and can resist against the compromise of sensor nodes. The two proposed schemes are very effective for the storage of keys in a wireless sensor network with a large number of sensors.

Keywords: Data Aggregation, Data Searching, Wireless Sensor Networks, Authentication, Privacy, Key Management

Acknowledge

經過七年的磨練，終於完成了博士學位。回想起七年前，還只是個初出社會二年的毛頭小子，捨棄一份不錯的薪水及工作，然後回到學校再就學。不同於一般求學的學生，在求學中間背負很多艱辛且不為人知的壓力，也曾受到某些人的曲解，雖然過程不能盡如人意，但也都一點一點的承受住，熬了過來。

這本論文的產生，首先要感謝我的指導教授謝續平老師。這七年中間，他給我不止課業上的指導，亦讓我在他身上學到許多做人做事的道理，讓我在面對困難及挑戰時，更有能力去應對。第二要感謝柏克萊大學的 Prof. Doug Tygar 及卡內基美隆大學的 Prof. Adrian Perrig，當我在柏克萊大學及卡內基美隆大學做研究時，他們不遺餘力的教導及頻繁的 meeting，讓我能快速的產出期刊論文及國際會議論文。而工研院的長官們，包括余孝先副所長、劉智友組長、張隆昌副組長等；及同事們，包括力群、博元等，在我於工研院服務的日子裡，給我莫大的支持及幫忙，讓我能同時兼顧學業、研究及工作，並且能因此到此二頂尖學府進修，讓我藉此培養更廣的國際觀。第四要感謝富源學長及嘉寧學長在研究上的大力協助，藉由跟他們的討論，讓我得知如何做更好的研究及了解自己研究的問題點。

此外，也要感謝我的口試委員：曾文貴教授、楊武教授、吳宗成教授、雷欽隆教授、孫宏民教授，及詹進科教授，在百忙中撥空，提供許多寶貴的意見與指正，使本論文能略臻於完備。

最後，我要以這份博士學位的榮耀獻與我最親愛的父母與家人，在這些日子裡，你們容忍我的不理性及任性，在我出國工作時，幫我照顧家庭，讓我能兼顧工作及求學，沒有您們的支持，將不會有這本論文。謹與我最親愛的父母與家人分享。

黃士一 2010 年 1 月 20 日于台中

Table of Content

List of Figures

List of Tables

Chapter 1

Introduction

With the popularity of laptops, cell phones, PDAs, GPS devices, RFID, and intelligent electronics in the post-PC era, computing devices have become cheaper, more mobile, more distributed, and more pervasive in daily life. It is now possible to construct, from commercial on-the-shelf components, a wallet size embedded system with the equivalent capability of a 90's PC. Such embedded systems can be supported with scaled down Windows or Linux operating systems. From this perspective, the emergence of wireless sensor networks (WSNs) is essentially the latest trend of Moore's Law toward the miniaturization and ubiquity of computing devices.

Typically, a wireless sensor node (or simply sensor node) consists of sensing, computing, communication, actuation, and power components. These components are integrated on a single or multiple boards, and packaged in a few cubic inches. With state-of-the-art, low-power circuit and networking technologies, a sensor node powered by 2 AA batteries can last for up to three years with a 1% low duty cycle working mode. A WSN usually consists of tens to thousands of such nodes that communicate through wireless channels for information sharing and cooperative processing. WSNs can be deployed on a global scale for environmental monitoring and habitat study, over a battlefield for military surveillance and reconnaissance, in emergent environments for search and rescue, in factories for condition based maintenance, in buildings for infrastructure health monitoring, in homes to realize smart homes, or even in bodies for patient monitoring. After the initial deployment, sensor nodes are responsible for self-organizing an appropriate network infrastructure, often with multi-hop connections between sensor nodes. The onboard sensors then start collecting acoustic, seismic, infrared or magnetic information about the environment, using either continuous or event driven working modes. Location and positioning information can also be obtained through the global positioning system (GPS) or local positioning algorithms. This information can be gathered from across the network and appropriately processed to construct a global view of the monitoring phenomena or objects. The basic philosophy behind WSNs is that, while the capability of each individual sensor node is limited, the aggregate power of the entire network is sufficient for the required mission.

In a typical scenario, users can retrieve information of interest from a WSN by injecting queries and gathering results from the so-called base stations, which behave as an interface between users and the network. In this way, WSNs can be considered as a distributed database. It is also envisioned that sensor networks will ultimately be connected to the Internet, through which global information sharing becomes feasible.

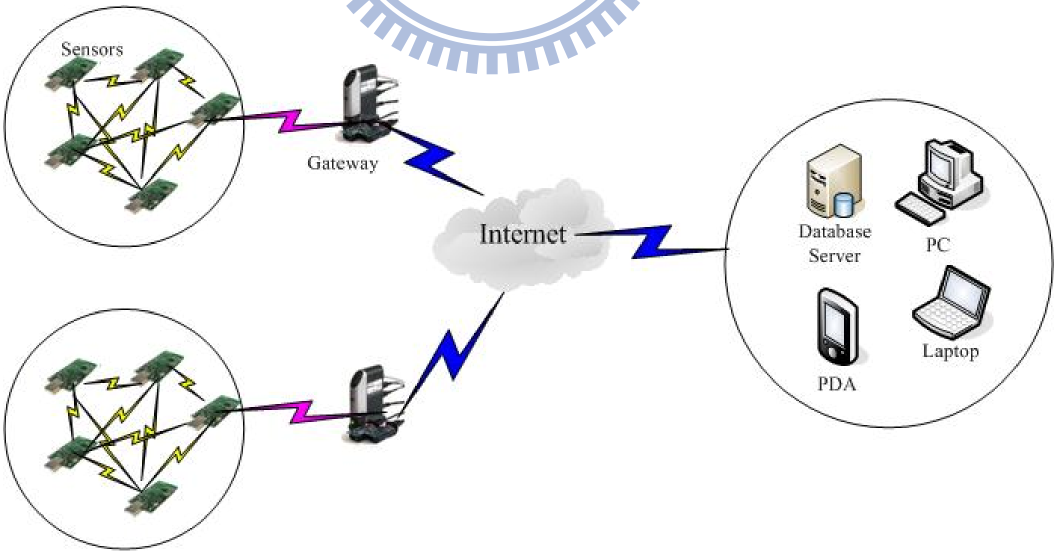


Figure 1 Wireless Sensor Network Architecture

Security Requirement

Due to resource limitations of WSN, the security requirements of WSN are different from others. Below we list security requirements which must be fulfilled in wireless sensor network architecture.

1) *Secrecy*: Storing data in an encrypted form helps retain its confidentiality. Because sensors are vulnerable, computation-limited, and low cost devices, allowing sensors to decrypt data to perform a search results in unnecessary risk of disclosure. Thus, sensors must execute a secret search directly on ciphertext, rather than plaintext. Furthermore, data transmitted over a wireless interface is susceptible to exposure. Therefore, sensors must only transmit encrypted data. In summary, the data must remain in an encrypted form and should not be decrypted unless necessary to minimize the possibility of disclosure.

2) *Authentication*: Since the network obtains data from a large number of sensors or tags, attackers can easily acquire readers with the same specifications to extract data stored in the tags. Therefore, both the reader and the tag need to verify the authenticity of its communication counterpart before executing read or write operations.

3) *Integrity*: Assuring data integrity prevents attackers from using unauthorized readers to modify or inject data into databases. Readers or tags must verify data integrity upon receipt of data.

4) *Performance*: Requiring a sensor node to decrypt data before searches causes significant and unnecessary delay. Also, the limited computation capabilities of sensor nodes and tags hinder them from performing complex operations, such as encryption and exponential calculations. Therefore, all operations must be redesigned to fit their computation capabilities.

Contribution

The contribution of this dissertation is threefold. First, we build a key distribution algorithm, associating random key distribution and hash chain, achieving end-to-end security requirement but saving more storage space. Second, we design an authentication and secret search mechanisms for WSN. Our authentication algorithm achieves mutual authentication among sensor nodes but reducing re-authentication effort. Our proposed secret searching algorithm provides searching method over encrypted data without knowing encryption / decryption key. This could keep both secrecy and privacy for data. Third, we proposed a data aggregation algorithm over encrypted data. Aggregators do not need to have encryption keys and could still perform data aggregation to eliminate redundant data without decrypting them.

Our proposed algorithms aim not only to provide secrecy, privacy, authentication, and data integrity for data, but also aim to be workable in resource limited sensor nodes. Only feasible security algorithms and affordable computation assumption can practically provide robust security to wireless sensor network.

Synopsis

In Chapter 2, we list related work for key distribution, authentication, secret searching, and data aggregation. We showed that the infeasibilities of current algorithms in these four categories.

In Chapter 3, we proposes a new method for determining and eliminating duplicate data while protecting privacy (using encryption) without excessive key-management or power management issues. Our scheme has the following contributions. First, we provide a lightweight data aggregation mechanism which protects data when data are processed in aggregators. Aggregators can help to eliminate redundant data without decrypting data. Thus, aggregators do not need to spend extra power in data decryption, and more network lifetime can be guaranteed. Second, our proposed scheme is resilient to known-plaintext attacks, chosen-plaintext attacks, ciphertext-only attacks, and man-in-the-middle attacks.

In Chapter 4, we propose an architecture consisting of passive RFIDs and RFID-aware sensor networks (*ARIES*). This architecture extends RFID's capabilities through a wireless sensor network by utilizing sensor nodes to locate targets at a distance. Second, we design a private mutual authentication protocol (*AMULET*) which is feasible for RFIDs and sensor nodes, and reduces the cost of re-authentication. Third, we present a secret search protocol (*SSP*) that enables readers to perform searches over encrypted data, allowing data to remain encrypted during transmission or at vulnerable locations. By only using one-way hash functions, pseudo random number generation functions, and XOR operations, SSP accommodates the resource limitations of both tags and sensors. In addition, SSP can solve the problem that same plaintexts at different places will be encrypted into the same ciphertexts.

In Chapter 5, we propose two key distribution schemes: Adaptive Random Pre-distributed scheme (ARP) and Uniquely Assigned One-way Hash Function scheme (UAO). Both schemes pre-distribute keys in each node before its deployment. According to random graph theory, a sensor network can be connected as long as enough keys are selected. Therefore, each node can communicate with each other without key exchange, which can save computational overhead for communications. More than that, both schemes minimize the storage requirement for key management. Though UAO scheme needs more storage space than ARP does, it provides mutual authentication. And, Chapter 6 gives an conclusion.

Chapter 2

Related Work

Much research has been done on key distribution in WSN over the past few years. Carman et al [18] analyzed various conventional approaches for key generation and key distribution in WSN on different hardware platforms with respect to computation overhead and energy consumption. The results showed that conventional key generation and distribution protocols are not suitable for WSN. To cope with the problem, a key management protocol [19] is proposed for sensor networks, which is based on group key agreement protocols and identity-based cryptography. This protocol used Diffie-Hellman key exchange scheme to perform group key agreement. However, the high storage and high computation requirements make it unrealistic to be implemented.

Perrig et al. [20] proposed a security protocol for sensor networks named SPINS. SPINS uses base station as a trusted third party to set up session keys between sensor nodes. Liu and Ning [21] extended Perrig's scheme and proposed an efficient broadcast authentication method for sensor networks. Their scheme uses multi-level key chains to distribute the key chain commitments for the broadcast authentication. Undercoffer et al [22] proposed a resource-driven security protocol, which consider the trade-off between security levels and computational resources. However, in a randomly dispersed wireless sensor network, the base station is not always available for all nodes. Without the base station, SPINS may cause a sensor network disconnected. Therefore, these schemes are not well suitable for sensor networks due to the need of the base station.

Eschenauer and Gligor [23] proposed a key management scheme based on Random Graph Theory. The Random Graph Theory is defined as follows. A random graph G(n, p) is a graph with n nodes, and the probability that a link exists between any two nodes in the graph is p. When p is zero, the graph G has no edges, whereas when p is one, the graph G is fully connected. This approach significant reduce key space requirement. Inspired by this paper, our algorithm could save more key space in compared with Eschenauer's algorithm.

For authentication and secret searching, some papers propose the use of public key infrastructure (PKI) to authenticate two parties through a trusted-third-party [43]. This solution is inadequate for RFID applications since the PKI requires the reader or tag to save private keys and verify the identity of others with the help of the trusted-third-party. Tags have little storage, and they can only transmit data to devices in close proximity. In other words, the trusted-third-party must be located near the tags, which is a difficult requirement to achieve and one that presents other security risks. Moreover, the tag cannot afford additional computational power required to verify others. Therefore, a PKI scheme is not feasible for RFID applications.

A randomized lock protocol [26] was proposed for private authentication in a highly constrained computation and storage environment. However, this scheme is neither private nor secure against passive eavesdroppers. As an improvement, a PRF-based private authentication protocol [5] was proposed. Unfortunately, both protocols [26][5] require re-authentication of a tag even if another authorized reader previously authenticates the tag. These extra steps are computationally wasteful and unnecessary.

Privacy is a major concern encountered in RFID applications [39]. A RFID tag may store sensitive data associated with a target, which must remain private. Since readers, tags, and sensor nodes send messages through a wireless medium, attackers can easily eavesdrop to their communication and extract secret information.

An intuitive way of protecting private data is encryption [30]. However, tags and sensor nodes have severely limited storage and computation capability; consequently, conventional cryptographic algorithms are not well-suited for these devices. As a result, we must redesign security mechanisms to support RFID tags and sensor nodes.

A new problem arise from encrypting data: RFID readers cannot easily perform queries on data in encypted form [21][5]. Researchers have investigated secret search over encrypted data in an untrusted file server or external memory environment [55][26][23][5][27]. A recent method [3] is proposed for secret searching on untrusted servers. Unfortunately, their scheme requires complex encryption operations unavailable to both tags and sensor nodes. Another problem of the scheme is that same plaintexts at different places will be encrypted into the same ciphertexts in their proposed scheme III. Hence, malicious attackers could inject meaningful plaintexts into the database and use the corresponding ciphertexts to find their interests without decrypting entire or part of the database.

Other researches tried to solve this searching problem by inserting specific encrypted keywords into the ciphertexts [43][89][8][68][22]. These encrypted keywords can be viewed as indices and could therefore be used in search operations [13][95]. However, these keywords are fixed and must be defined beforehand. Therefore, this inconveniency makes them difficult to use. Another solution is to support searching over encrypted data by using multi-party computation and oblivious functions [82][83][34]. However, this solution requires high computation overhead and therefore is not applicable in a tag or sensor system.

For data aggregation, previous work in data aggregation assumes that every mote is honest and only transmits their correct readings. Intanagonwiwat, Govindan, Estrin, and Heidemann [51] proposed a data-centric diffusion method to aggregate data. Their method enables diffusion to achieve energy savings by selecting empirically good paths and by caching and processing data in-network. Though their method can achieve significant energy savings, security is not put into consideration in their design.

Hu and Evans [48] further examined the problem that a single compromised sensor mote can render the networks useless, or worse, mislead the operator into trusting a false reading. They proposed an aggregation protocol that is resilient to both intruder devices and single device key compromises, but their scheme suffers a problem that the aggregated data will be expanded every time when it was aggregated and forwarded by any intermediate sensor mote.

Przydatek, Song and Perrig [72] proposed a secure information aggregation protocol to answer queries over the data acquired by the sensors. In particular, their proposed protocols are designed especially for secure computation of the median and the average of the measurements, for the estimation of the network size and for finding the minimum and maximum sensor reading. Even though their scheme provided data authentication to provide secrecy, the data is still delivered in plaintext format which provides no privacy during transmission.

Wagner [88] presented a paper studying related attacks on data aggregation in sensor networks. He thoroughly examined current aggregation functions and proved that these aggregation functions are vulnerable and insecure under several attacks. He also proposed a theoretical framework for evaluating data aggregation resiliently in sensor networks and in its security against these attacks. Still privacy is not guaranteed in his scheme.

Acharya and Girao [1] proposed an end-to-end encryption algorithm supporting operations over ciphertexts for wireless sensor networks. Their scheme uses a special class of encryption functions, *Privacy Homomorphisms (PH)* [9][12][25][28][94], that allow end-to-end encryption and provide aggregation functions that are applied to ciphertexts. PH is an encryption transformation that allows direct computations on encrypted data. Two functions E and D are additively homomorphic encryption and decryption if the following property is satisfied: for plaintext operands x and y and key k, $x+y=D_k(E_k(x)+E_k(y))$. However, privacy homomorphisms have exponential bound in computation. It is too computationally expensive to implement in wireless sensor networks. Moreover, it has been proved that privacy homomorphism is insecure even against ciphertext only attacks which are commonly encountered in wireless sensor networks.

Cam et al. [15] proposed a secure energy-efficient data aggregation (*ESPDA*) to prevent redundant data transmission in data aggregation. Unlike conventional techniques, their scheme prevents the redundant transmission from sensor motes to the aggregator. Before transmitting sensed data, each sensor transmits a secure pattern to the aggregator. The secure pattern is generated by associating original data with a random number. Instead of transmitting "real" data, the sensor mote transmits the secure patter to the cluster-head before transmitting it. The cluster-head then uses these secure patterns to check which sensors have same readings. Then, the cluster-head notifies certain sensor motes to transmit their data. Only sensors with different data are allowed to transmit their data to the cluster-head. However, since each sensor at least needs to transmit a packet containing a pattern once, power cannot be significantly saved. In addition, each sensor mote uses a fixed encryption key to encrypt data; data privacy cannot be maintained in their scheme.

Perrig and Tygar [70] proposed several secure broadcast schemes suitable for wireless sensor networks. The computation overhead for their schemes is affordable for tiny sensor motes. They proposed a hashed key-chain scheme to sequentially generate encryption/decryption keys for sensor motes without notifying others. Przydatek, Song and Perrig [72] further extended these schemes and proposed a secure data aggregation scheme for sensor network. Their scheme provided an efficient random sampling mechanisms and interactive proofs to enable the querier to verify that the answer given by the aggregator is a good approximation of the true value, even when the aggregator and some sensor motes were compromised.

Chapter 3

Secure Data-Aggregation for Wireless Sensor Networks

Wireless Sensor Networks (*WSN*) have emerged as an important new area in wireless technology. A wireless sensor network [3] is a distributed system interacting with physical environment. It consists of *motes* equipped with task-specific sensors to measure the surrounding environment, e.g. temperature, movement, etc. It provides solutions to many challenging problems such as wildlife, battlefield, wildfire, or building safety monitoring. A key component in a WSN is the *sensor mote*, which contains a) a simple microprocessor, b) application-specific sensors, and c) a wireless transceiver. Each sensor mote is typically powered by batteries, making energy consumption an issue.

A major application for a wireless mote is to measure environmental values using embedded sensors, and transmit sensed data to a remote repository or a remote server. Because of limited transmission capabilities, this often requires multi-hop forwarding of messages, and is power consuming.

One specific power-saving mechanism used in wireless sensor networks is *data aggregation* [1][4][6][14][15][19][20]. Our paper proposes a novel method for eliminating duplicate encrypted data during aggregation without decryption. Data aggregation [24][37][40][41][48] [88][90][91] has been put forward as an essential paradigm in sensor networks. The aggregator uses specific functions, such as addition, subtraction or exclusive-or, to aggregate incoming readings, and only aggregated result are forwarded [51][52][54][57][74][76][77]. Therefore, communication overhead can be reduced by decreasing the number of transmitted packets [58][59][63][64][65] [66][72]. Without encryption, adversaries can monitor and inject false data into the network. Encryption can solve this problem, but how can we aggregate over encrypted data [18]?

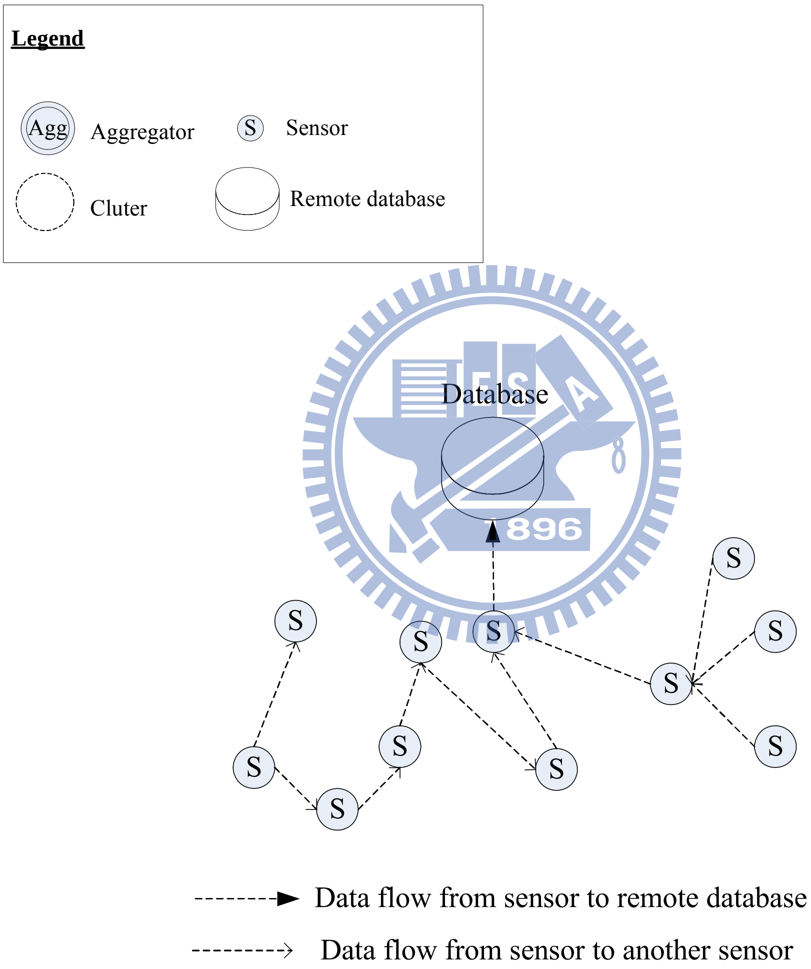

Legend
Agg
Aggregator
S
Sensor
Cluter
Remote database
Database
S
Data flow from sensor to remote database
Data flow from sensor to another sensor

Figure 2 Self-organized WSN architecture and its data aggregation flow

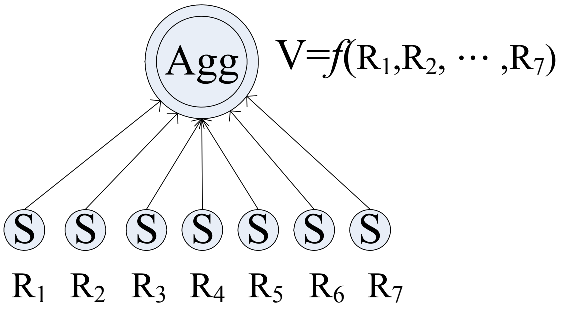


Figure 3 Conventional data aggregation process

We assume adversaries owned the following attacking capabilities.

- Adversaries can deploy sensors near existing sensors to determine their likely value.
- Adversaries can use common key encryption systems (which always encrypt common sensor data in the same way) to see when two readings are identical. By using nearby sensors under the adversaries' control, adversaries can conduct a *known-plaintext attack*.
- Adversaries can tamper with sensors to force them to predetermined values (such as heating a temperature sensor) and thus conduct a *chosen-plaintext attack*.
- Adversaries can inject false readings or resend logged readings from legitimate sensor motes to manipulate the data aggregation process, conducting a *man-in-the-middle attack*.

Table 1 presents encryption policies, possible attacks, and vulnerabilities in data aggregation schemes.

Encryption Policy	Possible Attacks	Secrecy	Privacy	Data Aggregation
Sensors transmit readings without encryption	man-in-the-middle	No	No	Generating wrong aggregated results
Sensors transmit encrypted readings with permanent keys	known-plaintext attack chosen-plaintext attack man-in-the-middle	Yes	No	Data aggregation cannot be achieved when data are encrypted unless the aggregator has encryption keys
Sensors transmit encrypted readings with dynamic keys	None of above	Yes	Yes	Data aggregation cannot be achieved when data are encrypted unless the aggregator has encryption keys

Table 1 Encryption policies, attacks and vulnerabilities in data aggregation schemes.

This paper proposes a new method for determining and eliminating duplicate data while protecting privacy (using encryption) without excessive key-management or power management issues. Our scheme has the following contributions. First, we provide a lightweight data aggregation mechanism which protects data when data are processed in aggregators. Aggregators can help to eliminate redundant data without decrypting data. Thus, aggregators do not need to spend extra power in data decryption, and more network lifetime can be guaranteed. Second, our proposed scheme is resilient to known-plaintext attacks, chosen-plaintext attacks, ciphertext-only attacks, and man-in-the-middle attacks. The rest of the paper is organized as follows: Section II provides background on related work. In Section III, we describe our system architecture and proposed aggregation protocol.Security analysis and performance evaluation are given in Section V, and Section IV offers conclusions and future directions.

Problem Statement and Proposed Data Aggregation

Data aggregation uses primitive functions, such as mean, average, addition, subtraction, and exclusive or to eliminate identical readings, and only unique results are be forwarded, reducing the cost of data transmission.

Figure 4 depicts an overview of data aggregation flow.

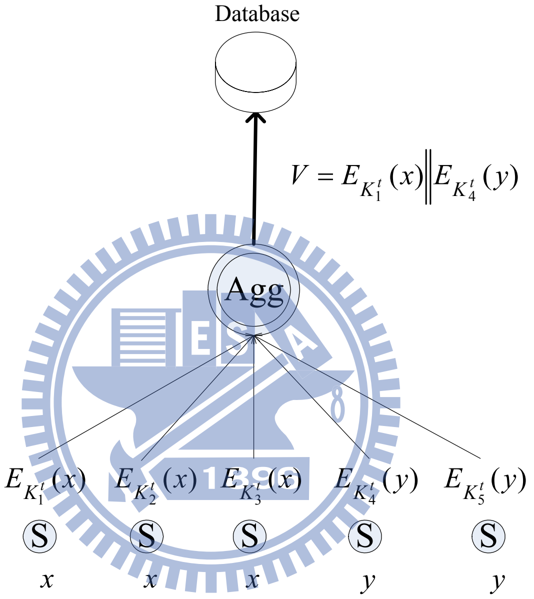


Figure 4 Encrypted-data aggregation

Proposed Data Aggregation Method

Architecture

There are two commonly used network topologies in sensor networks. One is the *self-organized sensor network*. A self-organized network is a multi-hop, temporary autonomous system composed of sensor motes with wireless transmission capability. It is easy to form such networks but every mote in such networks consumes significant amounts of power in data transmission as each node must spend power to transmit / forward data to other sensor nodes because of the dynamic network topology. The other network topology is the *clustered sensor network*. In this architecture, the entire network is partitioned into non-overlapping clusters. Each cluster has an *aggregator* (or *cluster head)* to receive readings from other sensor motes and to forward these readings to the remote server. To extend operation lifetime, we choose the clustered topology as our network architecture [66]. In a clustered sensor network, each mote temporarily belongs to a cluster, and sensors in this cluster will receive and forward data for sensors in the same cluster. Since a mote only transmits data for several motes instead of all motes, it can obviously reduce its power consumption for data transmission.

In a clustered WSN, we assume the network is divided into clusters. Each cluster owns an aggregator having a more powerful wireless transceiver that can transmit data directly to the backend server. In our framework, we also assume each sensor transmits data only to the aggregator; hence, each sensor mote can reduce overhead in forwarding data packets. We also assume sensor motes have no mobility, i.e. they are fixed in a position and will not be moved forever. The question of how to best deploy sensor motes and how to cluster these sensor motes is interesting to consider but it is beyond the scope of this paper.

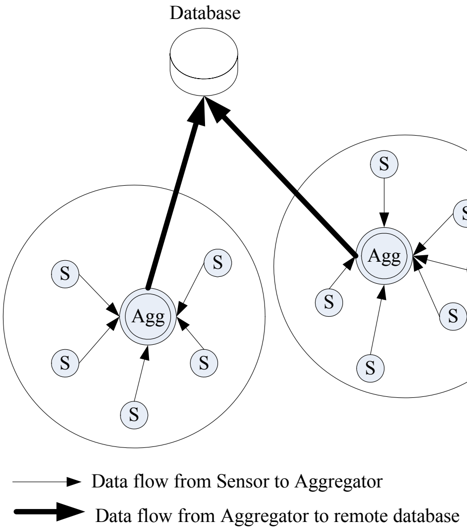

Database
S
S
Agg
S
S
S
S
S
S
S
S
Agg
S
S
Data flow from Sensor to Aggregator
Data flow from Aggregator to remote database

Figure 5 A clustered sensor network topology

Using a clustered network to reduce power consumption, we propose a data aggregation method which maintains both secrecy and privacy. In terms of secrecy, each sensor mote encrypts its reading and transmits the encrypted data to the aggregator. Adversaries will not be able to recognize what reading it is during data transmission. In terms of privacy, our design aims to eliminate redundant reading for data aggregation but this reading remains secret to the aggregator, i.e., the aggregator cannot know anything about these readings. Besides, our design can also prevent known-plaintext attacks, chosen-plaintext attacks and ciphertext-only attacks.

Here we list special notations we use in this paper:

Notations:

S_i : Sensor mote i

g : A one-way function having the following properties.

1. Given $x, y \in R^n$, $g(x \oplus y) = g(x) \oplus g(y) \in R^n$

2. Given $g(x)$, one cannot retrieve x in polynomial time

3. Given $x_1, x_2 \in R^n$, the condition $g(x_1) = g(x_2)$ is possible.

K_i^{EK}: An *encryption key* randomly generated by sensor mote *i*

K_i^{VK}: A *verification key* used to verify data from sensor mote *i*

System Setup

Before deploying a wireless sensor network, we have to set up three roles: the sensor mote, the aggregator, and the remote database.

1. The sensor mote: Each sensor mote *i* is assigned an one-way function *g*, and a *verification key* K_i^{VK}.
2. The aggregator: The aggregator is given the one-way function *g*, and all $K_i^{VK} \oplus K_{i+1}^{VK} \quad \forall i$. Hereafter, these keys are referred as *aggregation verification keys*.
3. The remote database: The remote database needs to decrypt aggregated data, and thus we need to store the one-way hash function *f*, the one-way function *g*, and all verification key K_i^{VK} for all *i*.

Necessary keys, identities, and functions are pre-distributed in the sensor mote, the aggregator, and the remote database before they are physically deployed and used. Table 2 lists all pre-installed elements in individual roles.

Role	Pre-installed elements
The sensor mote	SID_i, g, and K_i^{VK}
The aggregator	g, and $K_i^{VK} \oplus K_{i+1}^{VK}$ $\forall i$.
The remote server	g, and K_i^{VK} $\forall i$

Table 2 Pre-installed elements in three roles

Key pre-distribution is a scheme where keys are distributed among all sensor motes prior to deployment. Our proposed key pre-distribution scheme does not rely on prior deployment knowledge. Sensor motes are installed with random keys for encryption. These encryption keys have no mandatory relations between each other, and this makes system setup more flexible. Random keys can be generated by using random source of data, such as values based on CPU clock, radioactive decay, or atmospheric noise. The question of how to generating random numbers is interesting to consider but is beyond the scope of this paper.

Proposed Scheme

There are two phases in our proposed scheme: *data encryption phase* and *data aggregation phase*. The encryption phase provides a lightweight encryption algorithm that supports data aggregation property, and provides secrecy and privacy for data transmission. The data aggregation phase provides a method to eliminate redundant readings from sensor motes without decrypting them. Since the aggregator cannot decrypt incoming packets, the aggregator cannot know anything about the plaintext, and therefore more power can be saved.

■ *Data Encryption Phase*

Our encryption design aims to provide lightweight encryption overhead and secrecy while providing data aggregation property.

When a sensor mote *i* has a reading m_i and wishes to transmit this reading to the aggregator, it first randomly generates a new key K_i^{EK}, which will be used as the next-round encryption key. By using *g*, K_i^{EK}, and K_i^{VK}, the corresponding ciphertext $E_i(m_i)$ is defined in Eq(1).

$$E_i(m_i) = m_i \oplus g(K_i^{EK}) \oplus K_i^{EK} \| K_i^{EK} \oplus K_i^{VK}, \quad \text{Eq(1)}$$

Where $Length(m_i) = Length(K_i^{EK}) = Length(K_i^{VK})$ and $\|$ indicates data concatenation.

Our proposed scheme is very close to the one-time pad method [75] as each mote changes to a different key for encrypting data but provides more capabilities. It is obvious that the length of data is required to be at least as long as the length of encryption key in our proposed scheme. When the length of data is shorter than the length of the key, extra padding must be appended to the data so that the appended data can be encrypted. As the message m_i is xored with $g(K_i^{EK}) \oplus K_i^{EK}$, it does not matter if we pad random values or fixed values (e.g., all 0's or 1's). It does not reduce any security strength in our scheme.

Next, we will introduce how to find out redundant readings among these ciphertexts without decrypting them in our data aggregation phase.

- *Data Aggregation Phase*

Our data aggregation method provides a pair-wise method to identify if two readings are identical. Although the goal of our data aggregation scheme is to find redundant readings among *n* incoming encrypted packets in the aggregator, our aggregation scheme can be further extended by pairing off these *n* incoming encrypted packets. By iteratively performing pair-wise comparisons we can eliminate all redundant readings among them. If *n* same readings are encrypted and transmitted to the aggregator, the aggregator needs to check *n-1* times to verify these inputs and save *n-1* packet transmission. It needs computation overhead for data aggregation but saves more energy from fewer data transmissions.

In the following section, first we will introduce our approach to find redundant readings in two packets; then, we will introduce how to extend our approach to find redundant readings among *n* packets.

Assume sensor mote *i* and *j* sends two encrypted readings to the aggregator, and these encrypted readings can be expressed by the following equations:

$$E_i(m_i) = m_i \oplus g(K_i^{EK}) \oplus K_i^{EK} \| K_i^{EK} \oplus K_i^{VK} \,, \quad \text{Eq(2)}$$

$$E_j(m_j) = m_j \oplus g(K_j^{EK}) \oplus K_j^{EK} \| K_j^{EK} \oplus K_j^{VK} \,. \quad \text{Eq(3)}$$

First, the aggregator XOR the first parts of these two ciphertexts, and it can be expressed by the following equation:

$$m_i \oplus g(K_i^{EK}) \oplus K_i^{EK} \oplus m_j \oplus g(K_j^{EK}) \oplus K_j^{EK} \quad \text{Eq(4)}$$

Next, since the aggregator is pre-installed with $K_i^{VK} \oplus K_{i+1}^{VK}\ \forall i$, $K_i^{VK} \oplus K_j^{VK}$ can be obtained by $(K_i^{VK} \oplus K_{i+1}^{VK}) \oplus (K_{i+1}^{VK} \oplus K_{i+2}^{VK}) \oplus \cdots \oplus (K_{j-1}^{VK} \oplus K_j^{VK})$, the aggregator can XOR the last two parts of Eq(2) and Eq(3) to obtain:

$$\begin{aligned} & K_i^{EK} \oplus K_i^{VK} \oplus K_j^{EK} \oplus K_j^{VK} \oplus K_i^{VK} \oplus K_j^{VK} \\ & = K_i^{EK} \oplus K_j^{EK} \end{aligned} \qquad \text{Eq(5)}$$

It can be found that the aggregator can use $E_i(m_i)$ and $E_j(m_j)$ to retrieve $K_i^{EK} \oplus K_j^{EK}$, but cannot retrieve K_i^{EK} or K_j^{EK} separately; therefore, the aggregator cannot decrypt $E_i(m_i)$ and $E_j(m_j)$.

Next, we define a *check value* V, and V is calculated by XOR Eq(4), Eq(5) and $g(K_i^{EK} \oplus K_j^{EK})$. The check value is used to distinguish if two encrypted readings are the redundant in their plaintext format. As a result, the check value V can be expressed by the following equation:

$$\begin{aligned} V_{(i,j)} &= m_i \oplus g(K_i^{EK}) \oplus K_i^{EK} \oplus m_j \oplus g(K_j^{EK}) \\ &\oplus K_j^{EK} \oplus K_i^{EK} \oplus K_j^{EK} \oplus g(K_i^{EK} \oplus K_j^{EK}) \end{aligned} \qquad \text{Eq(6)}$$

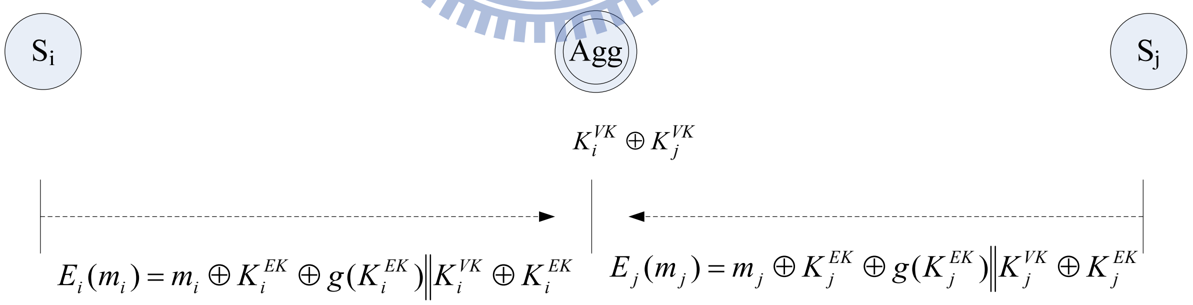
By using the properties of function g, Eq (6) can be further reduced to:

$$\begin{aligned} V_{(i,j)} &= m_i \oplus g(K_i^{EK}) \oplus K_i^{EK} \oplus m_j \oplus g(K_j^{EK}) \\ &\oplus K_j^{EK} \oplus K_i^{EK} \oplus K_j^{EK} \oplus g(K_i^{EK} \oplus K_j^{EK}) \\ &= m_i \oplus m_j \end{aligned} \qquad \text{Eq(7)}$$

It is easier observed that if m_i is equal m_j, $V_{(i,j)} = m_i \oplus m_j = 0$, and vice versa. We can formally describe $V_{(i,j)}$ by the following equations:

$$if \begin{cases} V_{(i,j)} = 0^n, then \quad m_i = m_j \\ V_{(i,j)} \neq 0^n, otherwise \end{cases} \qquad \text{Eq(8)}$$

Figure 6 depicts the data aggregation phase



$K_i^{VK} \oplus K_j^{VK}$

$E_i(m_i) = m_i \oplus K_i^{EK} \oplus g(K_i^{EK}) \| K_i^{VK} \oplus K_i^{EK}$

$E_j(m_j) = m_j \oplus K_j^{EK} \oplus g(K_j^{EK}) \| K_j^{VK} \oplus K_j^{EK}$

Step 1: $K_i^{EK} \oplus K_j^{EK} = K_i^{VK} \oplus K_j^{VK} \oplus K_i^{VK} \oplus K_i^{EK} \oplus K_j^{VK} \oplus K_j^{EK}$

Step 2: $V_{(i,j)} == m_i \oplus K_i^{EK} \oplus g(K_i^{EK}) \oplus m_j \oplus K_j^{EK} \oplus g(K_j^{EK}) \oplus K_i^{EK} \oplus K_j^{EK} \oplus g(K_i^{EK} \oplus K_j^{EK})$

$= m_i \oplus m_j$

Figure 6 Data Aggregation Phase

If these two readings are the same, the aggregator just needs to send either $E_i(m_i)$ or $E_j(m_j)$ to the remote server. If these two readings are different, the aggregator then sends $E_i(m_i) \| E_j(m_j)$ to the remote server. Since remote server is pre-installed with the verification key K_i^{VK}, the remote server therefore can use K_i^{VK} to obtain K_i^{EK} by: $K_i^{EK} = (K_i^{EK} \oplus K_i^{VK}) \oplus K_i^{VK}$. Then, the original data m_i can be recovered by: $m_i = (m_i \oplus g(K_i^{EK}) \oplus K_i^{EK}) \oplus g(K_i^{EK}) \oplus K_i^{EK}$. In above case, the aggregator only needs to examine two incoming ciphertexts, but in general cases, the aggregator usually receives more than two incoming ciphertexts. When the aggregator receive n (n>2) incoming ciphertexts ($E_1, E_2, \cdots, E_n$), our proposed scheme can be easily extended. First, we group these ciphertexts into pairs, i.e. $(E_i, E_j)\forall i$. Then, we can repeat above steps to generate their check value V. Next, we can use V to check if E_i has the same reading with E_j. Finally, if $V_{(1,2)} = V_{(2,3)} = \cdots = V_{(n-1,n)}$, then we can conclude that $E_1, E_2, \cdots, E_n$ has the same reading. Figure 7 depicts necessary comparisons for data aggregation when $n = 5$. It can be observed that these comparisons can be viewed as all edges in a complete graph, and we will discuss this property in next section.

Preliminaries: When the aggregator receives n encrypted readings, the minimum number of comparisons is n-1 under the condition that all these readings (when unencrypted) are the same. The maximum number of comparisons is $\frac{n(n-1)}{2}$ when all these readings (when unencrypted) are totally different from each other.

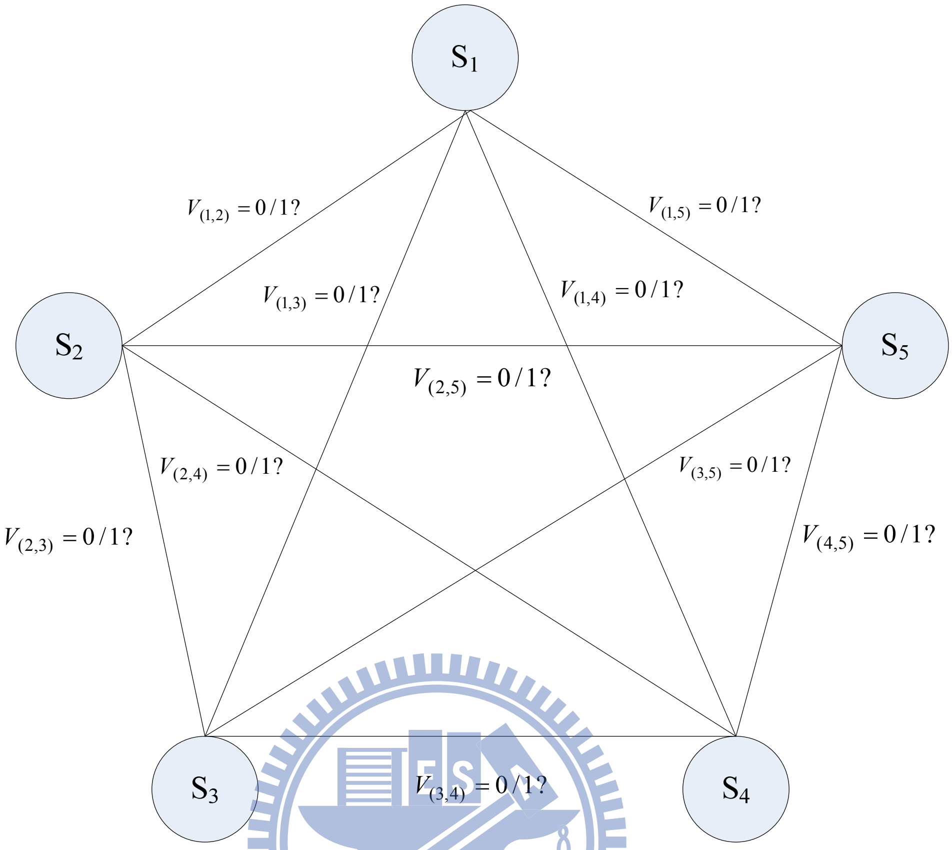


Figure 7 Data aggregation verification steps for *n*=5.

Threat Models

The goals of the adversaries are to read, insert, and even modify sensor readings. We consider several possible threats, classified according to the capabilities of the adversaries.

Known-plaintext Attacks.

To implement known-plaintext attacks, no capabilities are need except the ability to deploy malicious sensors close to legitimate sensors. In this scenario, an adversary can

- Collect all readings from all sensors, calculated aggregated values, know their routing paths, and inject wrong readings or aggregated values to the network.
- Collect abundant encrypted readings to enhance the compromise of encryption keys.

In practice, known-plaintext attacks can be easily achieved by deploying same sensor very close to legitimate sensors. The goal of these attacks is merely to read readings and to record corresponding responses of a sensor mote.

Chosen-plaintext Attacks.

- Adjust the sensors by changing physical conditions, such as temperature or moisture.
- Log all plaintext-ciphertext mappings without knowing what the encryption keys are.

In practice, adversaries can take some physical methods to adjust the sensing environment in order to make sensor motes generate false readings the adversaries desired. For example, adversaries can use heaters to raise the temperature to a certain degree, and temperature sensors will send the false temperature readings making the aggregators generate incorrect results.

Man-in-the-middle Attacks.

- Read, insert, or modify messages between sensor motes.
- Inject false readings or resend logged readings on behalf of legitimate sensor motes to malfunction data aggregation.

Significantly, we assume that an adversary cannot retrieve encryption keys from a sensor mote by physically compromising it. Otherwise, there will be no security at all.

Security Analysis and Performance Evaluation

In this section, we evaluate our proposed scheme according to two aspects: theoretical and practical. In theoretical aspect, we use random oracle model to justify our protocol is secure in terms of provable security. We firstly built an ideal random oracle model and show that our proposed encryption algorithm is an implementation of the ideal random oracle. Then, we use the random oracle model to justify that it can resist know-ciphertext attacks. In practical aspect, we estimate necessary time for compromising our proposed scheme using different key lengths. The result shows that using encryption keys longer than 80 bits would be considerable secure enough even if the adversary uses 1,000,000 4GHz PCs running simultaneously to compromise our scheme. Then, we show that our proposed scheme can resist known-plaintext attacks, chose-plaintext attacks, and know-ciphertext attacks.

Before we proceed to theoretical proof, we first describe the security requirement specifying the adversary's abilities and when the latter is considered successful. The abilities and disabilities of the adversary include:

- The adversary has an arbitrary polynomial-time computation power
- The adversary can eavesdrop on messages in the air
- The adversary can know the original readings of any sensor
- The adversary cannot access the encryption keys

An attack is considered to be successful if the adversary can compromise the encryption keys. In terms of system security, we adopt the idea in [16]. A system is considered secure if any adversary with the given abilities has only a negligible probability of success.

A random oracle is a theoretical black box that replies to queries with random response chosen uniformly in its output domain. A methodology for designing a cryptographic protocol can be divided into two steps. In first step, one designs an ideal system in which all participants as well as adversaries have oracle access to a truly random function, and proves the security of the ideal system. In second step, we replace the random oracle by a "good cryptographic hashing function". We can therefore obtain an implementation of the ideal system in a real-word where random oracles do not exist. This methodology is referred to as the random oracle methodology.

Before we build our ideal system, we first describe the notion.

$\{0,1\}^*$: the space of finite binary strings

$\{0,1\}^\infty$: the space of infinite binary strings

$G:\{0,1\}^* \rightarrow \{0,1\}^\infty$: a random generator

f : a trapdoor permutation with inverse f^{-1}

k : the security parameter

$H:\{0,1\}^* \rightarrow \{0,1\}^k$: a random has function

$G(r) \oplus x$: the bitwise XOR of x with the first $|x|$ bits of the output of $G(r)$

Preliminaries

Definition. A function $\varepsilon(k)$ is negligible if for every c there exists a k_c satisfying $\varepsilon(k) \le k^{-c}$ for every $k \ge k^{-c}$.

Definition. If A_P is a probabilistic algorithm, then for any inputs $m_1, m_2, \cdots$, $A_P(m_1, m_2, \cdots)$ is the probability space which to the sting σ assigns the probability that A_P outputs σ. For probabilistic spaces $S, T, \cdots$, $P_r[x \leftarrow S; y \leftarrow T; \cdots : p(x, y, \cdots)]$ denotes the probability that the predicate $p(x, y, \cdots)$ is true after the execution of the algorithms $x \leftarrow S$, $y \leftarrow T$, etc.

Definition. A random oracle R is a map from $\{0,1\}^*$ to $\{0,1\}^\infty$ chosen by selecting each bit of $R(x)$ uniformly for every x.

Without lost of generosity, our proposed scheme can be formulated as the following oracle:

$$E_r^G(m) = m \oplus G(r) \| f(r) \ldots \text{Eq(9)}$$

Known-Plaintext Security

For known-plaintext attacks, the adversary knows some m, and P_r[The attacker successfully guesses $G(r_1)$] can be described as:

$$P_r[r_1 \leftarrow G(r)] = \frac{1}{2^{|r_1|}} \approx 0\text{, when } r_1 \text{ is large enough.}$$

We suggest that $|r_1| \geq 88$ is adequate and mathematical induction will be given later.

Chosen-Plaintext Security

We adapt the notion of CP-adversary (chosen-plaintext adversary) in [11] to the random oracle model. A CP-adversary A is a pair of non-uniform polynomial algorithms (F, A_1), each with access to an oracle. For an encryption algorithm ϑ to be secure, it requires that

$$\begin{aligned}&P[\text{Chosen-PlaintextFails}] = P_r[R \leftarrow 2^{\infty}; (E, D) \leftarrow \vartheta(1^k);\\&(m_0, m_1) \leftarrow F^R(E); b \leftarrow \{0,1\}; \sigma \leftarrow E^R(m_b)\\&: A_1^R(E, m_0, m_1, \alpha) = b] \leq 0.5 + k^{-w(1)}\end{aligned} \quad . \qquad \text{Eq(10)}$$

Proof: The proof is by contradiction. Let $A = (F, A_1)$ be an adversary that defeats our protocol. Often, the adversary gains advantage $\lambda(k)$ for some inverse polynomial λ. We construct an algorithm $M(f, d, y)$ that, when $(f, f^{-1}, d) \leftarrow \vartheta(1^k)$; $r \leftarrow d(1^k); y \leftarrow f(r)$, manages to compute $f^{-1}(y)$. It simulates the oracle G and samples $(m_0, m_1) \leftarrow F^G(E)$. If G is asked an r such that $f(r) = y$, then M outputs r and halts; otherwise, the $F(E)$ terminates and M chooses $\alpha \leftarrow y \| s$ for $s \leftarrow \{0,1\}^{|m_0|}$. Then M simulates $A_1^G(E, m_0, m_1, \alpha)$, watching the oracle queries that A_1 makes to see if there is any oracle query *r* for which $f(r) = y$. Let A_k be the event that A_1 does not ask for the image of *G* at *r*. It satisfies that

$$1/2 + \lambda(k) = P_r[\text{A succeeds} \mid A_K] \cdot P_r[A_k] + P_r[\text{A succeeds} \mid \overline{A_k}] \cdot P_r[\overline{A_k}].$$

Thus, eq. (10) is satisfied.

Chosen-Ciphertext Security

The chosen-ciphertext attack is defined as: the adversary can adaptively choose ciphertexts and access to the decryption algorithm to get the corresponding plaintexts. Though it is usually occurred in asymmetric cryptographic systems, it can also be happened in our scheme as the adversary can know both ciphertexts and plaintexts (by using same sensors) in the same time. We adapt the definition of [11] and [73] to the random oracle [16] setting. An RS-adversary ("Rackoff-Simon adversary") A is a pair of non-uniform algorithms $A=(F,A_1)$, each with access to an oracle R and a black box implementation of D^R. The algorithm F is used to generate two messages m_0 and m_1 such that if A_1 is given the encryption α, A_1 won't be able to guess well whether α comes m_0 or m_1. Formally, an encryption scheme ϑ is secure against RS-attack if the following equation is satisfied:

$$\begin{aligned} & P[\text{Chosen - Ciphertext Fails}] = \\ & P_r[R \leftarrow 2^{\infty}; (E,D) \leftarrow \vartheta(1^k); (m_0,m_1) \\ & \leftarrow F^{R,D^R}(E); b \leftarrow \{0,1\}; \alpha \leftarrow E^R(m_b) \\ & : A_1^{R,D^R}(E,m_0,m_1,\alpha) = b] \leq 0.5 + k^{-w(1)} \end{aligned} \qquad \text{Eq(11)}$$

Proof: To see our scheme is secure against chosen ciphertext attacks, we prove the above equation is satisfied. Let A_k denotes the event that $a \| b \leftarrow F(E)$, for some a and b. Let $A=(F,A_1)$ be an RS-adversary that succeeds with probability $\frac{1}{2}+\lambda(k)$ for some non-negligible function $\lambda(k)$. The adversary A can make some oracle call of $G(r_1)$ or $H(G(r_1))$. Let L_k denotes the event that A_1 asked $D^{G,H}$ some queries where $a=m \oplus f^{-1}(r_1) \oplus H(f^{-1}(r_1))$, but A_1 never asked its *H*-oracle for $H(f^{-1}(r_1))$. Let $n(k)$ denotes the total number of oracle queries made. It is easy to see that $Pr[L_k] \leq n(k)2^{-k}$ and $Pr[\text{A succeeds}\, \overline{L}_K \cap \overline{A}_K] = 0.5$ according to [11].

Thus $P_r[\text{A succeeds}] = P_r[\text{Choosen - Cipher Attack succeeds}] = \frac{1}{2}+\lambda(k)$ is bounded above by

$$\begin{aligned}&P_r[\text{A succeeds } L_K]P_r[L_k]\\&+P_r[\text{A succeeds}\,|\,\overline{L}_k \cap A_k]P_r[\,\overline{L}_k \cap A_k]\\&+P_r[\text{A succeeds}\,|\,\overline{L}_k \cap \overline{A}_k]P_r[\overline{L}_k \cap \overline{A}_k]\cdot\\&\le n(k)2^{-k}+P_r[A_k]+\frac{1}{2}\end{aligned}$$

Therefore, our proposed scheme satisfies eq. (11), and is chosen-ciphertext-attack resistant.

In practice aspect, we evaluate the difficulties to brute force our proposed scheme. To brute force our proposed scheme, first the adversary need to spend time generating all possible keys and test the result with every possible key. We assume that the adversary can generate an encryption key and test the result in one duty cycle, our proposed scheme uses σ-bit keys to encrypt data, and the adversary uses a η G-Hz PC to brute force our propose scheme. To completely test all possibilities by exhaustive search, the adversary would need to spend

$$\frac{2^{\sigma}\,(cycles)}{\eta * 10{,}000{,}000 * 86400 * 365}$$

years to compromise our scheme.

Assume the adversary uses a 4G-Hz PC to brute force our scheme which uses a 64-bit encryption key, the adversary needs to generate all 2^{64} keys and uses these keys to test the result. If we assume that the adversary can test our encryption scheme within one duty cycle, the total computation time to test all 2^{64} keys is:

$$2^{64}/4G/86400/365 \approx 14{,}624 \text{ years.}$$

However, the adversary can use more PCs simultaneously to compromise our algorithm. If the adversary uses 1,000,000 PCs running simultaneously to compromise our scheme, the total computation time to test all conditions is

$$2^{64}/4G/86400/365/1M \approx 0.01 \text{ year.}$$

In this case, it takes about 3~4 days to compromise our scheme which is unacceptably insecure. Table 3 lists estimated time to brute force our proposed scheme with different key lengths. To maintain acceptable security while using minimal key length, we suggest use 80-bit keys to encrypt data as the adversaries need about 958 years to compromise our scheme even if they use 1,000,000 PCs to attack our scheme in parallel.

Key Length	One 4GHz PC	10000 4GHz PCs	100000 4GHz PCs	1M 4GHz PCs
64 bits	14624	1.5	0.15	0.01
72 bits	374363	374.4	37.4	3.74
80 bits	958369660	95837	9583.7	958.37
88 bits	2.45E+11	24534263	2453426.3	245342.6

Table 3 Estimated time (years) to brute force our proposed scheme with different key lengths

Moreover, using longer encryption keys can dramatically increase difficulties to compromise our scheme as it exponentially expend the key space which makes adversaries spend more time to brute force the proposed scheme. Figure 8 illustrate the growth rate of key size (2^{σ}) and the growth rate (η) of PCs.

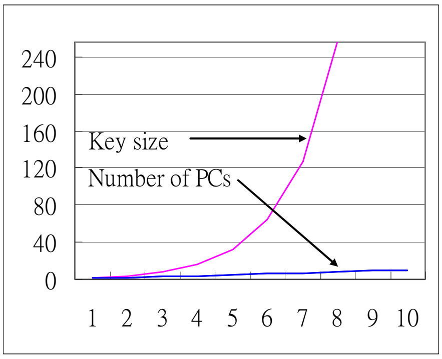


Figure 8 The growth rate of key size (2^{σ}) and the growth rate (η) of PCs

Assume the adversary know sensor reading m and corresponding ciphertext $E_i(m_i) = m_i \oplus g(K_i^{EK}) \oplus K_i^{EK} \| K_i^{EK} \oplus K_i^{VK}$, the adversary can therefore know $g(K_i^{EK}) \oplus K_i^{EK}$ and $K_i^{EK} \oplus X_i^{VK}$. Without knowing K_i^{VK} in advance, the adversary cannot compromise K_i^{EK} . Furthermore, since the encryption keys will be arbitrarily changed, our scheme can hence resist known-plaintext attacks. Even adversary can generate designate data m to confuse sensor motes, still the adversary cannot learn anything about the encryption keys. Therefore, our scheme can resist know-ciphertext attacks and chosen-ciphertext attacks.

One workload we have to pay is the number of comparisons it takes to verify encrypted-data from n motes. Our proposed scheme can reduce the number of comparisons as it has transitive property. The transitive property is described as: Given $E_h(m_h)$, $E_i(m_i)$, and $E_j(m_j)$, if $V_{(h,i)} = 0$ and $V_{(i,j)} = 0$, then $V_{(h,j)} = 0$. This is pretty simple to prove. If $V_{(h,i)} = 0$ and $V_{(i,j)} = 0$, then $m_h = m_i$ and $m_i = m_j$ It can therefore be easily seen that $m_h = m_i = m_j$.

With this transitive property, if all readings are the same, the minimum number comparisons for verifying data from n sensor motes is $n-1$. And, according to figure 7, the maximum number comparisons for verifying data from n sensor motes is equal to the number of edges in a n-complete graph which is $\frac{n(n-1)}{2}$. It is shown in Figure 9 that our computation bound is limited between $O(n)$ and $O(n^2)$, and this can be affordable for off-the-shelf sensor platforms.

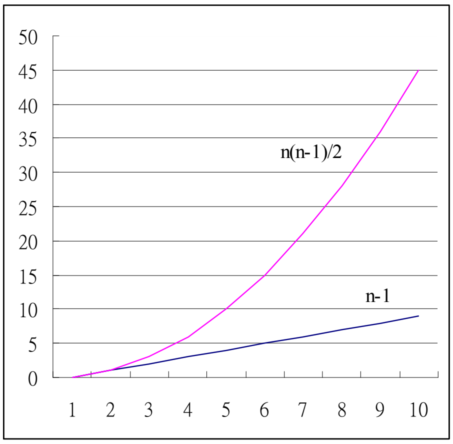


Figure 9 The number of comparisons for verifying *n* encrypted-data.

In comparison with other schemes, our encryption algorithm uses XOR and a hash function. Our encryption algorithm is more lightweight. Our proposed encryption algorithm changes its encryption key whenever there's a reading that needs to be transmitted. This makes our scheme more feasible for wireless sensor networks. Table 4 lists the differences between our scheme and other schemes.

	Our proposed scheme	Flooding-base Scheme	Privacy homomorphism-based scheme
Encryption	Lightweight	Heavyweight	Heavyweight
Encryption Key	Easy to change, and always changes	Only one encryption key, and is hard to change	Only one encryption key, and is hard to change
Decryption (in aggregator)	NO	YES	YES
Aggregated Result	Only one data	Many redundant data	Only one data

Table 4 Performance evaluations compared with other schemes

Key Compromise. One major issue in our scheme is the key compromise problem. As the aggregator stored $K_i^{VK} \oplus K_{i+1}^{VK} \ \forall i$, once an encryption key K_i^{VK} is been compromised, all other encryption key K_j^{VK} $\forall j \neq i$ will be compromised. Therefore, the aggregator must have stronger security protection than sensor motes. One way to enhance the hardware security strength in the aggregator is to install a TPM (Trust Platform Module) chip inside the sensor mote, and all paired encryption keys $K_i^{VK} \oplus K_{i+1}^{VK}$ $\forall i$ are stored inside TPM. It can significantly reduce the possibility that adversaries compromise the aggregators.

Data Size Variation. Here we discuss the storage requirement when the length of data is increased. When the length of data is increased, the encryption key must be increased correspondingly. Assume the length of data is increased by l', the length of key as well will increase l' bits. As each sensor mote stores K_i^{VK} only, it requires more l' bits to store the encryption key. For the aggregator, as the aggregator stores all paired encryption keys $K_i^{VK} \oplus K_{i+1}^{VK}$ $\forall i$, it requires more $\theta \cdot l'$ bits where θ is the number of sensor motes in the cluster. It can be seen that when the length of data is increased linearly, the storage requirement for storing keys is also increased linearly.

Efficiency. Here we discuss the efficiency caused from our proposed scheme. Our proposed saves power by eliminating redundant packets. Thus, the more packets are eliminated, the more power can be saved. As we mentioned earlier, the minimum number of comparisons is *n*-1 under the condition that all these readings (when unencrypted) are the same, amd the maximum number of comparisons is $\frac{n(n-1)}{2}$. For the best case, it reduces (*n-1*) packet transmissions. For the worst case, it does not reduce any packet transmission overhead. For average case, assume that there totally n packets and m of them are the same, the number of comparisons is $(m-1)+(n-m)(n-m-1)/2$. It reduces *n-m* packets in average case. Table 5 lists the efficiency comparisons for the best, average, and the worst cases.

	Best Case	Average Case	Worst Case
Number of comparison	*n-1*	$(m-1)+(n-m)(n-m-1)/2$	$n(n-1)/2$
Packet eliminated	*n-1*	*n-m*	0

Table 5 Efficiency comparisons for the best, average, and the worst case

Summary

In this chapter, we proposed a secure encrypted-data aggregation scheme for wireless sensor networks. Our scheme has the following enhancements: 1) the aggregator does not need to decrypt its received encrypted-data to verify if these data are the same; no extra power are wasted in data decryption, 2) the aggregator does not have decryption keys and therefore cannot know anything about the data, and 3) our proposed scheme uses random keys to encrypt data; this property makes our scheme resilient to known-plaintext attacks, chosen-plaintext attacks, ciphertext-only attacks, and man-in-the-middle attacks. Moreover, compared with conventional PH-based data aggregation schemes, received data can be recovered and decrypted to be further analyzed. Our proposed scheme provides secrecy and privacy in the sense that each sensor mote randomly generates a new encryption key each time providing semantic security for data encryption phase proposed data aggregation, and the intermediate aggregators cannot decrypt these encrypted-data. Aiming at secrecy and privacy, our proposed scheme is resilient to several attacks in sensor networks, and makes data aggregation more practical in these environments.

Our proposed scheme extends one-time pad to provide a secure encrypted-data aggregation paradigm for wireless sensor. Though it supports secrecy and privacy, our scheme provides only equality check. More general mathematical operations, such as addition, subtraction, and so on, should be further investigated under the same condition: the encryption keys are always changing and the aggregator cannot decrypt data through it. Except these mathematical operands, operands for strings, such as finding substring, should also be provided.

Currently, our scheme is workable in a one-level clustered network environment, i.e. the aggregator can one-hop to the base station. However, in real deployment, it is usually not the case. Our future work toward this problem is to extend our scheme to multi-level cluster environment. Another problem in our scheme is that our experimental sensor motes must be fixed to a cluster and can no longer be moved to another cluster. We will also address this issue in our future work.

For key management, our proposed scheme pre-installs keys for verification and data aggregation in the aggregator before deployment. This limits the flexibility of system deployment and aggregation. In future work, we expect to modify our key management method so that these keys will not be stored in aggregators in advance but will be exchanged and retrieved when necessary. We also look forward to extending privacy homomorphism functions to support dynamic key management to bring more flexibility in data aggregation.

Our protocol uses only XOR operations and an irreversible hash function to encrypt data. The security strength is not as strong as block cipher encryption algorithms, such as AES, DES, etc. We also expect to extend our scheme to adopt block-cipher encryption algorithms to provide higher security strength for aggregation.

Chapter 4

Authentication and Secret Search Mechanisms for RFID-Aware Wireless Sensor Networks

Radio-frequency identification (RFID) has been widely used in various applications. An RFID tag is a low-cost device with limited data storage space. An identification number (*ID*) is assigned to each tag for identification, and tagging specific targets with RFID tags allows for individualization and recognition of each target by the attached ID. Through the wireless interface, each tag can report data when queried over radio by a RFID reader. The RFID reader can execute read, write, and overwrite commands on each tag over the wireless interface. However, RFID readers can only recognize tags in close proximity; a data tag that is out of range cannot be read by a reader. This distance limitation severely restricts RFID deployment. Despite equipping readers and tags with longer-range wireless communication capability, RFID readers still have difficulties in tracking or monitoring tags at a distance. To solve this distance limitation problem, a wireless sensor network can act as a bridge between the tags and the readers when tracking or monitoring remote targets.

A wireless sensor network [33][36][56][71] consists of groups of sensor nodes connected by wireless links that perform sensing tasks, such as detecting changes in temperature, pressure, etc. These sensors are employed for specialized tasks like surveillance and security, environmental monitoring, location tracking, warfare, and health care.

Sensor nodes can communicate with RFID tags through the wireless interface. Since sensor nodes are cheap, they can be widely deployed to monitor every target, allowing readers to find targets at a distance. Although the use of sensor nodes solves the distance limitation problem, it introduces additional security challenges.

Examples of such a network composed of sensors and RFID tags include: the management of medical waste disposal, the management of blood storage bag in hospitals, the management of books in libraries, etc. In the aforementioned environment, the collaboration of sensor nodes and tags can form a dynamic, distributed database, where each sensor node contains a tiny database that tracks the data stored in RFIDs. Since sensor nodes are widely deployed, they form a group of distinctive databases. Simply encrypting the database ensures data security; however, it raises the issue of searching secrets.

Searching unencrypted data in a conventional remote database is relatively easy, but it leads to a serious problem: these queries may leak private information during transmission. One possible solution to prevent data leakage is to encrypt the original data and place it in a remote database. However, conventional cryptosystems and authentication schemes incur high computation cost, and may not be feasible for a network composed of wireless sensor nodes and RFIDs. Redesigning conventional cryptosystems and authentication schemes is a challenging task.

Due to the limited resource and computation capability of sensor nodes, it is desirable to search encrypted data without the need to decrypt it. In a typical application, sensor nodes encrypt data to improve security against intrusions. To search data, a sensor node must first decrypt the data, a process which usually causes significant delay. Moreover, computation-limited, low cost devices, such as sensor nodes and RFID tags, leave the decrypted data vulnerable to disclosure. In such an exposed environment, it is desirable to develop a new secret search method that performs secret search directly on ciphertexts without the need to decrypt them, thereby preserving secrecy and avoiding decryption delay.

For secret search in wireless sensor networks, the following requirements are considered important:

5) *Secrecy*: Storing data in an encrypted form helps retain its confidentiality. Because sensors are vulnerable, computation-limited, and low cost devices, allowing sensors to decrypt data to perform a search results in unnecessary risk of disclosure. Thus, sensors must execute a secret search directly on ciphertext, rather than plaintext. Furthermore, data transmitted over a wireless interface is susceptible to exposure. Therefore, sensors must only transmit encrypted data. In summary, the data must remain in an encrypted form and should not be decrypted unless necessary to minimize the possibility of disclosure.

6) *Authentication*: Since the network obtains data from a large number of sensors or tags, attackers can easily acquire readers with the same specifications to extract data stored in the tags. Therefore, both the reader and the tag need to verify the authenticity of its communication counterpart before executing read or write operations.

7) *Integrity*: Assuring data integrity prevents attackers from using unauthorized readers to modify or inject data into databases. Readers or tags must verify data integrity upon receipt of data.

8) *Performance*: Requiring a sensor node to decrypt data before searches causes significant and unnecessary delay. Also, the limited computation capabilities of sensor nodes and tags hinder them from performing complex operations, such as encryption and exponential calculations. Therefore, all operations must be redesigned to fit their computation capabilities.

Our contribution is threefold. First, we propose an architecture consisting of passive RFIDs and RFID-aware sensor networks (*ARIES*). This architecture extends RFID's capabilities through a wireless sensor network by utilizing sensor nodes to locate targets at a distance. Second, we design a private mutual authentication protocol (*AMULET*) which is feasible for RFIDs and sensor nodes, and reduces the cost of re-authentication. Third, we present a secret search protocol (*SSP*) that enables readers to perform searches over encrypted data, allowing data to remain encrypted during transmission or at vulnerable locations. By only using one-way hash functions, pseudo random number generation functions, and XOR operations, SSP accommodates the resource limitations of both tags and sensors. In addition, SSP can solve the problem that same plaintexts at different places will be encrypted into the same ciphertexts.

The rest of this paper is organized as follows. Section 2 introduces the proposed *ARIES* architecture for RFID and sensor networks, while section 3 presents our *AMULET* mutual authentication protocol for readers and tags. The SSP secret search protocol is presented to query encrypted data in section 4, and more advanced properties are discussed in section 5. Finally, section 6 provides security proof of the proposed schemes, and section 7 concludes our work.

ARIES

In this section, we introduce our system architecture, participating roles and their setups when deploying such a network. The notation we used are listed below:

ID_j : The identity of RFID tag *j*

$S_{i,j}$: A secret key shared by reader *i* and tag *j*

EK_i : A symmetric encryption key used by reader *i*

R : Nonce

$f:\{0,1\}^* \to \{0,1\}^\delta$ A pseudo random number generating function

$H:\{0,1\}^* \to \{0,1\}^\delta$: An one-way hash function

Motivated by the distance limitation problem of RFID readers, we propose an **AR**chitecture of RF**I**Ds and RFID-aware s**E**nsor network**S** (*ARIES*). Three roles are involved in our proposed system: RFID reader (abbr. as reader in what follows), RFID tag (abbr. as tag in what follows), and RFID-aware Sensor node (abbr. as sensor node in what follows). Since tags (tags on moveable targets) may be quite far away from readers, sensor nodes in our architecture is used as the gap between readers and tags by transmitting commands from reader to tag or sending tag data to readers, allowing readers to trace any tag located far away.

Although an *RFID reader* is called a reader by convention, it also has writing capability. Thus, a reader can perform read, write, and overwrite operations on RFID tags through the wireless interface. In our system, readers have access to a shared database storing all authorized *ID*s. To construct a secure channel between readers and tags, the readers share a unique secret key s with each tag. While readers save all tag pairs (s, ID) in the shared database, each tag stores its individual secret key s locally. Additionally, each reader possesses a unique encryption key EK_i to encrypt data, which it saves locally and remotely (on the shared database). EK_i can be used to verify the ownership of encrypted data.

An *RFID tag* is a small, thin, readable, and writeable device that can store limited data. Embedded with a transceiver, each tag can communicate via wireless channels with other devices, such as readers or sensor nodes. Because tags have limited computation capability, intensive operations, such as encryption, are impractical for tags. Therefore, we will introduce new methods supporting lightweight authentication in section 3.

An *RFID-aware sensor node* is a tiny device capable of detecting RFID tags. It is also outfitted with a transceiver to communicate with readers and tags through a wireless interface. Like tags, sensor nodes are cheap and widely dispersible.

As mentioned earlier, sensor nodes can compensate for the distance limitation of RFID readers. To reach readers, we assume that the sensor network allows for multi-hop communication. Furthermore, readers, tags, and sensor nodes can maintain secure communications. However, we do not introduce a security scheme between readers and sensors, tags and sensors, or readers and tags. Instead, we merely indicate that secure channels exist through shared secret keys or pre-distributed verifiable key pairs..

To prevent replay attacks, we assume that each reader, tag, and sensor node has a synchronized timer, allowing them to verify that an authentication process has not expired. Though it is unpractical to put a timer into a tag, the tag yet can have a timer virtually by neighboring sensor nodes periodically sending their timer readings. Our system merely requires loose time synchronization because of infrequent authentication. Because past researchers have investigated time synchronization [42][38]. We do not address this issue here. Another consideration is tags can be compromised and send bogus timestamps. Several existing protocols using majority vote can successfully solve this. We do not intend to discuss this as it is beyond our scope.

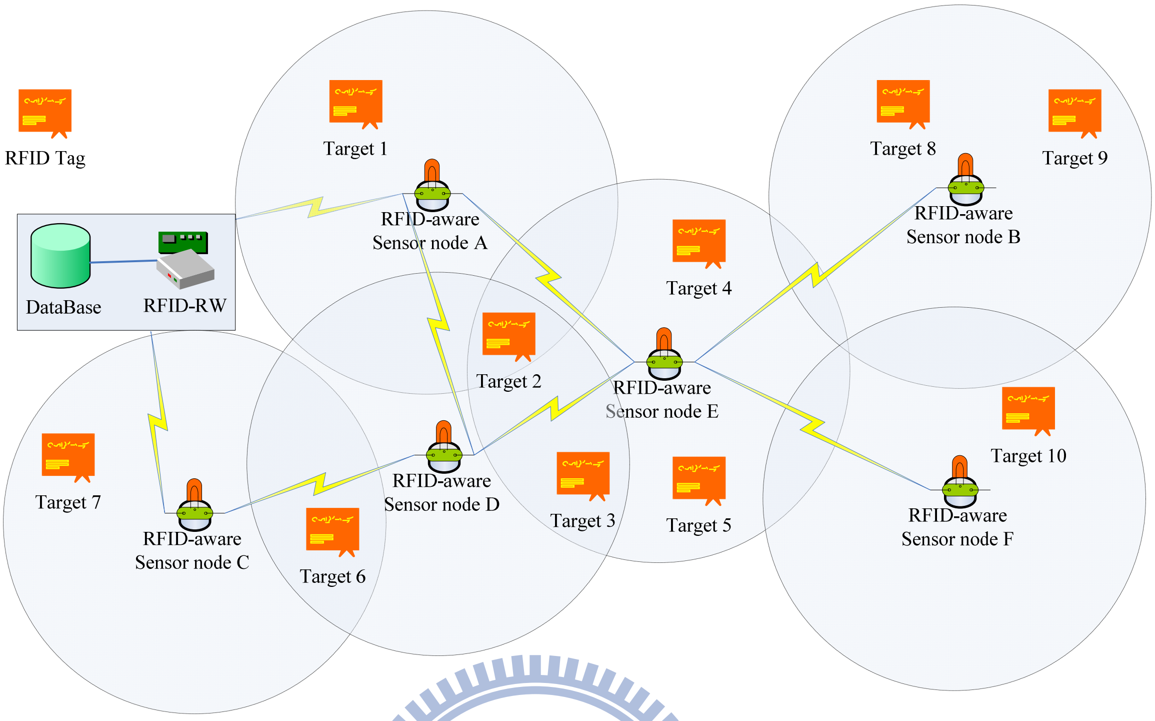


Figure 10 ARIES architecture.

In our architecture, readers can request data from faraway tags via sensor nodes. Figure 10 depicts the RFID readers, RFID tags, and RFID-aware wireless sensor nodes that make up the *ARIES* architecture. The sensor node collects data from tags in its vicinity and stores it in a local tiny database, where each attribute represents characteristics of the target. Table 6 represents a sample distributed tiny database. The (Target ID, Sensor ID) pair indicates the Target ID which is detect by Sensor ID. These pairs roughly reveal the geographical information about all targets. The ($Attr_1, Attr_2 \cdots, Attr_n$)-tuple manifests the data stored in the target. This distributed database can be used not only to search specific event with some user-interesting values, but also can be used to track the location of every sensor node and target.

Target ID	Sensor ID	Attr 1	Attr 2	------------	Attr N
ID_1	$Sensor \quad A$	$Attr(A1)$	$Attr(A2)$	------------	$Attr(An)$
ID_2	$Sensor \quad A$	$Attr(A1)$	$Attr(A2)$	------------	$Attr(An)$
ID_2	$Sensor \quad B$	$Attr(B1)$	$Attr(B2)$	------------	$Attr(Bn)$
			⋮		
ID_7	$Sensor \quad K$	$Attr(K1)$	$Attr(K2)$	------------	$Attr(Kn)$

Table 6 Distributed tiny database.

This architecture is workable for passive RFID tags needs only RF signals to charge and becomes active. No extra power waste will be needed, and thus each sensor nodes can reduce unnecessary power consumption in reading or writing RFID tags.

AMULET

In this section, we are going to introduce a lightweight authentication mechanism between readers and tags. Since all query actions are initiated by readers, the sensor nodes are merely viewed as generic routers and used only to forward these queries to tags. Therefore, our scheme only focuses on building low-computation authentication between readers and tags.

AMULET involves two phases. The setting phase and authentication phase. The setting phase initializes necessary components, such as IDs and keys, which will be used for authentication. The authentication phase performs mutual authentication for sensor nodes and tags.

Setting Phase

In AMULET, we need to setup two components: the tag and the reader. For each tag, it is assigned with an unique identification, ID_i, and a unique secret, s_i. All pairs of (ID_i, s_i) are stored in the reader's database, that will be used in authentication phase. These settings are performed in factor or library before deploying them into real work.

Since the passive tag has limited computation capabilities, it cannot afford complicated operations. It is reasonable assume that in our paradigm the tag can afford lightweight operations including XOR and a pseudo random number generating function f [66]. The pseudo random number generating function f also be stored in both the tag and the reader.

Authentication Phase

Authentication is the first step in building a trust relationship between readers and tags. Since readers and tags rely on wireless communication, attackers may eavesdrop on transmitted data and extract passwords. Previous research characterizes RFID communication as asymmetrical in signal strength. That is, attackers have an easier time listening in on signals from reader to tag than on data from tag to reader. Additionally, attackers can easily purchase readers and tags to perform malevolent operations. Therefore, we propose **A** **MU**tua**L** auth**E**ntication pro**T**ocol (*AMULET*) for readers and tags to prevent attackers from impersonating authorized entities.

Wagner et al. propose a PRF-based private authentication protocol in [66], which extends Weis's randomized hash lock protocol. Their authentication scheme comprises of a triple of probabilistic polynomial time algorithms (G, R, T) (for Generator, Reader, and Tag). Also, each tag possesses a unique secret s_i and identification ID_i, and the reader contains a database storing all pairs of (s_i, ID_i). In their protocol, each reader needs to authenticate every target, even if another reader previously validates the tag. This redundant authentication imposes unnecessary overhead on low computation power devices.

In our scheme, we assign each tag a unique secret s_i and identification ID_i and store all the tag pairs (s_i, ID_i) in a database. According to the protocol outlined in Figure 11, *AMULET* involves the following steps:

1. To begin the authentication process, the reader chooses a random number $R_1 \in \{0,1\}^n$, checks the current time T_1, and calculates $f(s_i \| R_1 \| T_1)$, where $\|$ indicates string concatenation. For a reader to authenticate a tag with ID_i, the reader then sends a *Hello* packet to the tag that includes R_1, T_1, and $f(s_i \| R_1 \| T_1)$.

2. When the tag receives a *Hello* packet, it chooses a random number $R_2 \in \{0,1\}^n$, checks the current time T_2, and calculates $\alpha = ID_i \oplus f(R_1 \| R_2 \| T_2)$. The tag sends a packet containing R_2, T_2, and α back to the reader and also saves a copy of R_2 and T_2. It is quite reasonable that tags have enough memory space to store these two parameters.

3. Upon receiving R_2, T_2, and α, the reader verifies that $\alpha = ID_i \oplus f(s_i \| R_1 \| R_2 \| T_2)$ and $T_2 > T_1$. It then checks for the current time T_3, computes the time difference $T = T_3 - T_2$, calculates $\beta = ID_i \oplus f(s_i \| R_1 \| R_2 \| T)$, and returns an *Ack* (acknowledgement) packet to the tag that includes T and β. In addition, the reader updates the original tag pair (s_i, ID_i) to (s_i, ID_i, R_2, T_2).

4. Finally, the tag validates the *Ack* packet by checking that $ID_i = \beta \oplus f(s_i \| R_1 \| R_2 \| T)$.

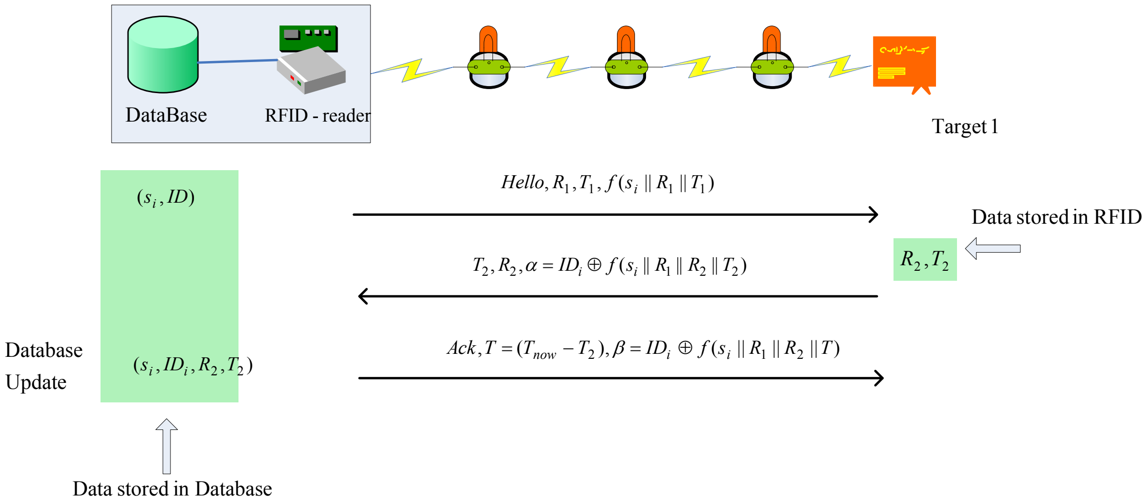

DataBase
RFID - reader
Target 1
(s_i, ID)
$Hello, R_1, T_1, f(s_i \| R_1 \| T_1)$
Data stored in RFID
$T_2, R_2, \alpha = ID_i \oplus f(s_i \| R_1 \| R_2 \| T_2)$
R_2, T_2
Database
Update
(s_i, ID_i, R_2, T_2)
$Ack, T = (T_{now} - T_2), \beta = ID_i \oplus f(s_i \| R_1 \| R_2 \| T)$
Data stored in Database

Figure 11 AMULET architecture.

AMULET can reduce the re-authentication cost when a reader wishes to send commands to an authenticated tag. The reader need not re-authenticate the tag because the database stores the tag's information (s_i, ID_i, R_2, T_2). As depicted in Figure 12, the tag can verify future commands by the following two steps:

1. If a new reader queries the database and obtains (s_i, ID_i, R_2, T_2) instead of (s_i, ID_i), then it recognizes that another reader already authenticated the tag with this ID_i. As a result, it chooses a random number $R_1' \in \{0,1\}^n$, checks for the current time T_3, computes the difference in time $T = T_3 - T_2$, and calculates $\beta' = ID_i \oplus f(s_i \| Cmd' \| R_1' \| R_2 \| T)$. The reader then sends its command Cmd', along with R_1', T, and β', to the tag.
2. Upon receipt of the Cmd' packet, the tag verifies that $T = T_3 - T_2$ and $ID_i = \beta' \oplus f(s_i \| Cmd' \| R_1' \| R_2 \| T)$ before executing Cmd'. Otherwise, the tag drops the command.

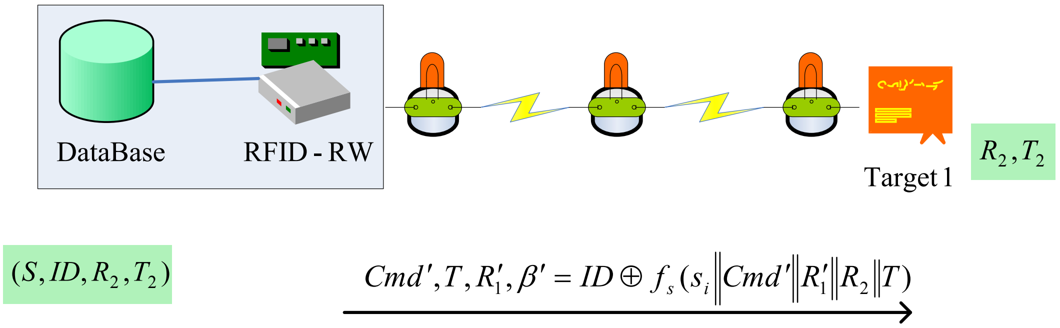

DataBase
RFID - RW
Target 1
R_2, T_2
(S, ID, R_2, T_2)
$Cmd', T, R_1', \beta' = ID \oplus f_s(s_i \| Cmd' \| R_1' \| R_2 \| T)$

Figure 12 Commands verification without re-authentication process.

As previously mentioned, it is harder to eavesdrop on the channel from tag to reader than from the reader to tag; accordingly, *AMULET* provides security against passive eavesdropping on the reader-to-tag link. A common attack to authentication protocols is man-in-the-middle attack, which *AMULET* naturally resists. Although an attacker can gather R_1 and T_1 from the reader and R_2, T_2, and $\alpha = ID_i \oplus f(s_i \| R_1 \| R_2 \| T_2)$ from the tag, it does not possess the secret key s_i, and thus cannot modify or inject its own α. Consequently, man-in-the-middle attacks will not succeed against our protocol, and we will formally prove this property in section 4. Furthermore, *AMULET* can defeat replay attacks when tags check that T has not expired and β or β' is valid for a first-time authentication or re-authentication procedure, respectively.

SSP

To preserve data privacy, simply encrypting data prevents attackers from discerning the contents. However, traditional cryptography is not feasible in tags and sensor nodes because of their limited computation capability. Moreover, it is difficult to search encrypted data. To solve this problem, we propose **a** **S**ecret **S**earch Pro**t**ocol (*SSP*), which maintains data in an encrypted form but allows authorized readers to perform searches without disclosing data during transmissions or queries.

SSP involves two phases: data encryption phase, and data search phase. The data encryption phase encrypts data and stores corresponding ciphertext to tags. The data search phase describes how to achieve private search on ciphertexts.

Data Encryption Phase

In SSP, tags store each characteristic of their associated target as an attribute of the target. We can formally describe a target as $B = (Attr1, Attr2, \cdots AttrN)$, where N is the number of attributes. For example, a tag attached to a book may store the book's ID, title, authors, check-in and check-out time, borrower's ID, etc. Personal attributes like borrower's ID must not be exposed to unauthorized readers or attackers. As shown in Figure 13, SSP involves the following steps:

1. For an attribute $AttrK$, the reader first generates $H^K(s_i, R_2)$ by iteratively hashing (s_i, R_2) K times, where K indicates the number of the sequential order of $AttrK$.
2. Next, the reader generates $H^K(EK_i)$ by iteratively hashing EK_i K times.
3. After calculating $f(s_i \| K \| R_2 \| H^K(s, R_2))$, the reader XOR it with K to form $\lambda = K \oplus f(s_i \| K \| R_2 \| H^K(s_i, R_2))$.
4. Finally, the reader computes $Attr'K = AttrK \oplus H^K(EK_i) \oplus \lambda$ and overwrites $AttrK$ with $Attr'K$.

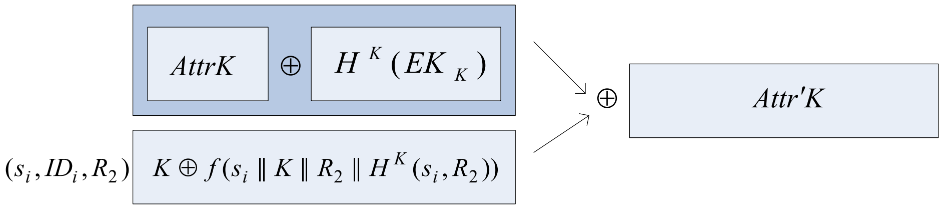


Figure 13 SSP operations for attribute K.

Once every attribute is overwritten, attackers will learn nothing from the encrypted data. Since K is different for all attributes, each attribute generates a different encrypted attribute value even if some attribute values happen to be the same. This will keep attributes relatively private. Figure 14 illustrates SSP's operations.

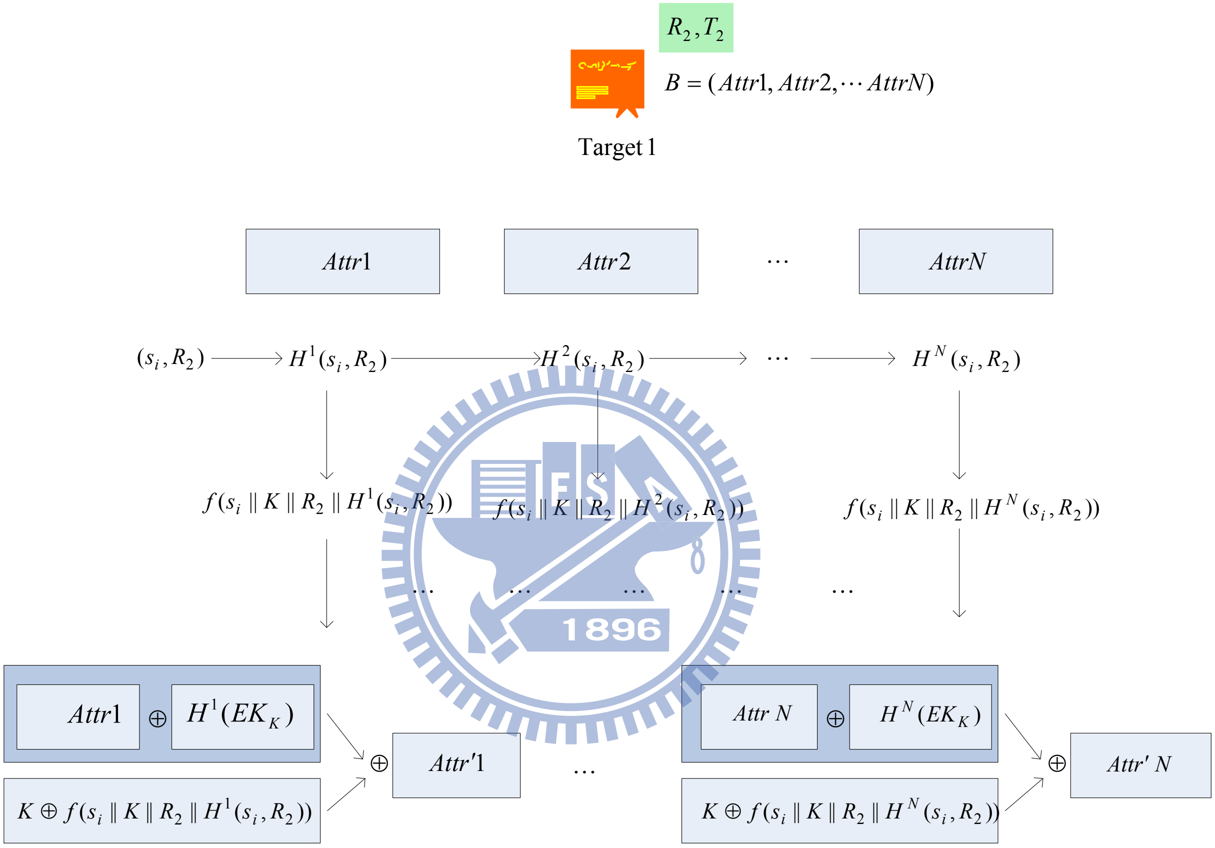

R_2, T_2
$B = (Attr1, Attr2, \cdots AttrN)$
Target 1
$Attr1$
$Attr2$
$\cdots$
$AttrN$
(s_i, R_2)
$H^1(s_i, R_2)$
$H^2(s_i, R_2)$
$H^N(s_i, R_2)$
$f(s_i \| K \| R_2 \| H^1(s_i, R_2))$
$f(s_i \| K \| R_2 \| H^2(s_i, R_2))$
$f(s_i \| K \| R_2 \| H^N(s_i, R_2))$
$Attr1 \oplus H^1(EK_K)$
$K \oplus f(s_i \| K \| R_2 \| H^1(s_i, R_2))$
$Attr'1$
$Attr\ N \oplus H^N(EK_K)$
$K \oplus f(s_i \| K \| R_2 \| H^N(s_i, R_2))$
$Attr'\ N$

Figure 14 SSP operations.

Authorized readers can inversely-transform $Attr'K$ back to $AttrK$ by computing $AttrK = Attr'K \oplus H^K(EK_i) \oplus K \oplus f(s_i \| K \| R_2 \| H^K(s_i, R_2))$. Because authorized readers can retrieve (s_i, R_2) from the database, they can easily calculate $AttrK$ without exposing sensitive and private data during wireless transmission.

A major contribution of SSP is that it ensures the privacy of the remaining attributes in the event that some attributes are compromised. Since $f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ varies by K, $Attr'(K+1) = Attr(K+1) \oplus H^{K+1}(EK_i) \oplus (K \oplus f(s_i \| K \| R \| H^{K+1}(s_i, R_2)))$ will remain secure even when $f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ is compromised.

Data Search Phase

To search for an attribute $AttrK$, the RFID-reader broadcasts an encrypted query $AttrK \oplus H^K(EK_i)$ to all sensor nodes. Next, each sensor node calculates $Attr'K$ by $Attr'K = AttrK \oplus H^K(EK_i) \oplus K \oplus f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ with its own s_i, R_2, and every value of K. The sensor node must calculate an $Attr'K$ for all $K's$ because it does not know the value of K. If any sensor node finds a match, it returns $Attr'K$ and K to the RFID-reader. Since data is encrypted, privacy is maintained during the transmission.

Security Analysis

In this section, we first demonstrate the security of *AMULET* under man-in-the-middle attacks. Second, we provide an analysis that discusses the resources required to break SSP.

Before we proceed to theoretical proof, we first describe the security requirements specifying the attacker’s abilities and when the later is considered successful. The abilities and disabilities of the attackers include:

- The attacker has an arbitrary polynomial-time computation power
- The attacker can eavesdrop to messages in the Air
- The attacker can modify encrypted messages

- The attacker can compromise tags
- The attacker cannot compromise readers
- The attacker cannot not know the shared secrets s_i and the encryption keys EK_i

An attacker is considered to be successful if the attacker can comprise the original messages or attributes, or forge an legal encrypted data. In our system, we consider only passive attacks where attackers can only listen to the messages transmitted in the Air or modify the messages. We do not intend to solve active attackers' problem as these kinds of attacks are not hard to be solved merely by any cryptographic algorithms.

Before we begin our proof, we give several definitions below.

Definition 1: (*Instance*) We can formally describe a target by its *ID* and attributes, where $B=(ID_B, Attr1, Attr2, \cdots AttrN)$. An *instance* X_B is defined as $X_B=(Attr1, Attr2, \cdots, AttrN)$, and a verification function V_f is defined as $V_f(X_B)=\sum_{i=1}^{n} Attri$. Each instance is a part of the distributed database, and the verification function is used to distinguish one instance from another.

Definition 2: (*Distinguishable*) Two instances of a target are distinguishable if any attribute has different values.

Defintion 3: (*R*-Breakable) Let an instance $X_B=(Attr1, Attr2, \cdots, AttrN)$. If X_B can be derived from *R* ($R \leq N$) attributes, then it is *R*-Breakable. Under the same condition, a system is *R*-Breakable if it needs *R* resources to break the system.

Security of AMULET

We classify man-in-the-middle attacks into three categories: type-1 attack modifies R_1 only, type-2 attack modifies R_2 only, and type-3 attack modifies R_1, R_2, and α. We will show that these three types of attacks fail against our authentication protocol.

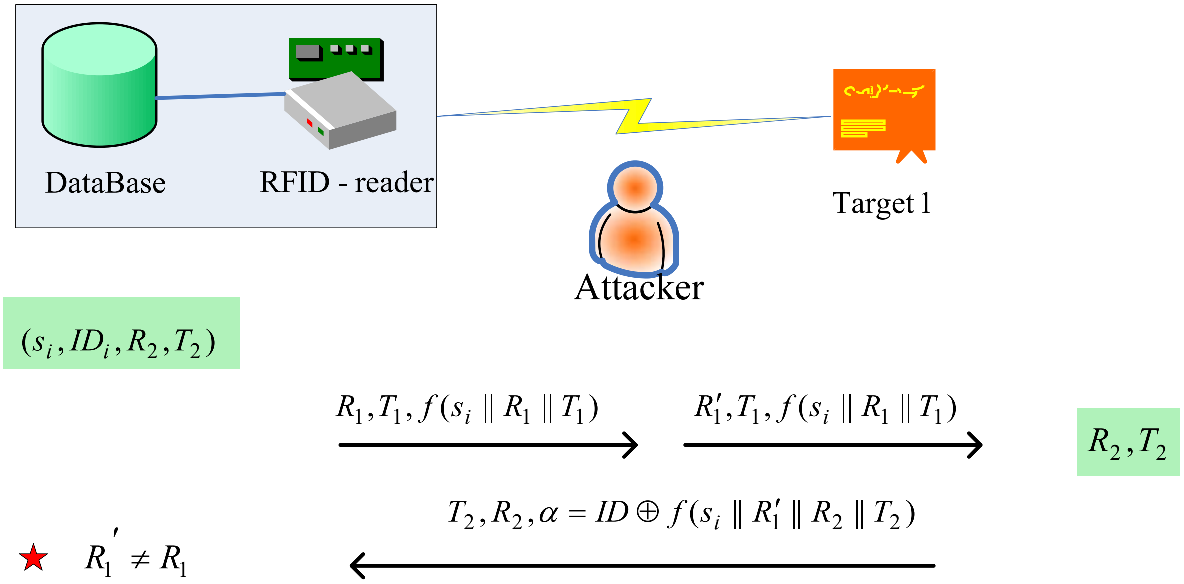

DataBase
RFID - reader
Target 1
Attacker
(s_i, ID_i, R_2, T_2)
$R_1, T_1, f(s_i \| R_1 \| T_1)$
$R_1', T_1, f(s_i \| R_1 \| T_1)$
R_2, T_2
$T_2, R_2, \alpha = ID \oplus f(s_i \| R_1' \| R_2 \| T_2)$
★ $R_1' \neq R_1$

Figure 15 Type-1 man-in-the-middle attack

Type-1 attacker, shown in Figure 15, eavesdrops on R_1, generates a false value R_1', and delivers it to the tag. The tag then uses R_2 to generate $\alpha = ID \oplus f(s_i \| R_1' \| R_2 \| T_2)$ and sends R_2, T_2, and α back to the reader. Since $R_1 \neq R_1'$, the reader will find that $f(s_i \| R_1 \| R_2 \| T_2) \neq f(s_i \| R_1' \| R_2 \| T_2)$. As a result, the readers can prevent type-1 man-in-the-middle attacks.

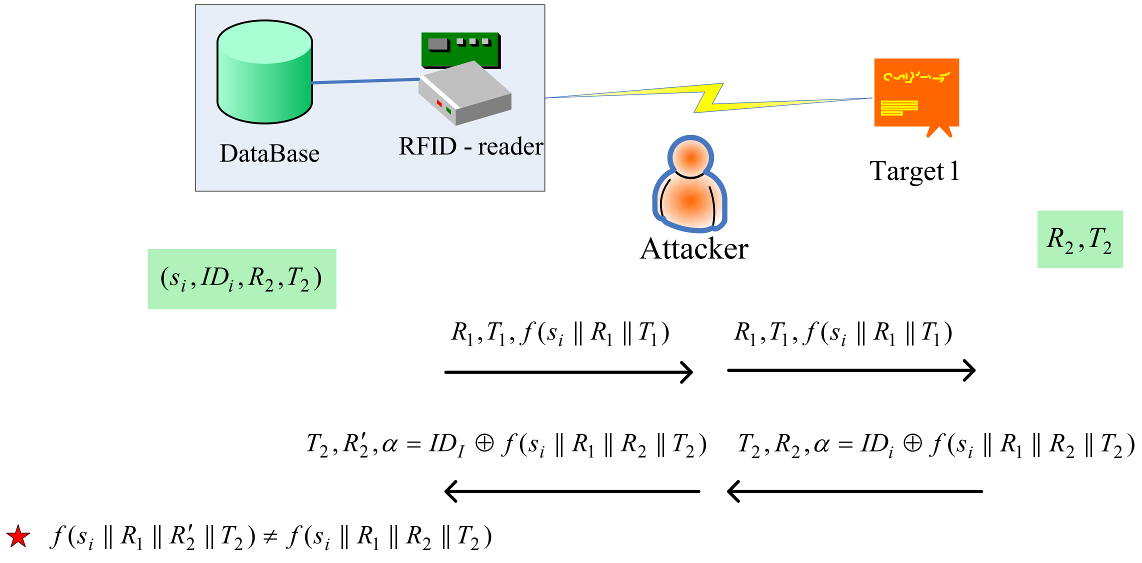

DataBase
RFID - reader
Target 1
Attacker
(s_i, ID_i, R_2, T_2)
R_2, T_2
$R_1, T_1, f(s_i \| R_1 \| T_1)$
$R_1, T_1, f(s_i \| R_1 \| T_1)$
$T_2, R_2', \alpha = ID_I \oplus f(s_i \| R_1 \| R_2 \| T_2)$
$T_2, R_2, \alpha = ID_i \oplus f(s_i \| R_1 \| R_2 \| T_2)$
★ $f(s_i \| R_1 \| R_2' \| T_2) \neq f(s_i \| R_1 \| R_2 \| T_2)$

Figure 16 Type-2 man-in-the-middle attack.

As depicted in Figure 16, a type-2 attacker eavesdrops on R_2, produces a false value R_2', and transmits R_2 and α back to the reader. Because $R_2 \neq R_2'$, the reader will find that $f(s_i \| R_1 \| R_2 \| T_2) \neq f(s_i \| R_1' \| R_2 \| T_2)$, thus thwarting type-2 man-in-the-middle attacks.

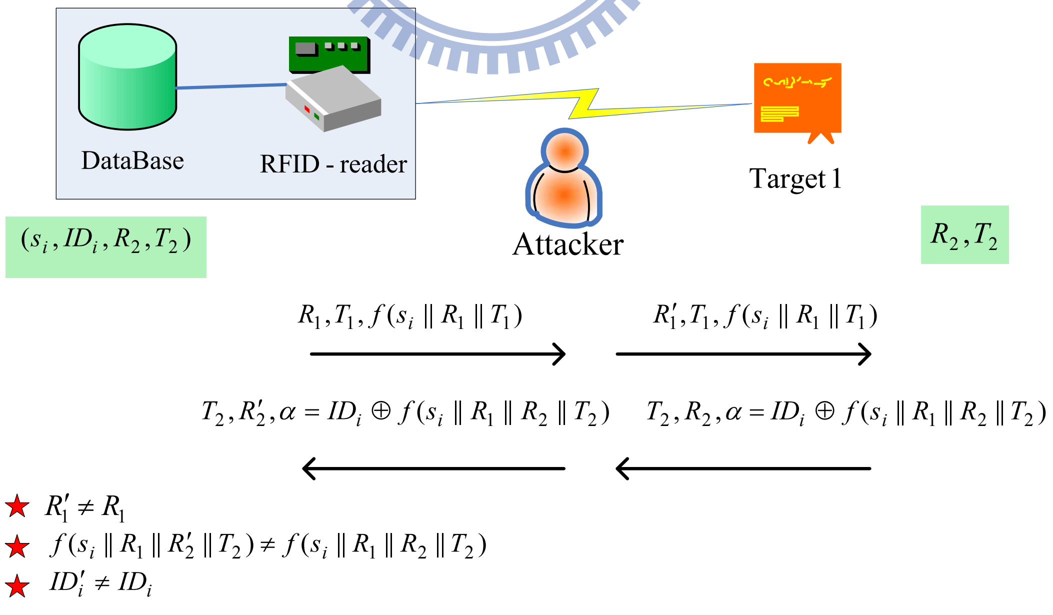

DataBase
RFID - reader
Target 1
Attacker
(s_i, ID_i, R_2, T_2)
R_2, T_2
$R_1, T_1, f(s_i \| R_1 \| T_1)$
$R_1', T_1, f(s_i \| R_1 \| T_1)$
$T_2, R_2', \alpha = ID_i \oplus f(s_i \| R_1 \| R_2 \| T_2)$
$T_2, R_2, \alpha = ID_i \oplus f(s_i \| R_1 \| R_2 \| T_2)$
★ $R_1' \neq R_1$
★ $f(s_i \| R_1 \| R_2' \| T_2) \neq f(s_i \| R_1 \| R_2 \| T_2)$
★ $ID_i' \neq ID_i$

Figure 17 Type-3 man-in-the-middle attack.

In Figure 17, a type-3 attacker generates false R_1', R_2', and α' back to the reader and the tag separately. Since s_i remains secret, the reader will observe that $f(s_i \| R_1 \| R_2 \| T_2) \neq f(s_i \| R_1 \| R_2' \| T_2)$ and $ID_i \neq ID'$, causing type-3 man-in-the-middle attacks to fail.

Security of SSP

We provide a proof of SSP's security strength in terms of the secrecy of its attributes. By establishing the number of resources required to compromise a system, we can evaluate its security strength. Theorem 1 states that an attacker must have knowledge of both s and R_2 to compromise $Attr'K$, where

$$Attr'K = AttrK \oplus H^K(EK_i) \oplus (K \oplus f(s_i \| K \| R_2 \| H^K(s_i, R_2))) \qquad \text{Eq(13)}$$

Theorem 1: $f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ is (s_i, R_2)-breakable.

Proof: Since attackers may extract the values of N and K, only s and R_2 must be kept secret. Attackers must know both s_i and R_2 to compromise $f(s_i \| K \| R_2 \| H^K(s_i, R_2))$. Thus, $f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ is (s_i, R_2)-breakable.

An instance is a collection of all attributes of a tag whose security strength is defined by the number of attributes needed to compromise the tag. Thus, as the number of distinguishable attributes increase, the instance will attain a higher security level.

Theorem 2: Given an instance of any two attributes *Attr'I*, *Attr'J*, where $I \neq J$, there does not exist a different instance *Attr''I*, *Attr''J*, such that the verification function evaluates to the same value $V_f(Attr'I, Attr'J) = V_f(Attr''I, Attr''J)$.

Proof: Let *AttrI*, *AttrJ* be two original attributes such that *I* > *J*, *Attr'I*, *Attr'J*, be their transformed attributes, and $V_f(Attr'I, Attr'J)$ be the verification of the transformed attributes. We will prove that an attacker cannot generate attributes *Attr''I*, *Attr''J* that satisfies $V_f(Attr'I, Attr'J) = V_f(Attr''I, Attr''J)$.

From equation 1, we know that

$$V_f(Attr'I + Attr'J) =$$
$$(AttrI \oplus H^I(EK_i) \oplus (K \oplus f(s_i \| K \| R_2 \| (H^I(s_i, R_2)))) + (AttrJ \oplus H^J(EK_i) \oplus (K \oplus f(s_i \| K \| R_2 \| (H^J(s_i, R_2))))$$

An important property of our protocol is that *AttrI* can be used to authenticate *AttrJ* by checking that

$$H^J(s_i, R_2) = H^{J-I}(H^I(s_i, R_2)) \qquad \text{Eq(14)}$$

If an attacker generates attributes *Attr''I*, *Attr''J*, $H^I(s_i, R_2)$ and $H^J(s_i, R_2)$ can be calculated by the following two equations.

$$Attr''I \oplus AttrI = I \oplus f(s_i \| I \| R_2 \| H^I(s_i, R_2)) \qquad \text{Eq(15)}$$

$$Attr''J \oplus AttrJ = J \oplus f(s_i \| J \| R_2 \| H^J(s_i, R_2)) \qquad \text{Eq(16)}$$

Because only authorized readers and tags know s_i and R_2, the attacker cannot falsify $H^I(s_i, R_2)$ and $H^J(s_i, R_2)$. This property is vital because if the attacker could successfully generate false attributes and the readers or tags cannot aware of the falsity, the attacker could therefore inject unnecessary data or operation to tags. This makes readers or tags unreliable.

The next theorem stipulates that an attacker must compromise all attributes of an instance to deceive readers. If only a portion of the attributes are compromised, the reader can still verify the instance. We will use induction to show that an instance of a target *B* is *N*-breakable and distinguishable, where *N* is the number of attributes of *B*.

Theorem 3: Let $V_f(B)=\sum_{i=0}^{n} Attri = Attr'1+Attr'2+\cdots+Attr'N$. B is N-breakable and distinguishable.

Proof: Let $B=(Attr1, Attr2,\cdots, AttrN)$ be the original attributes and $B'=(Attr'1, Attr'2,\cdots, Attr'N)$ be the attributes after transformation.

For $N=2$, B is 2-breakable by theorem 2.

Suppose when N=P, B is P-breakable. We want to prove B is P-breakable when N=P+1. Let $B_1=(Attr1, Attr2,\cdots, AttrN, AttrN+1)$. From theorem 2, we know that every pair of attributes is distinguishable. Therefore, $AttrN+1$ and $AttrM$ are distinguishable for $M=1,2,\cdots N$ by verifying $H^{N+1}(s_i,R_2)$ and $H^{1}(s\ ,R_2), H^{2}(s_i,R_2),\cdots, H^{N}(s_i,R_2)$ respectively. Since all $N+1$ attributes are distinguishable, we have shown that an instance of a target is N-breakable.

If the new attribute $AttrK$ is inserted between $Attr1$ and $AttrN$, $AttrK$ can be verified by both its predecessor attribute $Attr(K-1)$ and its successor attribute $AttrK+1$ through equations 5 and 6.

$$H(H^{K-1}(s_i,R_2))=H^{K}(s_i,R_2) \qquad \text{Eq(17)}$$

$$H(H^{K}(s_i,R_2))=H^{K+1}(s_i,R_2) \qquad \text{Eq(18)}$$

If both equations 5 and 6 are satisfied, the added attribute $AttrK$ is valid. Otherwise, $AttrK$ is invalid and should be discarded. Since an instance B is N-breakable, it needs to compromise entire N attributes to achieve falsity. Moreover, if only one attribute is compromised, the attacker cannot use this attribute to generate other false attributes.

Summary

In this section, we discuss some practical considerations for the proposed schemes and give comparisons with related work.

Practical Considerations

The proposed secret search protocol (SSP) is feasible for a network with sensor nodes and RFID tags as SSP uses low-computation operations, that is, a hash function and a random number generating function, to encrypt and search data. Unlike other schemes, SSP gets clients (tags) involved in the data encryption process. The reader uses the random number R_2 generated by each tag as a parameter in encryption process. It is clear that in the scheme, each tag generates different R_2, and this enhances security and privacy strength of the data encryption process. It can be verified by the readers if the data is being copied to another tag if the tag does not know R_2.

Supporting Fixed-Index Search Queries

A problem occurred in private search schemes [39] is that private search schemes are hard to provide fixed-index search among different clients. Most private search schemes uses different encryption keys for each client (databases) to provide better security strength when a client is compromised. However, using different encryption keys causes inconvenience in searching data. The private search scheme provides fixed-index search queries, however, it leads to a security and privacy problem, that is, all encrypted data are identical in clients.

In SSP, a reader can use the same encryption key EK for every tag, but still the encrypted data are different for every tag as long as these tags do not use the same R_2. In this case, the data encryption process can be reduced to:

$$ciphertext = ATTR\, K \oplus H(EK) \oplus K \oplus f(s_i \,\|\, K \,\|\, R_2 \,\|\, H^K(s_i, R_2)) \qquad \text{Eq(19)}$$

Since each tag owns different R_2, the ciphertext will be different according to different tags. To search, the reader uses $ATTR\,K \oplus H(EK)$ to search all tags (clients), and each tag will generate their own $K \oplus f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ and check if there's any attribute satisfying eq. 7. Our proposed encryption scheme simplifies search query by supporting fixed-index search but remains data secrecy and privacy for different clients (tags).

Supporting Ciphertext Update

It is obvious that if a reader is compromised, the compromised reader can retrieve all encrypted data in clients and recover all private information stored in clients. In AMULET, this can be solved as tags can support ciphertext update if a tag was notified that a reader was compromised. The tag then can choose a new random number R_2' and update the original ciphertext $ciphertext = ATTR\,K \oplus H(EK_i) \oplus K \oplus f(s_i \| K \| R_2 \| H^K(s_i, R_2))$ to

$$(updated)\, ciphertext = ATTR\,K \oplus H(EK_i) \oplus K \oplus f(s_i \| K \| R_2' \| H^K(s_i, R_2'))\ .$$

Then the tag just notify all but the compromised readers to update their (s_i, R_2) to (s_i, R_2') to finish the ciphertext update.

It is our advantage that even the random number R_2 is changed to R_2', all reader can still use $ATTR\,K \oplus H(EK_i)$ to query data. The query process remains the same. However, when a compromised reader receives $ATTR\,K \oplus H(EK_i) \oplus K \oplus f(s_i \| K \| R_2' \| H^K(s_i, R_2'))$, since the compromised reader still stored unmodified (s_i, R_2) in its database, the compromised reader cannot retrieve $ATTR\,K$. The data remains secure and private.

Chapter 5

Adaptive Random Key Distribution Schemes for Wireless Sensor Networks

Wireless Sensor Network (WSN) is a kind of network composed of nodes associated with sensors. Each node has the characteristics of small size, limited power, low computation and wireless access. The sensor node is responsible for collecting and delivering data over wireless network, and it is desirable to keep the delivered data confidential along the wireless transmission path from one node to another.

To ensure secure peer-to-peer wireless communication, the shared session key between any two nodes must be derived. Some protocols use a trusted third party to deliver keys to every node, while other protocols pre-distribute communication keys to all nodes. Since WSNs are self-organized and trusted third party may not be available, key pre-distribution protocols are often adopted in such networks. However, key pre-distribution protocols need to store session keys in every node. This may be difficult in a sensor network where thousands of nodes are deployed with limited storage space only enough to store a small number of session keys. It is desirable to design a new key pre-distribution protocol, which can reduce the storage space of session keys for a large WSN without degrading its security.

Eschenauer and Gligor [23] proposed a key management scheme based on Random Graph Theory. The Random Graph Theory is defined as follows. A random graph $G(n, p)$ is a graph with n nodes, and the probability that a link exists between any two nodes in the graph is p. When p is zero, the graph G has no edges, whereas when p is one, the graph G is fully connected. Erdős and Rényi [9] showed the monotone properties of a random graph $G(n, p)$ that there exists a threshold value of p, over which value the property exhibits a “phase transition”, i.e. the probability for G to have that property will transit from “likely false” to “likely true”. The threshold probability is defined by:

$$p = \frac{\ln(n) - \ln(-\ln(P_c))}{n} \qquad \text{Eq(20)}$$

where Pc stands for desired probability of the property.

Furthermore, the expected degree of a node can be calculated by:

$$d = p*(n-1) = \frac{(n-1)(\ln(n) - \ln(-\ln(P_c)))}{n} \qquad \text{Eq(21)}$$

Therefore, the scheme only needs to select d keys to keep a network connected under probability p. It can then significantly reduce the key space. These results are adopted herein as base assumptions. We will propose two key distribution schemes: Adaptive Random Pre-distributed scheme (ARP) and Uniquely Assigned One-way Hash Function scheme (UAO). Both schemes pre-distribute keys in each node before its deployment. According to random graph theory, a sensor network can be connected as long as enough keys are selected. Therefore, each node can communicate with each other without key exchange, which can save computational overhead for communications. More than that, both schemes minimize the storage requirement for key management. Though UAO scheme needs more storage space than ARP does, it provides mutual authentication.

The rest of this paper is organized as follows: The Adaptive Random Pre-distributed scheme and the Uniquely Assigned One-way Hash Function scheme are presented in Section II and III, respectively. Finally we give a conclusion in Section IV.

Adaptive Random Pre-distribution Scheme

ARP scheme is composed of two parts. One is the key pool, and the other is the key selection algorithm. The purpose of key pool is to store randomly generated keys, and the key selection algorithm is to select a set of keys from the key pool. Every node needs to select a set of keys from the key pool by using key selection algorithm before its deployment. These selected keys are saved in each node's storage space. Any two nodes shares a common key is able to securely communicate with each other by using this shared key. In ARP, the key pool is a two-dimensional key pool in which keys are generated in two phases, and are arranged in two-dimensional order matrix. The key is pre-generated as follows:

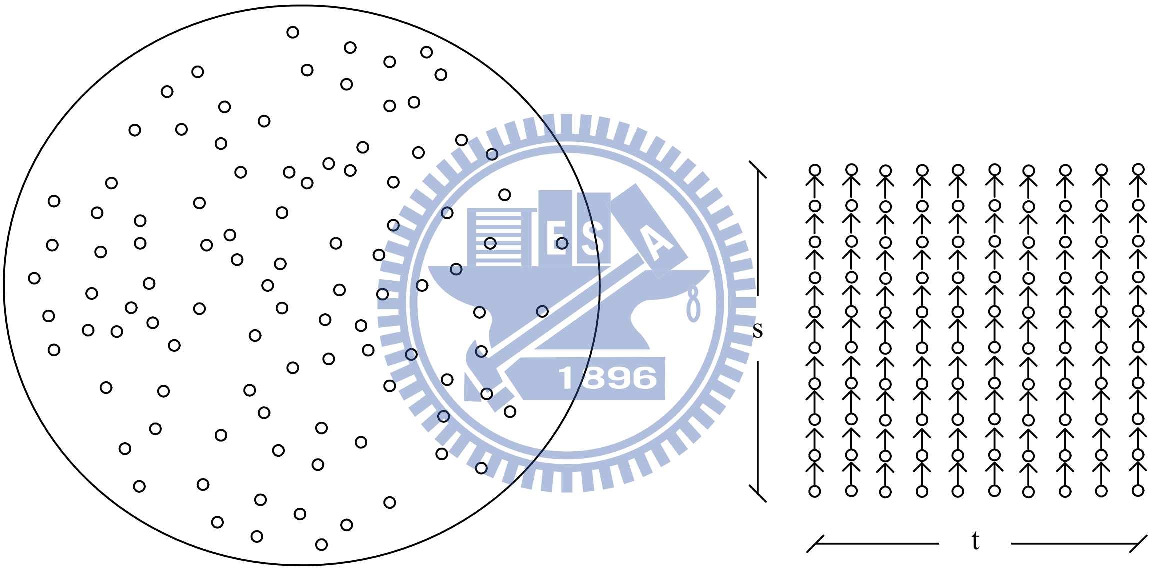
Key Pool Generation Algorithm

Step 1: Randomly generate t keys, called *seed keys*, and any t one-way hash functions.

Step 2: For every seed key $K_{i,0}$ and one-way hash function F_i, a one-way key chain is generated. It uses $K_{i,0}$ as initial input, and computes the generated key with a one-way hash function F_i. The generated key is fed back into F_i to generate a third key. The procedure $K_{i,j+1} = F(K_{i,j})$ is repeated until the entire key chain is generated.

Consequently, the key chain KC_0 of length *s*, is composed of a series of keys, $K_{i,0}, K_{i,1}$, …, $K_{i,s-1}$. With *t* seed keys and *t* one-way hash functions, *t* key chains generated, namely $KC_0, KC_1, \ldots, KC_{t-1}$.

Figure 18 demonstrates the difference between the conventional random key pool and the Two-Dimension Key Pool. As shown in Figure 18(a), the original random key pool can be regarded as a set of keys disorderly spread into a large pool. In Figure 18(b), keys of the Two-Dimension Key Pool are arranged in an *s* by *t* matrix.



(a) The unordered key pool (b) The Two-Dimension key pool

Figure 18 Unordered key pool and the Two-Dimension key pool with t = 10, s = 10.

Key Selection Algorithm

The key selection algorithm is used to select a set of communication keys by all nodes before its deployment. The detail of the key selection algorithm for ARP scheme is described as follows.

First, suppose we need *r* keys, each sensor node randomly pick up an one-way key chain $KC_i = (KC_{i,0}, KC_{i,1}, \cdots, KC_{i,t-1})$ from the Two-Dimension Key Pool. Second, each sensor node randomly picks up the rest $r' = r - t$ keys. Each key is selected from different key chains, except the key chain KC_i selected in step 1. Third, each sensor node will randomly pick up one key in the key chain selected in step 2. Fourth, each sensor node has chosen one key chain KC_i and *r'* single keys. For each sensor node, it will only need to memorize those *r'* keys and the one-way hash function F_i and seed key $KC_{i,0}$ of the key chain KC_i.

Figure 19 shows an example of key selection, where *t* = 10, *s* = 10, and *r'* = 5. The randomly selected one-way key chain is KC_3, and the rest *r'* randomly picked keys are $KC_{0,6}$, $KC_{5,8}$, $KC_{6,3}$, $KC_{8,7}$, and $KC_{9,4}$.

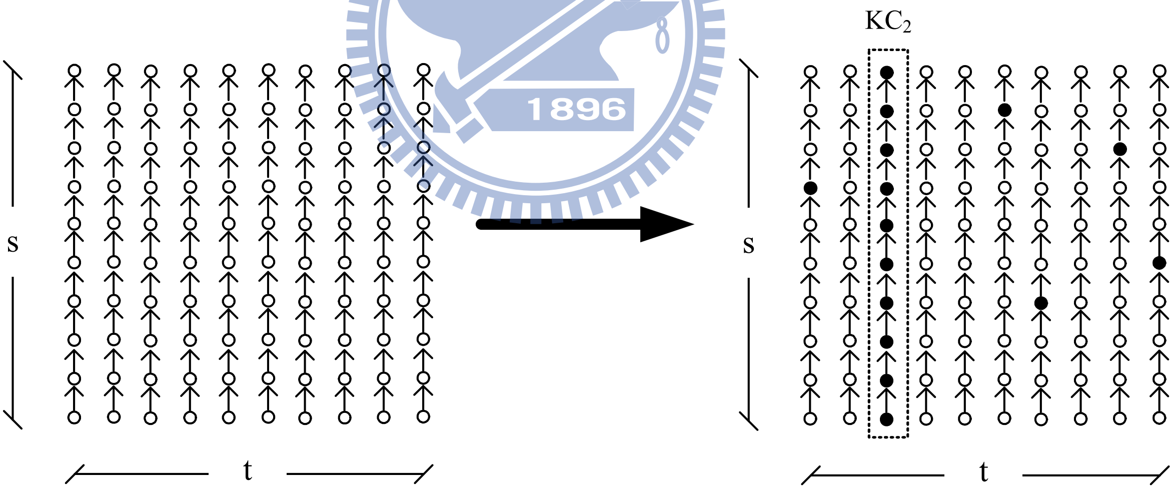


Figure 19 A key selection example

Uniquely Assigned One-Way Hash Function Scheme

In ARP, any two nodes shared a session key can directly communicate with each other in a secure way. However, a key in ARP may be shared by more than two nodes, and therefore a node may not be able to authenticate with the shared key the identity of an individual. To cope with the problem, UAO extends ARP to authenticate individual sensor node identities. The detail of UAO is describes as follows.

For each sensor node *SNi*, it was assigned a unique identity *IDi* and a uniquely assigned one-way hash function *Fi* before its deployment. In contrast to ARP key selection algorithm, UAO scheme does not select key. Instead, it uses *IDi* and *Fi* to decide a key, where *IDi* can be the node's MAC address or identifier; and *Fi* is a one-way hash functions. The UAO key decision algorithm is as follows:

Key Decision Algorithm:

Step1: Assume the required number of keys to achieve the Random Graph theory is *r*. For each sensor node *SNi* in the network, the first *SNi* will randomly select *r* of other sensor nodes in the network. We denote the sensor nodes selected by *SNi* as SN_{v1}, SN_{v2}, …, SN_{vr}.

Step 2: For each *SNvj* , where *j* ranges from 1 to *r*, it will use its unique one-way hash function *Fj* to generate a unique *K*j for *SNi*. The *Kj* is generated by the following equation:

$$K_j = F_j(ID_i)$$

SNi will memorize all pairs of *Kj* and *IDj* in its key ring.

Mutual Authentication

After applying key decision algorithm, every node is deployed in a WSN. For any communication between two nodes, *SNi* and *SNj*, *SNi* shares unique session key *Kj* with *SNj, and*

SNj shares unique session key *Ki* with *SNi*. We can achieve mutual authentication due to *SNi* is the only one node which owns the unique one-way hash function *Fi*. If *SNi* can correctly calculate *Kj* and decrypt the cipher, then *SNj* can authenticate the identity of *SNi*. Due to *Kj* is derived from *Fi* and *IDj*, if *SNj* really owns the key *Kj* then it will make the correct response. Therefore the *SNi* will be able to authenticate *SNj* with *IDj*.

Evaluation

To evaluate ARP scheme and UAO scheme, both schemes are analyzed in space storage and security strength separately.

Evaluation of ARP Scheme:

To evaluate ARP scheme, the connectivity probability is analyzed because it was observed in the preceding section that ARP is proposed based on Random Graph Theory. If the connectivity probability of different schemes is putting the same, the scheme which needs minimum keys will need less storage space than other schemes.

To evaluate the required probability of connectivity, the network size *n* and the expected probability *Pc* of forming a connected graph must be determined. By given *n* and *Pc*, we can calculate the threshold probability *p* and the expected degree *d* by Equation (20) and (21). Moreover, since a sensor node cannot communicate with all other nodes in the network, only a limited number of neighbor nodes *n'* can be contacted. Therefore, the probability of sharing a common key between any two nodes in a neighborhood is:

$$p' = \frac{d}{n'}$$

Also, the required key ring size *s* and the key pool size *K* to achieve the probability of neighborhood connectivity can be determined.

We denote the probability of any two nodes in the neighborhood sharing at least one common key in Two-Dimension Key Pool Selecting scheme as *p'*. It is proved that *p'* is related to the number of key chains *t*, key chain length *s*, and the number of selected keys *r'*. The *p'* can be calculated by one minus the probability that any two nodes in the neighborhood do not sharing any key. To calculate the probability that any two nodes A and B do not sharing any key, the calculation can be categorized into four parts:

(1) A's one-way key chain does not match with B's one-way key chain.

(2) A's one-way key chain does not match with any B's selected keys.

(3) A's selected keys do not match with B's one-way key chain.

(4) A's selected keys do not match with any B's selected keys.

Since B selects one hash function and *r'* selected keys in different key chains, A's one-way key chain must belong to the rest of the *h – (r'+1)* key chains. Therefore, the probability of matching the first and the second conditions both are $\frac{h-(r'+1)}{h}$.

In the third condition, it is taken into consider that we randomly choose *r'* key chains from the key pool. A's *r'* selected keys must not belong to A's key chain. As to match the third condition, it must not also belong to B's key chain. Thus the probability can be calculated as

$$\frac{\binom{h-2}{r'}}{\binom{h-1}{r'}} = \frac{h-r'-1}{h-1}.$$

For the fourth condition, it is supposed that A and B have exactly *i* selected keys belonging to the same *i* key chains and the probability that A and B have exactly *i* selected keys belonging to the same *i* key chains as p(i). There are $\binom{r'}{i}$ ways to pick *i* common key chains from B's selected key ring. There are only (*h – 2 – r'*) key chains for us to pick up the rest of A's (*r' – i*) selected keys, due to we have to eliminate A's and B's key chains and the other *r'* key chains that B's *r'* selected keys belonging to. Thus there are $\binom{h-2-r'}{r'-i}$ ways to pick up the rest (*r' – i*) key chains. The total number of ways for A to pick r' key chains is $\binom{h-2}{r'}$. Therefore we get the following equation:

$$p(i) = \frac{\binom{r'}{i} \cdot \binom{h-2-r'}{r'-i}}{\binom{h-2}{r'}}$$

Moreover, considering that A and B have exactly *i* selected keys belonging to the same key chains, the probability that A's selected keys do not match with any B's selected keys becomes:

$$p(i) \cdot \left(1 - \frac{1}{y}\right)^i$$

Hence, to calculate the probability of matching the fourth condition, we have to consider all possible value of *i*, where *i* = 0, 1, 2, …, *r'*. Thus the probability for the fourth condition is:

$$\sum_{i=0}^{r'} p(i)\cdot\left(1-\frac{1}{y}\right)^{i}$$

By Summarizing the above four conditions, we can calculate the probability *p'* by the following equation:

$$p'=1-\left(\frac{h-(r'+1)}{h}\right)\cdot\left(\frac{h-r'-1}{h-1}\right)\cdot\left(\sum_{i=0}^{r'} p(i)\cdot\left(1-\frac{1}{y}\right)^{i}\right)$$

Figure 20 shows the probability of connectivity with different configurations of number of key chains *t* and the key chain length *s*.

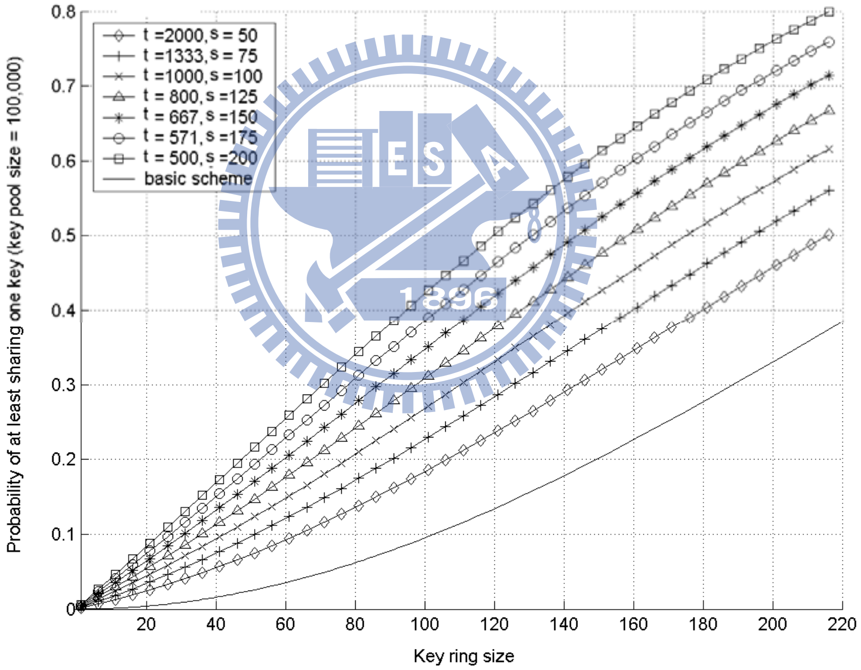


Figure 20 Comparison of different configured Two-Dimension Key Pool Selecting Schemes and basic scheme (key pool size is 100,000)

As Figure 20 shows, under the same connectivity probability, the ARP scheme requires fewer keys than basic method. In other words, the ARP scheme demands for less storage space than the basic scheme does. Moreover, with different h and y value, the ARP scheme needs different storage space. This can be left as an option for deployment consideration.

Evaluation of UAO:

In this section, evaluations of the probability of connectivity and the maximum supported network size are analyzed consequently. The maximum supported network size stands for maximum sensor node capacity that can achieve mutual authentication under the same memory storage space attached in every sensor node. In addition, we also make a comparison with the random-pairwise scheme in maximum supported network size and the probability of connectivity.

Probability of Connectivity:

In UAO scheme, the probability of any two nodes in the neighborhood sharing a common key can be evaluated by one minus the probability of that either nodes does not have any key derived from the other's unique one-way function. The probability for any node to get a key derived from a particular node's one-way function is $\frac{r}{n-1}$. Because each node gets r keys in the key ring, those keys are derived from r of $n - 1$ other nodes in the network. The probability of any two nodes in the neighborhood sharing a common key will be

$$p' = 1 - (1 - \frac{r}{n-1})^2 \qquad \text{Eq(22)}$$

Maximum Supported Network Size:

By combining Equation (22) and (23), the following the equation can be derived.

$$\frac{d}{n'} = 1-(1-\frac{r}{n-1})^2$$

Furthermore, by using Equation (21), the above equation can turn to be:

$$\frac{(n-1)(\ln(n)-\ln(-\ln(P_c)))}{n \cdot n'} = 1-(1-\frac{r}{n-1})^2$$

The above equation can be simplified to:

$$r^2 - 2(n-1)r + (n-1)^2 \cdot \frac{(n-1)(\ln(n)-\ln(-\ln(P_c)))}{n \cdot n'} = 0$$

By calculating the root of the above quadratic equation, we can get:

$$r = (n-1)(1-\sqrt{1-\frac{(n-1)(\ln(n)-\ln(-\ln(P_c)))}{n \cdot n'}}) \qquad \text{Eq(23)}$$

It can be more simplified as:

$$r = (n-1)(1-\sqrt{1-\frac{d}{n'}})$$

In comparison with Random-pairwise scheme, we assume the network size is *n*, expected degree of graph connectivity is *d*, the number of neighbor nodes is *n'*, and the key ring size is *r*. According to the definition of pairwise scheme, there are only *r* nodes having common shared keys with each sensor node and it still has to achieve the expected degree in the neighborhood. Then we can find the following equation:

$$d = \frac{r \times n'}{n} \Rightarrow r = \frac{d \times n}{n'} \qquad \text{Eq(24)}$$

To analysis the relationship between memory space and network size, first we combine Equation (21) and Equation (25) to obtain the following equation:

$$r = \frac{(n-1)}{n'} \cdot (\ln(n) - \ln(-\ln(P_c))) \qquad \text{Eq(25)}$$

According to the Equation (25), we can evaluate that the complexity of memory space requirement for the Random-pairwise scheme is $O(n \cdot \ln(n))$. In addition, according to the Equation (24), it is found that the complexity of memory space requirement for the UAO scheme is $O(n \cdot \sqrt{\ln(n)})$.

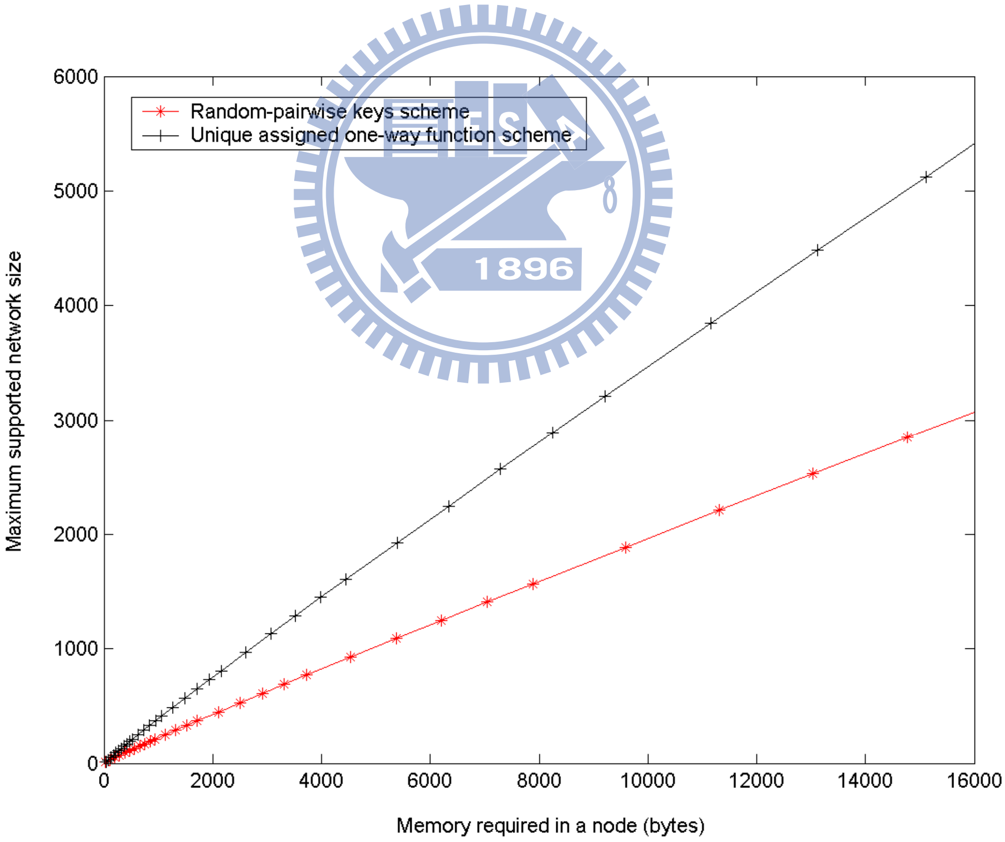


Figure 21 Comparison of Random-pairwise keys scheme and UAO scheme in memory requirement and maximum supported network size.

Figure 20 shows the comparisons of UAO scheme and random-pairwise keys distribution scheme in memory space requirement and the maximum supported network size. As Figure 21 shows, UAO scheme achieves better performance in maximizing network size under the same memory requirement. Therefore, with the same sensor node hardware equipment, UAO can adapt more sensor nodes in a network while remaining more security strength than random-pairwise key distribution scheme.

Summary

Key distribution is a critical and fundamental issue for the security service in wireless sensor networks. The pre-distributed and symmetric cryptography based key management system will be well suitable for the resource limited sensor network. Two efficient schemes are proposed which are based on the Random Graph Theory to provide key distribution for the secure sensor network services.

Adaptive Random Pre-distributed scheme needs less memory space at a given level of security strength. Uniquely Assigned One-Way Hash Function scheme possesses the characteristics of mutual authentication and resistance against node compromise. For a node with limited storage space, ARP can be used in the WSN with a large number of nodes.

The choice of schemes depends on the network size and available memory space. If there is enough memory space and authentication of individuals is desirable, Uniquely Assigned One-Way Hash Function scheme will be the better choice. This is mainly due to the fact that if a node is compromised, only one link is broken. Otherwise, the Adaptive Random Pre-distributed scheme serves as an alternative for the trade-off between memory space and security strength.

Chapter 6

Conclusion and Future Work

A wireless sensor network is a special network which has many constraints compared to a traditional computer network. These constrains include:

- Limited Memory and Storage Space

A sensor is a tiny device with only a small amount of memory and storage space. In order to build an effective security mechanism, it is necessary to limit the code size of the security algorithm and to limit the computation which can be affordable in sensor node.

- Power Limitation

Energy is the biggest constraint to wireless sensor capabilities. When implementing a cryptographic function or protocol within a sensor node, the energy impact of the added security code must be considered.

- Exposure to Physical Attacks

The sensor may be deployed in an environment open to adversaries. The likelihood that a sensor suffers a physical attack in such an environment is therefore much higher than the typical PCs, which is located in a secure place and mainly faces attacks from a network.

Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks. Therefore, to develop useful security mechanisms is very important.

In this dissertation, we proposed a security infrastructure for wireless sensor network. The proposed security infrastructure includes: a key distribution protocol (in Chapter 4), a mutual authentication and encrypted-data searching protocol (in Chapter 3), and a secure data aggregation protocol (in Chapter 2). When designing these protocols, we take consider into these constrains and make our protocol to be affordable in wireless sensor network.

Except design criteria listed above, we also need to consider the security requirement which a security protocol should fulfill in wireless sensor network.

- Data Confidentiality

A sensor network should not leak sensor readings to its neighbors and it is extremely important to build a secure channel in a wireless sensor network to keep data confidentiality. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, thus achieving confidentiality.

- Data Integrity

With the implementation of confidentiality, an adversary may be unable to steal information. However, this doesn't mean the data is safe. The adversary can change the data, so as to send the sensor network into disarray. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Thus, data integrity ensures that any received data has not been altered in transit

- Authentication

An adversary can not just limited to modifying the data packet but also can change the whole packet stream by injecting additional packets. So the receiver needs to ensure that the data are originated from the correct source. On the other hand, when constructing the sensor network, authentication is necessary for many administrative tasks. From the above, we can see that message authentication is important for many applications in sensor networks.

In the future, more investigations can be extended to our current algorithms.

- Mathematical Operation

Currently our proposed data aggregation algorithm can only eliminate duplicate data. In the future, more mathematical operations, such as add, sub, mean, etc., should be added. However, it is critical to balance both the security and computation consumption.

- Mobility Architecture

  Mobility is getting more and more important in WSN architecture. The sensor nodes could probe larger area with mobility. In mobility architecture, the case is totally different. First, nodes might be randomly deployed in one place but activate in another place. Therefore, algorithms required global knowledge is not feasible. Second, during data transmission, sensor nodes might be moving and secure data transmission is one new issue we're facing. Third, routing routines might be frequently changed. How to quickly update routing information is another issue.

For our proposed protocols, none of them can fulfill all the requirements mentioned above. However, mixing our proposed protocols can build a very good defense wall against attacks.

Aggregation is very useful technique deployed in wireless sensor network for saving more power and security should be serious considered when transmitting or aggregating data. We proposed a lightweight aggregation protocol feasible for off-the-shelf computation limited sensor node. Except aggregation, our proposed key distribution protocol, mutual authentication protocol, and secret searching protocol can adopted together to build a more secure wireless sensor network.

Reference

[1] M. Acharya and J. Girao, "Secure comparison of encrypted data in wireless sensor networks", In 3rd Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, p.47-53, 2005.

[2] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, pages 102-114, August, 2002.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", IEEE Communication Magazine, p.102-114, 2002.

[4] J. Al-Karaki, R. Ul-Mustafa, and A. Kamal, "Data aggregation in wireless sensor networks – exact and approximate algorithms", In Proceedings of the Workshop on High Performance Switching and Routing, p.241-245, 2004.

[5] N. Alon, Z. Galil and M. Yung, "Efficient dynamic-resharing verifiable secret sharing against mobile adversary", In *Proceedings of European Symposium on Algorithms*, pp.523-537, 1995.

[6] N. Asokan and P. Ginzborg, "Key agreement in ad hoc networks," Computer Communications, vol. 23, pp. 1627–1637, 2000.

[7] T. Atonishi and T. Matsuda, "Impact of aggregation efficiency on GIT routing for wireless sensor networks," In Proceedings of IEEE International Conference oon Parallel Processing Workshops, 2006.

[8] L. Ballard, M. Green, B. Medeiros and F. Monrose, "Correlation-Resistant Storage via Keyword-Searchable Encryption", In *Proceedings of EUROCRYPT*, 2005.

[9] F. Bao, "Cryptoanalysis of a provable secure additive and Multiplicative Privacy Homomorphism", In Proceedings of the International Workshop on Coding and Cryptography, p.43-50, 2003.

[10] S. Basagni, K. Herrin, E. Rosti, D. Bruschi, and E. Rosti, "Secure Pebblenets," In Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 156 – 163, 2001.

[11] M. Bellare and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” In Proceedings of 1st Conferendce on Computer and Communications Security, p. 62-73, 1993

[12] J. Benaloh, “Secret Sharing Homomorphisms: Keeping shares of a Secret Sharing”, In Advances in Cryptology - CRYPTO, p.251-260, 1986.

[13] K. Bennett, C. Grothoff, T. Horozov, and I. Patrascu, “Efficient Sharing of Encrypted Data”, In *Proceedings of the 7th Australian Conference on Information Security and Privacy*, pp.107-120, 2002.

[14] L. Buttyan, P. Shaffer and I.n Vajda, “Resilient aggregation with attack detection in sensor networks”, In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, p.332, 2006.

[15] H. Cam, S. Ozdemir, P. Nair, D. Muthuavinashinappan and H. Ozgur Sanli, “ESPDA: Energy-efficient secure pattern based data aggregation for wireless sensor networks”, Computer Communication, Vol.29, 2006.

[16] R. Canetti, O. Goldreich, and S. Halevi, “The random oracle methodology, revisited”, In Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, p.209-218, 1998.

[17] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” IEEE Symposium on Security and Privacy, May 2003.

[18] R. Chandramouli, S. Bapatla, and K. P. Subbalakshmi, “Battery power-aware encryption,” In Proceedings of ACM Transactions on Information and System Security, p.162-180, 2006.

[19] Y. Chen, A. Liestman, and J. Liu, “A hierachical energy-efficient framework for data aggregation in wireless sensor networks,“ IEEE Transactions on Vehicular Technology, p.789-796, 2006

[20] J. Choi, J. Lee, K. Lee, S. Choi, W. Kwon and H. Park, “Aggregation time control algorithm for time constrained data delivery in wireless sensor networks”, In Proceedings of Vehiculare Technology, p.563-567, 2006.

[21] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private Information Retrieval”, In *Proceedings Journal of the ACM*, pp.965-981, 1998.

[22] S. Chow, “Exclusion-Intersection Encryption and Its Application to Searchable Encryption”, In *Proceedings of EUROCRYPT*, 2005.

[23] I. Clarke, O. Sandberg, B. Wiley, S. Oskar, O. Wiley, and T. Hong, “Freenet: A Distributed Anonymous Information Storage and Retrieval System”, In *Proceeding of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, pp.311-320, 2000.

[24] J. Considine, F. Li, G. Kollios, and J. Byers, “Approximate aggregation techniques for sensor databases”, In Proceedings of IEEE Conference on Data Engineering, p.449, 2004.

[25] R. Cramer, I. Damgard, and J. B. Nielsen, “Multiparty Computation from Threshold Homomorphic Encryption,” In Advances in Cryptology – EUROCRYPT, p. 280-299, 2001.

[26] F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger, “Building Peer-to-Peer Systems With Chord, A Distributed Lookup Service”, *In Proceedings of the. 8th Workshop on Hot Topics in Operating System,* pp.81, May 2001.

[27] P. Devanbu and S. Stubblebine, “Stack and Queue Integrity on Hostile Platforms”, In *Proceedings of IEEE Transactions on Software Engineering*, pp.100-108, 2002.

[28] J. Domingo-Ferrer, “A provably secure additive and multiplicative privacy homomorphism”, In Proceedings of Information Security Conference, p.471-483, 2002.

[29] P. Erd˝os and A. R´enyi, “On the evolution of random graphs,” Publ. Math. Inst. Hungat. Acad. Sci., vol. 5, pp. 17–61, 1960.

[30] L. Eschenauer, V. Gligor, “A key-management scheme for distributed sensor networks”, In *Proceedings of the 9th ACM Conference on Computer and Communication Security*, pp.41-47, 2002.

[31] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” In Proceedings of the 9th ACM Conference on Computer and Communication Security, pp. 41–47, November 2002.

[32] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, “Next Century Challenges: Scalable Coordination in Sensor Networks,” In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, August 1999.

[33] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, “Next Century Challenges: Scalable Coordination in Sensor Networks”, In *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp.263-270, 1999.

[34] P. Feldman, “A practical scheme for non-interactive verifiable secret sharing”. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pp.427-438, 1987.

[35] J. Ferrer, “Privacy homomorphisms for statistical confidentiality”, *Qüestiió*, p.505-521, 1996.

[36] J. Frank, P. Cheeseman, and J. Stutz, “On The Complexity of Blocks-World Planning”, In *Proceedings of Artificial Intelligence*, pp.139—403, 1992.

[37] L. Gatani, G. Lo Re and M. Ortolani, “Robust and efficient data gathering for wireless sensor networks”, In Proceedings of the 39th Hawaii International Conference on System Sciences, p.235, 2006.

[38] S. Generiwal, R. Kumar and M. Srivastava, “Time-sync Protocol for Sensor Networks”, In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp.138-149, 2003.

[39] Y. Gertner, Y. Ishai, and E. Kushilevitz, “Protecting Data Privacy in Private Information Retrieval Schemes”, In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp.151-160, 1998.

[40] J. Girao, D. Westhoff, and M. Schneider, “CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks“, In Proceedings of 40th International Conf. on Communications, p.3044-3049, 2005.

[41] J. Girao, D. Westhoff, and M. Scheneider, “Concealed data aggregation in wireless sensor networks”, In Proceedings of ACM WiSe Conference, 2004.

[42] J. Greunen and J. Rabaey, “Lightweight Time Synchronization for Sensor Networks”, In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp.11-19, 2003.

[43] C. Gu, Y. Zhu and Y. Zhang, “Efficient Public Key Encryption with Keyword Search Schemes from pairings”, In *Proceedings of EUROCRYPT*, 2006.

[44] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, “Speed: A stateless protocol for real-time communication in sensor networks,” In International Conference on Distributed Computing Systems (ICDCS 2003), May 2003.

[45] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energyefficient communication protocols for wireless microsensor networks,” Proc. Hawaaian Int’l Conf. on Systems

Science, January 2000.

[46] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, August 1999.

[47] A. Hodjat and I. Verbauwhede, “The energy cost of secrets in adhoc networks,” IEEE CAS Workshop on Wireless Communications and Networking, September 2002.

[48] L. Hu and D. Evans, “Secure aggregation for wireless networks”, In Proceedings of Applications and Internet Workshops, p.27-31, 2003.

[49] J. Hubaux, L. Buttyan, and S. Capkun, “The quest for security in mobile ad hoc networks,” In Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing, October 2001.

[50] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM ’00), August 2000.

[51] C. Intanagonwiwat, R. Govindan, D. Estrin, and J. Heidemann, “Directed diffusion for wireless sensor networking”, In IEEE/ACM Transactions on Networking, p.2-16, 2003.

[52] H. Jiang and S. Jin, “Scalable and robust aggregation techniques for extracting statistical information in sensor networks”, In Proceedings of the 26th IEEE International Conference on Distributed Computing systems, p.69, 2006.

[53] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing robust and ubiquitous security support for mobil ad-hoc network,” Network Protocols Ninth International Conference on ICNP 2001, 2001.

[54] L. Krishnamachari, D. Estrin and S. Wicker, “The impact of data aggregation in wireless sensor networks”, In Proceedings of Distributed Computing Systems Workshops, 2002.

[55] E. Kusilevitz, and R. Ostrovsky, “Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval”, In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pp.364-373, 1997.

[56] J. Ledlie, J. Taylor, L. Serban, and M. Seltzer, “Self-Organization in Peer-to-Peer Systems”, In *Proceedings of 10th SIGOPS European Workshop*, 2002

[57] H. Li, H. Yu and A. Liu, "A tree based data collection scheme for wireless sensor network", In Proceedings of the IEEE International Conference of Networking, p.119, 2006.

[58] Z. Li, K. Li, C.n Wen, and Y. Soh, "A new chaotic secure communication system," In Proceedings of IEEE Transactions on Communications", p.1306-1312, 2003.

[59] T. Li, Y. Wu, and H. Zhu, "An Efficient Scheme for Encrypted Data Aggregation on Sensor Networks," In Proceedings of Vehicular Technology Conference, p. 831-835, 2006.

[60] C. Lin, W. Peng, and Y. Tsen, "Efficient in-network moving object tracking in wireless sensor networks", In Proceedings of IEEE Transactions on Mobile Computing, p.1044-1056,2006.

[61] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," the 10th Annual Network and Distributed System Security Symposium, February 2003.

[62] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," In Proceedings of Seventh International Symposium on Computers and Communications (ISCC 2002), pp. 567–574, 2002.

[63] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks", In Proceedings of 5th Symposium on Operating Systems Design and Implementation, 2002.

[64] A. Mahimkar and T. Rappaport, "SecureDAV: a secure data aggregation and verfication protocol for sensor networks", In Proceedings of Global Communication, 2004.

[65] R. Misra and C. Mandal, "Ant-aggregation: Ant Colony Algorithm for Optimal Data Aggregation in Wireless Sensor Networks", In Proceedings of International Conference on Wireless and Optical Communication Networks, p.5, 2006.

[66] D. Molnar and D. Wagner, "Privacy and Security in Library RFID Issues, Practices, and Architectures", In *Proceedings of ACM Conference on Computer and Communication Security*, pp.210-219, 2004.

[67] O. Moussaoui, A. Ksentini, M. Naimi, and M. Gueroui, "Efficient energy saving in wireless sensor networks through hierarchical-based clustering," In Proceedings of the Seventh IEEE International Symposium on Computer Networks, 2006.

[68] R. Ostrovsky and W. Skeith III, “Private Searching on Streaming Data”, In *Proceedings of Advances in Cryptology*, 2005.

[69] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, “Spins: Security protocols for sensor networks,” In Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking, July 2001.

[70] A. Perrig, and J. D. Tygar, “Secure boradcast communication in wired and wireless networks”, Kluwer Academic Publisher, 2002.

[71] G. Pottie, “Wireless Sensor Networks”, In *Proceedings of Information Theory Workshop*, pp.139-140, 1998.

[72] B. Przydatek, D. Song and A. Perrig, “SIA: secure information aggregation in sensor networks”, In Proceedings of ACM SenSys Conference, p.255-265, 2003

[73] C. Rackoff and D. Simon, “Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack,” In Proceedings of Advances in Cryptology, 1991

[74] M. Raina, S. Ghosh, R. Patro, G. Viswanath and Chadrashekhar T, “Secure data aggregation using commitment schemes and quasi commutative functions”, In Proceedings of 1st International Symposium on Wireless Pervasive Computing, p.16-18, 2006.

[75] M. Schneider, and E. Felten, “Efficient commerce protocols based on one-time pads”, In Proceedings of 16th Annual Computer Security Applications Conference , p.317, 2000.

[76] S. Shin, J. Lee, J. Baek, and D. Seo, “Reliable data aggregation protocol for ad-hoc sensor network environments,” In Proceedings of the 8th International Conference on Advanced Technology, 2006.

[77] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, “Medians and beyond: new aggregation techniques for sensor networks,” In Proceedings of the 2nd International Conferece on Embedded Networked Sensor Systems, p.239-249, 2004.

[78] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B. Srivastava, “On communication security in wireless ad-hoc sensor network,” Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’02), June 2002.

[79] D. Song, D. Wagner, and A. Perrig, “Practical Techniques for Searches on Encrypted

Data", In *Proceedings of IEEE Symposium on Security and Privacy*, pp.44-55, 2000.

[80] J. Spencer, The Strange Logic of Random Graphs. Springer-Verlag, 2000.

[81] A. Stephen, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", In *Proceedings of Pervasive Computing,* pp.201-212, 2004.

[82] H. Sun and S. Shieh, "An EfficientCconstruction of Perfect Secret Sharing Schemes for Graph-based Access Structures", In *Proceedings of Computers and Mathematics with Applications*, pp.129-135, 1996

[83] H. Sun and S. Shieh, "On Dynamic Threshold Schemes", *In Proceedings of Information Processing Letters*, pp.201-206, 1994.

[84] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless microsensor network models," ACM Mobile Computing and Communications Review (MC2R 2002), 2002.

[85] J. Undercoffer, S. Avancha, A. Joshi, and J. Pinkston, "Security for sensor networks," 2002 CADIP Research Symposium, 2002.

[86] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and approaches for distributed sensor network security," NAI Labs Technical Report #00- 010, September 2000.

[87] D. W. Carman, B. J. Matt, and G. H. Cirincione, "Energy-efficient and low-latency key management for sensor networks," In Proceedings of 23rd Army Science Conference, December 2002.

[88] D. Wagner, "Resilient aggregation in sensor networks", In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, p.78-87, 2004.

[89] D. Waters, D. Balfanz, G. Durfee and D. Smetters, "Building An Encrypted and Searchable Audit Log", In Proceedings of 11th Annual Network and Distributed Security Symposium (NDSS), 2004.

[90] D. Westhoff, J. Girao, and M. Acharya, "Concealed data aggregation for reverse multicast traffic in sensor networks: encryption, key distribution and routing adaptation", In Proceedings of IEEE Transactions on Mobile Computing, p.1417-1431, 2006.

[91] K. Wu, D. Dreef, B. Sun and Y. Xiao, “Secure data aggregation without persistent cryptographic operations in wireless sensor networks”, In Proceedings of Performance, Computing, and Communications Conference, p.6, 2006.

[92] S. Yi and R. Kravets, “Moca: Mobile certificate authority for wireless ad hoc networks,” 2nd Annual PKI Research Workshop Program (PKI03), April 2003.

[93] S. Yi and R. Kravets, “Key management for heterogeneous ad hoc wireless networks,” the 10th IEEE International Conference on Network Protocols (ICNP2002), 2002.

[94] Y. Yu, J. Leiwo, and B. Premkumar, “A study on the security of privacy pomomorphism,” In Proceedings of The 3rd Conference on Information Techonology, p.470-475, 2006.

[95] Y. Zheng, T. Hardjono and J. Seberry, “How to recycle shares in secret sharing schemes”, In *Proceedings of Austral Computer Science Communications*, pp.1053-1064, 1992.

[96] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” IEEE Networks Magazine, vol. 13, no. 6, pp. 24–30, November/December 1999.

Curriculum Viate

Shih-I Huang received B.S. and M.S. degree in Applied Mathematics from Nation Sun-Yat Sen University He ever worked with teams from U.C. Berkley and Carnegie Mellon University in an International Collaboration Project where he played an important role in one of the sub-project for two years. He’s currently a Ph.D. candidate in CSIE, NCTU, and a PMP (Project Management Professional). His research interests include network security, information security, wireless sensor network, data protection, data privacy, system integration, and project management.

Publication List

● Journal Papers

[1] **Shih-I Huang**, Shiuhpyng Shieh, J. D. Tygar, Secure Encrypted-Data Aggregation for Wireless Sensor Networks, ACM/Springer Wireless Networks, 2009.

[2] ***Shih-I Huang***, Shiuhpyng Shieh, Authentication and Secret Search Mechanisms for RFID-Aware Wireless Sensor Networks, International Journal of Security and Networks, 2008.

[3] ***Shih-I Huang***, Introduction to Mathematical Approaches for Data Aggregation in Wireless Sensor Networks, ICL Technical Journal, 2007.

● Conference Papers

[4] Phoebus Chen, Parvez Ahammad, Coly Boyer, ***Shih-I Huang***, Leon Lin, Edgar Lobation, Marci Meingast, Songhwai Oh, Simon Wang, Posu Yan, Allen Yang, Chuohao Yeo, Lung-Chung Chang, Doug Tygar, and Shankar Sastry, Low-Bandwidth Wireless Camera Networks: Architecture and Application, ACM/IEEE Int'l Conf. on Distributed Smart Cameras, U.S., 2008

[5] Po-Yuan Teng, ***Shih-I Huang***, and Adrian Perrig, Multi-Layer Encryption for Multi-Level Access Control in Wireless Sensor Networks, 23rd Int'l Information Security Conference, Milan, Italy, 2008.

[6] ***Shih-I Huang***, Shiuhpyng Shieh, SEA: Secure Encrypted-data Aggregation for Mobile Wireless Sensor Networks, IEEE Int'l Conf. on Computational Intelligence and Security, Harbin, China, 2007.

[7] ***Shih-I Huang***, Using Self Data Aggregated Sensor Networks to Achieve Higher Security in E-Society, Int'l Conf. on E-Society, Hong Kong, China, 2007.

[8] ***Shih-I Huang***, Shiuhpyng Shieh, Secret Searching in Wireless Sensor Networks With RFIDs, Information Security Conference, Kaohsiung, Taiwan, 2005

[9] ***Shih-I Huang***, Shiuhpyng Shieh, and S. Y Wu, Adaptive Random Key Distribution

Schemes for Wireless Sensor Network, International Workshop on Advanced Developments in Software and System Security, Taipei, Taiwan, 2003.

[10] Shiuhpyng Shieh, ***Shih-I Huang***, and Fu-Shen Ho, An ID-Based Proxy Authentication Protocol Supporting Public Key Infrastructure, The 2nd International Workshop for Asia Public Key Infrastructures, Taipei, Taiwan, 2002.

Book Chapter

[11] ***Shih-I Huang***, Shiuhpyng Shieh, S-Y Wu, Adaptive Random Key Distribution Schemes for Wireless Sensor Network,Computer Security in the 21st century, Springer

Patent

[12] Po-Yuan Ten, ***Shih-I Huang***, Hierarchical Multi-Layer Encryption system with forward and backward secrecy, TW and US patent pending, 2008 (pending)

[13] ***Shih-I Huang***, Shiuhpyng Shieh, System and Method for Secure Data Aggregation in Wireless Sensor Networks, TW, US, and CN patent pending. 2007 (pending)

[14] ***Shih-I Huang***, C-W Wang, Shiuhpyng Shieh, Light-Weight Authentication and Secret Retrieval Scheme and its applications for Wireless Ad Hoc Networks with limited resources, TW, US, and CN patent pending. 2007 (pending)

[15] ***Shih-I Huang***, System and Method for Secure Data Aggregation in Multilayered and Mobile Wireless Sensor Networks, TW, US, and CN patent pending. 2007 (pending)

[16] ***Shih-I Huang***, Po-Yuan Ten, System and Method for Hierarchical Multi-Layer Image Encryption in Surveillance Networks, TW and US patent pending. 2007 (pending)

[17] ***Shih-I Huang***, C-Y Wu, L-C Kuo, C-C Shen, An Energy Efficient Parking System, TW, US, and CN patent pending. 2006 (pending)