Journal of Systems Architecture xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A co-commitment based secure data collection scheme for tiered wireless sensor networks

Yang Zhao [a,*], Youtao Zhang [b], Zhiguang Qin [a], Taieb Znati [b]

[a] School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 610054, China
[b] Department of Computer Science, University of Pittsburgh, Pittsburgh 15260, USA

ARTICLE INFO

Article history:
Received 15 November 2009
Received in revised form 15 March 2010
Accepted 14 May 2010
Available online xxxx

Keywords:
Network security
Wireless sensor network
Security data collection
Co-commitment
Time-based query

ABSTRACT

Tiered wireless sensor networks (WSNs) have many advantages over traditional WSNs. However, they are vulnerable to security attacks, especially the attacks to the storage nodes that buffer and process the data readings from sensors. In this paper, we propose a secure data collection protocol SDC to support time-based queries in tiered WSNs. With small overhead introduced to data communication, SDC protects both data confidentiality and data integrity. In particular it employs data co-commitment scheme such that it can detect the seriousness of data losing and estimate the value of lost data in the network.

1. Introduction

A tiered wireless sensor network is a wireless sensor network (WSN) that, in addition to a large number of normal sensors, contains many *storage agent* nodes that are enhanced with larger storage and more computation resources, e.g. StarGate [1] and RISE [2] nodes. Sensors in tiered WSNs generate data readings and, due to limited storage, periodically send these readings to storage agents. Instead of forwarding all readings to the sink node, storage agent buffers received readings and return selected ones based on the query from the sink.

In recent years, WSNs have gained popularity as they provide a promising low-cost solution to a variety of challenges in both military and civilian domains, e.g. real-time traffic monitoring, military surveillance, and homeland security [3]. Comparing to traditional homogeneous WSNs, a tiered design shows better *cost-effectiveness*, *longevity* and *scalability* [4].

- A WSN application usually requires different functionalities: sensing, storing data, and data communication. Sensing typically requires a large number of nodes to ensure coverage, and few resources on each node; in contrast, data transmission and data storage require more system resources. Therefore deploying a homogeneous network tends to result in unnecessarily high cost of the network.
- Prolonging sensor network lifetime is critical due to the slow improvement of battery capacity, and the constraints of sensor physical sizes and cost. Kumar et al. studied the suitability of two hardware platforms – Mica motes and iPAQ for various tasks [5]. Their results confirmed that a tiered WSN with function partition prolongs network lifetime.
- The bandwidth and lifetime of a WSN should scale with increasing number of sensors. However, studies showed that hierarchical organizations provide better scalability over flat ones [6].

For many WSN applications, the sensed readings are sensitive and thus demand for data security – confidentiality, integrity, and freshness. However, the tight resource constraints of wireless sensors restrict the adoption of traditional computation-intensive security algorithms. In this paper we study the security issues that arise from time-based data collection in tiered WSNs, and focus on defending attacks to storage agents due to their importance in data collection. A compromised storage agent may (i) reveal its saved readings; (ii) compose forged data readings; (iii) drop important readings; and (iv) replay old data reading. Without carefully designed security enhancements, the above attacks can leave the network useless in a hostile environment.

In this paper we propose a secure data collection (SDC) protocol for security protection in tiered WSNs. The following summarizes our contributions.

* Corresponding author. Tel.: +86 2868198181.
E-mail addresses: zhaoyang@uestc.edu.cn (Y. Zhao), qinzg@uestc.edu.cn (Y. Zhang), zhangyt@cs.pitt.edu (Z. Qin), znati@cs.pitt.edu (T. Znati).

1383-7621/$ - see front matter © 2010 Published by Elsevier B.V.
doi:10.1016/j.sysarc.2010.05.010

(1) We study secure data collection in tiered WSNs and propose SDC to protect both data confidentiality and data integrity. In particular, we focus on defending message dropping attacks and propose to employ data co-commitment to detect and evaluate the abnormality during data collection.
(2) SDC introduces low-cost to existing data communication. When each sensor has n neighbours, the communication overhead between storage agents and normal sensors is $O(n(\log n + l_h + rl))$ and in practice less than 5% in most cases, where l_h and rl are two constant parameters.

The rest of this paper is organized as follows. We briefly review the related work in Section 2. Section 3 describes the system models and our design goals. We present the secure data collection protocol SDC in Section 4. The security and performance are analysed in Sections 5 and 6. Section 7 concludes the paper.

2. Related work

While security has been a long concern in WSNs, only few secure schemes have been designed for tiered WSNs.

The work most close to ours is a recently proposed scheme [7] that employs an *encoding number* method to preserve data privacy of range queries in tiered WSNs. In response to a range query from the sink, storage agents return selected data as well as encoding numbers that can be used by the sink to verify the integrity and correctness. However, a compromised storage agent may drop both data readings and the encoding numbers and just alleges no data readings to be received from the corresponding nodes. Usually the sensor network operations require low-power radio frequency (RF) communication, which cannot rely on using high power to boost the link reliability when operating under harsh radio conditions. Thus it is difficult to distinguish security attacks from channel error. The proposed scheme in this paper helps to defend message dropping attacks to the data collection system.

Secure data aggregation [8–10] targets at preventing data forging during data aggregation, and shares some design goals with our scheme. However, secure data aggregation schemes tend to incur high computation and communication overhead, and are not directly applicable for those data collection systems that need the original readings. For other related security designs, Shao et al. studied the problem of security and privacy protection in data-centric WSNs but their method employ homogeneous network architecture and can not apply to a tiered WSN [11].

3. System model and design goals

3.1. Network model

As shown in Fig. 1, a tiered WSN consists of three kinds of nodes, i.e. *regular sensors* (referred as sensors thereafter), *storage agents* and sink node. Sensors monitor interesting events and, due to their limited storage, periodically send raw readings to the storage nodes with a RF communication channel. Storage agents are enhanced with more storage and thus can keep readings of a long time interval. Instead of sending everything to the sink node, storage nodes buffer readings and respond with selected ones that match the query from the sink node.

3.2. Motivation

In this paper, we study time-based queries. For example, in a country boundary monitoring system, each sensor generates a reading recording the number of moving objects around it in the past 5 min, and reports all readings to its storage agent every 12 h. An inspector may issue a query as follows:

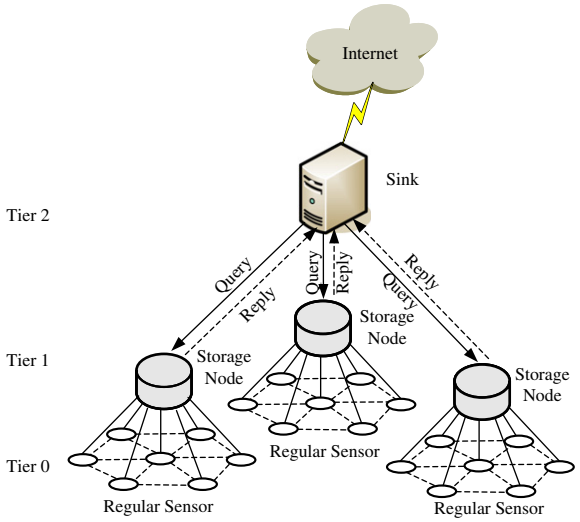

Fig. 1. Two tiered wireless sensor network.

select all
when time ∈ [5am, 12pm]
and day ∈ [today-1, today]

The interval between two consecutive reports from a sensor to its storage agent is referred as an *epoch*. We adopt a simple time synchronization mechanism [12] to align the epoch across different nodes. The report from low level sensors is as follows: $(t|s_i|v|d_1|\cdots|d_k)$, where t indicates the epoch count and v is a bit vector with indicating if a reading exists in the corresponding monitoring interval (5 min interval for the above example); s_i is the sensor ID; and $d_1 \cdots d_k$ are data readings. The query from the sink node is $(q_i, [t_a, t_b])$, where q_i is the query identification and $[t_a, t_b]$ indicates the time range that the data readings should be returned.

3.3. Attack model and security goals

We adopt Byzantine attacking model [13] to describe the malicious behaviours, i.e. an adversary, with polynomial bound on computation resources, may compromise sensors as well as storage agents. Once compromised, the node is under the full control of the adversary and may misbehave arbitrarily. We assume the sink node is always trustworthy and only a small fraction of other nodes may be compromised. We also assume the channel error rate has an upper limitation in a real WSN.

A security scheme cannot prevent a compromised sensor from disclosing its own readings, or forging false readings. However, the impact from compromising a regular sensor is limited for a WSN with a large number of sensors. Thus in this paper our scheme focuses on defending the following attacks from compromised sensors (in particular storage agents) to the readings of other nodes. The adversary tries to destroy the data collection process by the following ways without being detected.

- *Data confidentiality attack*. The adversary tries to get the raw readings from other sensors.
- *Data authentication attack*. The adversary tries to fool the sink node to accept forged readings.
- *Data integrity attack*. The adversary tries to drop some readings from other sensors.

Please cite this article in press as: Y. Zhao et al., A co-commitment based secure data collection scheme for tiered wireless sensor networks, J. Syst. Architect. (2010), doi:10.1016/j.sysarc.2010.05.010

The first two attacks are ease to be defended by encrypting the message and attaching a message authentication code (MAC). However, the data integrity attack is difficult to detect because the adversary may attribute it to the channel error or no data being generated by the victim sensor. The main idea of our scheme is to make use of a property of RF communication, i.e. a sensor can overhear its neighbours' data submitting activities. If each sensor reports some valuable information about its neighbours' data submitting and only the sink node can understand this kind of information, the adversary will not be able to simply drop the data readings from a victim sensor because the sink node may detect this data losses from the data readings of the victim sensor's neighbours with high probability. We will discuss the concrete method in Section 4.

4. Secure data collection protocol

4.1. Preliminaries

(1) *Master key based end-to-end encryption*
To protect *data confidentiality*, i.e. a sensor should not leak its readings to either its neighbours or its storage agent, we enforce master key based end-to-end data encryption [14]. The sink node randomly chooses a master key k_m and generates the shared keys for each node by calculating $H(k_m, s_i)$. To enhance the security, we use one-way key technique [15] to update the share key for each epoch, i.e. let $k_{i,t}$ be the share key of sensor s_i at epoch t, the shared key for next epoch is $k_{i,t+1} = H(k_{i,t}, t+1)$.
This key will be used for both message encryption and authentication. Symmetric encryption algorithms have relative low overhead and thus are suitable for low-cost sensors. In particular, we chose RC5 [16] due to its small code size [14].

(2) *Modular encryption operation*
In order to generate co-commitment without revealing sensitive information, we define a sample encrypting operation with modular arithmetic.
Let q be a large integer and k be a secret key. To encrypt message $m < q$ with k, we calculate

$$E_M(m,k) = (m+k) \bmod q \tag{1}$$

To decrypt a cipher-text c with k, we calculate

$$D_M(c,k) = (c-k) \bmod q \tag{2}$$

It is obvious that the modular encryption is additively homomorphic, i.e.

$$E_M(m_1,k_1) + E_M(m_2,k_2) = E_M(m_1+m_2, k_1+k_2) \tag{3}$$

If $m_1 + m_2 < q$, to decrypt $E_M(m_1+m_2, k_1+k_2)$ with (k_1+k_2), we can get a unique $m(<q) = m_1 + m_2$.

(3) *Data descriptor*
To assist data retrieving, we use a data descriptor, i.e. a bit vector, to denote the data distribution and the data range for each epoch. Assuming the value of interesting readings $val \in [v_{\min}, v_{\max}]$, we uniformly divide the whole range into 2^l-1 sub-ranges and associate a tag with each sub-ranges. The data descriptor consists of r (the number of monitoring interval in each epoch) tags. For example, there are 8 monitoring interval and 2^3-1 sub-range. A bit vector $v_{i,t} = 000001101110000011000111$ indicates sensor s_i has readings at 2, 3, 4, 6, 8 intervals in epoch t and corresponding range tags are, respectively, 1, 5, 6, 3, 7. It should be noted that 000 means no interesting reading generated. Data descriptor has two main functions: Firstly, it provides the information of data distribution in an epoch. Secondly, it marks the value range of each reading.

(4) *Notations*
The following notations are used in the description of the protocol.

- t is the count of an epoch,
- q is a large prime number,
- s_i refers to a sensor where i is the node identification,
- $\mathscr{S}$ refers to the set of sensors,
- $\mathscr{A}$ refers to sink node,
- $\mathscr{B}$ refers to the set of storage agents,
- $\mathscr{B}_i$ refers to the storage agent of sensor s_i,
- $\mathscr{N}_i$ refers to the neighbour set of sensor s_i,
- $\mathscr{N}_{i,t}$ refers to a neighbour set being overheard by sensor s_i at epoch t,
- $k_{i,t}$ is the share key of sensor s_i at epoch t,
- $v_{i,t}$ is the data descriptor of sensor s_i at epoch t,
- $C_{i,t}$ is the data certificate of sensor s_i at epoch t,
- $CC_{i,t}$ is the co-commitment of sensor s_i at epoch t,
- $H(\cdot,\cdots)$ is a secure hash function [17],
- $I_a \to I_b{:}m$ denotes entity I_a use a broadcast channel to send message m to entity I_b,
- $I_a \Rightarrow :m$ denotes entity I_a broadcast message m,
- $m_1|m_2$ denotes concatenating messages m_1 and m_2,
- $\mathscr{E}(m, key)$ denotes using symmetric algorithm to encrypt message m with key,
- $E_M(m, key)$ denotes using modular arithmetic to encrypt message m with key.

4.2. The co-commitments scheme

Due to the broadcast property of RF communication, the messages sent out from one sensor may be overheard by its neighbours. If each sensor reports the overheard data submission of its neighbours, then the sink node can cross-check if some data are lost. For example, s_i reports that its neighbour s_j submitted data in epoch t but there is nothing received by the sink node. In this case the sink node knows that some data were lost during the data collection. To provide more useful information and guarantee the privacy, we propose a *co-commitment* based scheme that works as follows.

First, for every sensor $s_i \in \mathscr{S}$ that has readings in epoch t, it generates data descriptor $v_{i,t}$ and calculates

$$od_{i,t} = E_M(v_{i,t}, k_{i,t}) \tag{4}$$

$$C_{i,t} = H(i|t|v_{i,t}|k_{i,t}) \tag{5}$$

Then the sensor attaches $od_{i,t}$ to the submitted data message. After overhearing the data submission actions of its neighbours in epoch t, sensor s_i calculates

$$sum_{i,t} = \sum_{s_j \in \mathscr{N}_{i,t}} ob_{j,t} \tag{6}$$

$$CC_{i,t} = (sum_{i,t}|s_{j_1}|C_{j_1}|s_{j_2}|C_{j_2}|\cdots) \tag{7}$$

where s_{j_n} denotes the identification of s_i's neighbours. $CC_{i,t}$ is a co-commitments from s_i that the sink node can use to detect the data loss of s_i's neighbours. Furthermore, the sink node may retrieve the descriptor of lost data to meet certain requirements (see details in Section 6).

Please cite this article in press as: Y. Zhao et al., A co-commitment based secure data collection scheme for tiered wireless sensor networks, J. Syst. Architect. (2010), doi:10.1016/j.sysarc.2010.05.010

4.3. Protocol description

SDC protocol contains three phases: *system setup*, *data submission and co-commitment generation*, and *time-based query and data verification*.

(1) *Phase I: System setup*

For a short period after deployment, the network is reliable such that each sensor initializes to (i) determine its storage agent and (ii) detect its neighbours and reports the neighbour list to the sink node. The detailed steps are

Step 1:

$\mathscr{B} \Rightarrow$: $hello(storage\ agent)$

$\mathscr{S} \Rightarrow$: $hello(senor)$

Step 2:

$s_i \rightarrow \mathscr{B}_i : (0|i|\mathscr{E}(m_i, k_{i,0}))$

$m_i = (i_1|i_2|\cdots|i_n)$

where $i_1, i_2, \ldots, i_n$ are the identification of s_i's neighbours.

Step 3:

$\mathscr{B} \rightarrow \mathscr{A} : (\mathscr{E}(m_1, k_{1,0})|\cdots|\mathscr{E}(m_{|\mathscr{S}|}, k_{|\mathscr{S}|,0}))$

In this phase each node (sensor or storage agent) broadcasts several *hello* messages. A sensor collects these messages and knows its nearby storage agents and sensors. We assume each sensor has at most N neighbours and at least one storage agent. If there are more than one storage agents, the sensor randomly selects one storage agent.

These IDs are encrypted and sent through the storage agent to the sink node. At the end of this phase the sink node can establish a mapping table recording the neighbours of each sensor (Table 1). Considering the dynamic behaviour of WSNs, the sink node needs to periodically invoke the system setup process to update the mapping table. Since performing such a process requires additional expenditure of energy, the sink node must deliberate whether to do it. In this paper we assume the sensors are quasistatic after being scattered, therefore the neighbour relationship between sensors are relatively stable. It means that the sink node only needs to invoke the system setup process when there are some new sensors to be deployed.

(2) *Phase II: Data submission and co-commitment generation*

Step 1:

Data preparation

We assume one epoch has r subintervals and the readings have $2^l - 1$ range tags. Since a sensor may or may not produce a reading in each subinterval, a rl-bits data descriptor $v_{i,t}$ is generated at the end of each epoch. The sensor then encrypts the data

$$\mathscr{E}(data_{i,t}, k_{i,t}) = \mathscr{E}(v_{i,t}|data_1|data_2|, \ldots, k_{i,t}) \tag{8}$$

and calculates a certificate $C_{i,t}$ using one-way hash function.

$$C_{i,t} = H(i|t|v_{i,t}|k_{i,t}) \tag{9}$$

Since the data descriptor is also a sensitive information, so we use the modular encryption operation to hide it in $od_{i,t}$. The treated data descriptor can prevent the neighbours know the relevant information, so we call it *obfuscated data descriptor*, only the nodes who know $k_{i,t}$ can retrieve the data descriptor when needed.

$$od_{i,t} = E_M(v_{i,t}, k_{i,t}) = (v_{i,t} + k_{i,t}) \bmod q \tag{10}$$

where $v_{i,t}$ is the data descriptor of s_i's readings in the epoch t and q is a large integer to ensure $q > n \cdot 2^{rl}$.

Step 2:

Data submission

In this step, each sensor submits data readings to its corresponding storage agent. That is, a sensor s_i reports

$$s_i \rightarrow \mathscr{B}_i : (\underbrace{i|t|od_{i,t}|C_{i,t}|}_{Message\ \ header}\mathscr{E}(data_i, k_{i,t}))|H(*, k_{i,t})$$

where $H(*, k_{i,t})$ is the message authentication code.

Step 3:

Co-commitment generation

$$s_i \rightarrow \mathscr{B}_i : \mathscr{E}(CC_{i,t}, k_{i,t})|H(*, k_{i,t})$$

Each sensor buffers $n(<N)$ overheard reports from surrounding sensors in step 2, where n is a predefined parameter and the method for choosing proper number are offered in the following section. It constructs a co-commitment $CC_{i,t}$ from obfuscated data descriptors and corresponding certificates.

$$sum_{i,t} = \sum_{s_j \in \mathscr{N}_{i,t}} od_{j,t} \bmod q \tag{11}$$

$$CC_{i,t} = (i|t|sum_{i,t}|a_1|C_1|a_2|C_2|\cdots) \tag{12}$$

where $|\mathscr{N}_{i,t}| = n$ is a subset of $\mathscr{N}_i$, $a_1, a_2, \ldots, a_n$ are the indices of its neighbours in the recorded neighbour set and $C_1, C_2, \ldots, C_n$ are respective certificates. Then s_i encrypts this information and sends it through its storage agent to the sink node.

(3) *Phase III: Time-based query and data verification*

Our scheme supports time-based query service for authorized users. The sink acts as the intermediary between the user and storage agents. When the sink receives a query from an authorized user, it propagates the query to storage agents who respond with required readings.

Step 1:

$\mathscr{A} \rightarrow \mathscr{B} : Q_i = (q_i|[t_a, t_b])$

where q_i is the query ID and $[t_a, t_b]$ indicates the time range in which interesting readings should be returned.

Step 2:

$\mathscr{B} \rightarrow \mathscr{A} : \mathscr{M}_{t \in [t_a, t_b]}$

where $\mathscr{M}_{t \in [t_a, t_b]}$ is the set of readings in $[t_a, t_b]$.

Step 3:

$\mathscr{A} \rightarrow usr_i : (t_a|i_1|v_{i,t_a}|data1|\cdots|i_2|v_{i,t_a}|data1|\cdots), \cdots,$
$(t_b|i_1|v_{i,t_b}|data1|\cdots|i_2|v_{i,t_b}|data1|\cdots)$

The sink then verifies the data correctness and integrity before sending data back to the user. Since both the encrypted data and the co-commitments are returned from storage agents, the sink first decrypt the data and get $v'_{i,t}$ and data serials $data1$, $data2$, ..., It calculates

$$C'_{i,t} = H(i|t|k_{i,t}|v'_{i,t}) \tag{13}$$

and verify if the received data generates the same certificate. A warning message is generated if there is a mismatch.

Table 1
The neighbour mapping.

Sensor ID	1	2	⋯	n
s_1	$i_{1,1}$	$i_{1,2}$	⋯	$i_{1,n}$
s_2	$i_{2,1}$	$i_{2,2}$	⋯	$i_{2,n}$
⋯	⋯	⋯	⋯	⋯

5. Security analysis

The SDC protocol aims at ensuring data confidentiality, authentication and integrity even if an adversary can compromise a small fraction of sensors and storage agents. This section discusses that SDC protocol how matches such security requirements separately.

Please cite this article in press as: Y. Zhao et al., A co-commitment based secure data collection scheme for tiered wireless sensor networks, J. Syst. Architect. (2010), doi:10.1016/j.sysarc.2010.05.010

5.1. Data confidentiality

We first discuss how to ensure data confidentiality, i.e. preventing data from being disclosed to storage agents or other sensors. For this purpose, all the readings are encrypted with a secret key that is shared by the sink node and the sensor. The session key used at each epoch is generated using one-way chaining hash [15]. Therefore, compromising a sensor s_i will not disclosure its prior data readings. Since the data readings are time-sensitive, we also need to hide the time distribution of reported readings. In our scheme, we attach the encrypted data count in each epoch such that the data distribution is also protected.

5.2. Data authentication

We then discuss how to ensure data authentication and freshness. Since a compromised storage agent may behave maliciously, it can send back arbitrary data as a response to the query. However, false readings can be easily detected since the storage agent does not have the secret keys of non-compromised sensors. Stale data and outdated data can also be easily detected as the epoch time is enclosed as part of the reply, and the session key gets updated in every epoch.

5.3. Data integrity

Since the message integrity is protected by MAC, a malicious storage node cannot tamper with the contents of its received messages. For data integrity we mainly focus on how to detect message dropping attack, i.e. a compromised storage agent drops all messages that are related to some interesting readings. In our solution, sink node will cross-check the data by using co-commitments in the data reporting phase. The discussion in the next section shows that the sink node can detect the message dropping attacks with high probability (close to 100%) when the success rate of message transmission is more than 0.8 and the number of neighbours is more than 4. This indicates the sink can accurately calculate the message loss rate based on the detection process, and thus detect the message dropping attacks with high confidence.

However, an adversary may try to destroy the data retrieving process by misleading the sink with bad information: the compromised nodes use a selected bit vector to replace the actual obfuscated data descriptor. Obviously if the sink uses such a bit vector to retrieve the descriptor of lost data the results will be wrong. In our protocol the retrieved descriptor gets double checked using certificate $C_{i,t}$, the wrong results get detected unless the adversary can solve the following problem:

Given $i, t, C_{i,t} = H(i|t|v_{i,t}|k_{i,t})$, find $v'_{i,t}$ such that

$$H(i|t|v_{i,t}|k_{i,t}) = H(i|t|v'_{i,t}|k_{i,t}) \tag{14}$$

Since the adversary cannot get the private keys from non-compromised sensors, solving above problem is NP-hard. When sink node detects such an inconsistency, it locates the compromised nodes by tracing the providers of related co-commitments. An adversary may also use a brute force method to prevent the sink from retrieving a sensor's data readings by dropping all data reports of its neighbours. This method can be easily detected by the sink because the data loss is concentrated in some special area which forms a black hole.

6. Performance analysis

6.1. Message losing detection capability

If some data readings are lost (either due to lossy channel or message dropping attack), the sink node uses the received co-commitments to detect and evaluate its severity. The key problem of detection is how to distinguish the channel error and the message dropping attack.

Our main idea is based on the following assumption, i.e. in a particular WSN the channel error rate will be distributed in a certain range if there are no attacks. Therefore, if we detect the message loss rate exceed the upper bound of channel error rate, it is possible that there is a message dropping attack. The following is the method of how to detect a message loss incident: Assuming s_i is the sensor with no data, the sink first identifies s_i's neighbour list, and then decrypts the co-commitments of these sensors. If s_i is not overheard by any of them, then it is most likely that no interesting reading was reported by s_i. If s_i was overheard by some of its neighbours but not the storage agent, it is most likely that there are some messages lost. Based on such detection the sink can calculate the message loss rate, and evaluate how severe it was. If the loss rate exceeds the threshold that the network can tolerate, an alarm message will be triggered. According to [18], the channel error rate for WSN applications is usually tolerable in the range of 5–10%.

As discussed above the key problem of our scheme is how to accurately detect a message loss incident. In our scheme, each sensor summarizes and reports its overheard obfuscated data descriptors of its neighbours. If a sensor reports interesting readings, but messages have p_1 and p_2 probabilities (which cover the message loss due to either lossy channels or security attacks) to be transmitted or overheard, respectively, between sensors and storage agents, and between storage agents and the sink, then the probability of the data loss being detected is

$$\Pr(\text{detection}) = 1 - (1 - p_1 p_2)^n \tag{15}$$

i.e. the loss can be detected when at least one neighbour reported s_i "had data" in its co-commitment and the sink receives this information. Assume $p = p_1 = p_2$, Fig. 2 shows the detection rate under different p. Apparently, under the same p, the more the number of neighbours is, the higher the detection rate is. If $p = 0.8$ and $n = 4$, then the detection rate approaches 1.

6.2. Data retrieving capability

We then discuss how to retrieve the data descriptor of lost readings when the sink did not receive the report message from a sensor, which is important for evaluating how severe the data lost was.

Theorem 1. *SDC can recover the data descriptor of a data-losing node s_a if the following two conditions are both satisfied (i) one of s_a's*

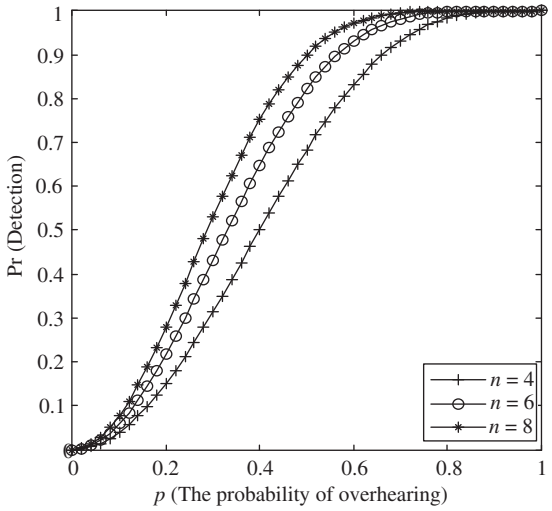

Fig. 2. Detection rate under different conditions.

neighbour s_i overheard its data descriptor and (ii) all other neighbours of s_i are either recovered, or have their readings received by the sink.

We omit the proof whose sketch is as follow. The sink computes

$$
\begin{aligned}
&R = \sum_{s_j \in \mathcal{N}_{i,t}} k_{j,t} \bmod q \\
&\because sum_{i,t} = \sum_{s_j \in \mathcal{N}_{i,t}} od_{j,t} \bmod q = \sum_{s_j \in \mathcal{N}_{i,t}} (v_{j,t} + k_{j,t}) \bmod q \\
&\therefore \sum_{s_j \in \mathcal{N}_{i,t}} v_{j,t} = kq + sum_{i,t} - R \\
&\text{and } \because 0 < \sum_{s_j \in \mathcal{N}_{i,t}} v_{j,t} \leqslant n \cdot 2^{rl} < q \\
&\therefore \sum_{s_j \in \mathcal{N}_{i,t}} v_{j,t} = sum_{i,t} - R, \ \text{when } sum_{i,t} - R > 0 \\
&\text{or } \sum_{s_j \in \mathcal{N}_{i,t}} v_{j,t} = q + sum_{i,t} - R, \ \text{when } sum_{i,t} - R < 0
\end{aligned}
\tag{16}
$$

According to the assumption, the sink knows readings except s_a, so the sink can find out

$$
v_{a,t} = \sum_{s_j \in \mathcal{N}_{i,t}} v_{j,t} - \sum_{s_j \in \mathcal{N}_{i,t}/s_a} v_{j,t} \tag{17}
$$

After receiving the encrypted data readings and co-commitments from storage agents, the sink node constructs a *message overhearing relationship graph* (MOG) as shown in Fig. 3. In the graph, the white and gray nodes represent, respectively, those sensors with and without returned data. With incoming edges denoting message overhearing relationship, the integer value in a white sensor summarizes its overheard obfuscated data descriptor, i.e. each white node s_i has a weight $W(s_i) = \sum_{s_j \in \mathcal{N}_{i,t}} v_{j,t}$, and each edge $e_{i,j} = s_i \to s_j$ has a weight $W(e_{i,j}) = v_{i,t}$. To simplify the demonstration, we use integer to represent $W(s_i)$ and $W(e_{i,j})$ in Fig. 3.

To find out how many data readings were lost, the sink node applies the Algorithm 1 to iteratively recover the data descriptor of gray nodes. When a white node has only one incoming edge from a gray node, the sink node can subtract the sum of the weight of known edges from the weight of node, then the data descriptor of corresponding gray node can be recovered. We change it to a white node and iterate until no white node has one incoming edge from a gray node. Fig. 4 illustrates an example for the process of data retrieval. Gray nodes s_i, s_b, and s_g are recovered consecutively.

Algorithm 1. Input: a directed graph $G = (V,E)$, like Fig. 3. Output: the unknown weight of the dotted edges.

Procedure:

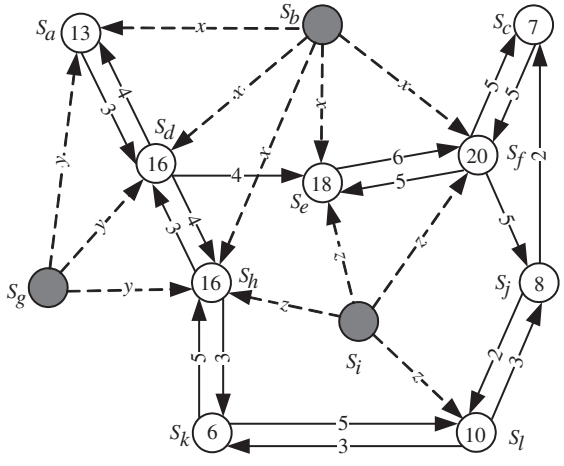

Fig. 3. Message overhearing relationship graph.

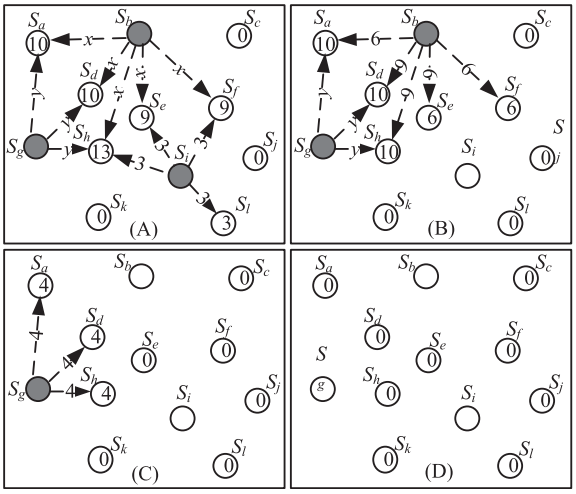

Fig. 4. Recovering the data descriptor of gray nodes.

```
while (E≠φ)
  { edges=|E|
    For each  e_{x,y}∈E
    if W(e_{x,y}) is known
      { delete e_{x,y};
        W(s_y)=W(s_y)−W(e_{x,y});
      }
    For each s_y∈V and W(s_y)≠0
      { if s_y only has one edge that W(e_{x,y}) is unknown
        output count_{x,t}=W(e_{x,y})=W(s_y);
        W(s_y)=0;
      }
      If edges=|E| break;
  }
```

If the descriptor of lost data readings in one epoch from a sensor s_i is $v_{i,t}$, the sink can know there are how many readings generated by s_i and it can further estimate the value of lost data based on the range tags. For example, if the value of readings $val \in [a, 8a]$ and the tag of it is "110", the sink will know $6a \leqslant val < 7a$.

6.3. The efficiency of SDC

6.3.1. Simulation parameters setting

To study the effectiveness of adopting co-commitment in SDC, we simulated a tiered WSN with 2500 sensors randomly deployed in a 100 m × 100 m field. There are 50 storage agents each of which roughly controls 50 sensors. The detection range of a sensor is 10 m. We set the time span of each epoch to be 80 s, and the readings to be generated per 10 s, and the range of reading size is specified from 0.25 kb to 2.5 kb. Since in WSN applications the channel error rate is usually in the range of 5–10% (or slightly higher) [18], in our experiments we vary the message loss rate from 5% to 35%. Furthermore, we assume the rate of "bad" sensors is 10%, which means the messages form "bad" sensors cannot be used to retrieve the descriptor of lost data, and measure the effectiveness of our proposed algorithm.

The simulation data readings are randomly generated by MATLAB 2007 and we implement the retrieving algorithm with Microsoft VC++ 6.0. The program runs on a personal computer: Pentium(R) IV 2.80 GHz, 1G, Windows NT. The relevant cryptographic algorithms like encryption and hash are all from OpenSSL library [19].

We first discuss how to choose n the number of neighbours. As analysed before, if n is too small the detection rate of data losing will lower (see Fig. 2). If n is too big, the irretrievable rate will in-

crease because the probability of one sensor having two irretrievable neighbours will be higher. According to Theorem 1, SDC can recover the data descriptor of a data-losing node s_a in the first round of invoking algorithm, if one of its neighbour s_i correctly report its obfuscated data count and all the neighbours of s_i have their correct readings head by sink. Therefore, the probability of retrieving is as following.

$$\Pr(\text{retrievability}) = n((1-p)(1-p_0))^n \tag{18}$$

where p is the message loss rate and p_0 is the sensor "bad" rate. In our experiment, p_0 is fixed and equals to 0.1.

Fig. 5 shows the results with deferent number of neighbours. From the figure, we found that it is important to determine an appropriate neighbour threshold. Our experiments showed that setting n to be 4 gives the best results when p is between 5% and 20%. The actual retrieving rate is slightly higher than the results shown in Fig. 5 because there are some data-losing nodes that can be retrieved by the subsequent rounds of invoking Algorithm 1.

We recorded the rounds of invoking algorithm with different (n,p) and summarized the results in Fig. 6. The number of rounds increases with higher data loss probability. This is expected as more messages are lost and need to cross-checked when the data loss probability is high. It requires larger number of checking rounds when there are more neighbours. This is due to the fact that with more neighbours, a missing report may be included in more co-commitments.

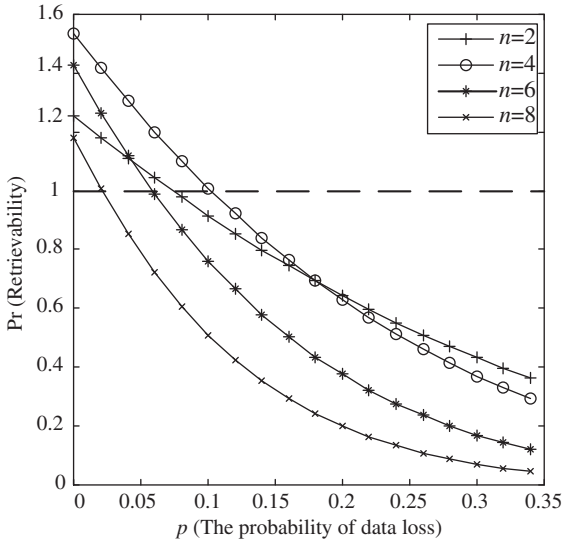

Fig. 5. The retrieving rate with different number of neighbours n.

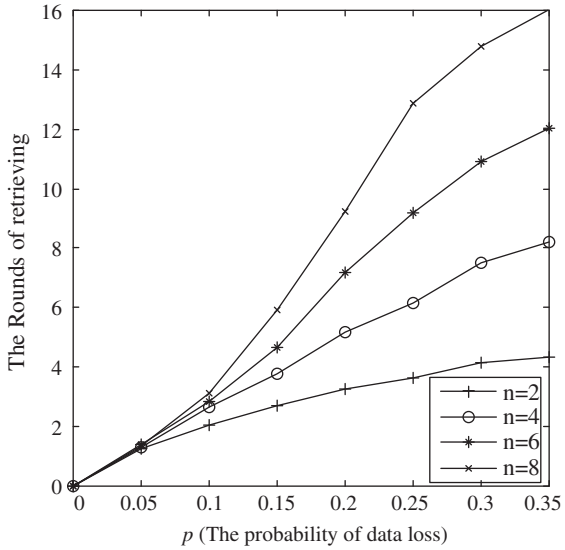

Fig. 6. The rounds of invoking algorithm with different.

6.3.2. Data retrieving efficiency

We first define a parameter $\rho = \frac{N_i}{N_l}$ which is the percentage of irretrievable sensors, where N_i is the number of sensors whose readings cannot be recovered, and N_l is the number of sensors from which data was not received by the sink. We changed the number of neighbours n that each sensor has, and the data loss rate p. To detect and evaluate the data loss, the sink node invokes the Algorithm 1 and retrieves the number of lost data readings. Fig. 7 reports the averaged results over 100 runs. From the figure, it is almost always recoverable when data loss probability is low.

6.3.3. System overhead

Comparing to the communication overhead in WSNs, the computation overhead is modest, for example in [20] the authors stated that the transmission of 1 bit consumes about as much power as executing 1000–8000 instructions for the Mica2 mote. Thus we focus on evaluating the communication overhead of the protocol. In the protocol, the data report from sensors contains data readings and co-commitment. The length of the former is rm while that of the latter is $n(\log n + l_h + rl)$ where m is the length of data readings generated in each interval, l_h is the length of hash function $H(\cdot,\cdots)$'s output and rl is the length of data descriptor. We

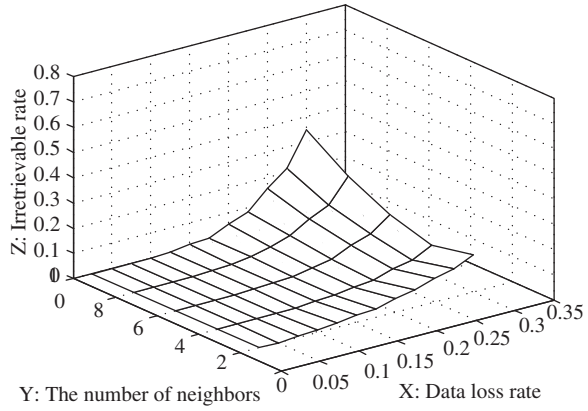

Fig. 7. The irretrievable rate with different (n,p).

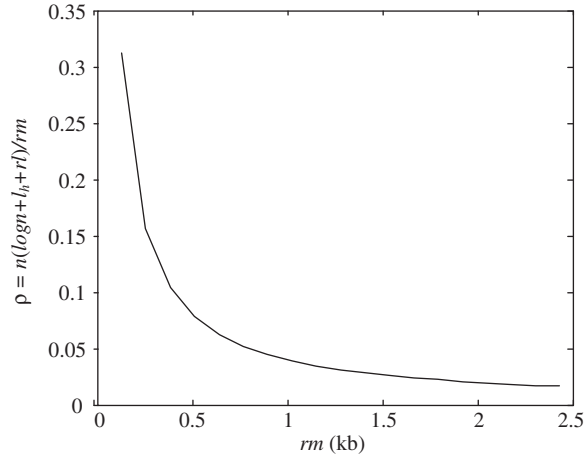

Fig. 8. The communication overhead.

Please cite this article in press as: Y. Zhao et al., A co-commitment based secure data collection scheme for tiered wireless sensor networks, J. Syst. Architect. (2010), doi:10.1016/j.sysarc.2010.05.010

Table 2
The run duration of data retrieving (ms).[a]

(n,p)	0.05	0.10	0.15	0.20	0.25	0.30	0.35
2	5.4	5.6	6	5.6	5.5	6.3	6.6
4	5.8	6.9	7.3	7.0	5.9	6.2	6.4
6	7.7	6.9	7.2	7.6	7.4	7.9	7.9
8	9.5	8.7	8.0	9.6	7.8	9.8	9.4

[a] As measured on a personal computer: Pentium(R) IV 2.80 GHz, 1G, Windows NT.

chose $l_h = 64$ bits similar as [21]. Fig. 8 plots the ratio between data readings and co-commitment when $n = 4$, $l_h = 64$ bits, $rl = 24$ bits. As we can see the overhead decreases significantly when the size of data report increases, so the extra communication overhead introduced by co-commitment is negligible.

Since there are no high-overhead operations in retrieving algorithm, the run duration of it is very short. Table 2 shows our experiment's result.

7. Conclusion

While tiered WSNs provide better energy tradeoffs, the storage agents in such networks might become the attacking targets, makes it a challenging task to perform secure data collection. In this paper, we propose a co-commitment scheme to support time-based data queries. In the face of data loss, our protocol strives to distinguish with high confidence the security attacks from normal communication signal losses without heavy system overhead.

Acknowledgments

The work presented in this paper was partially supported by the National Natural Science Foundation of China (60473090), the Ph.D. Program Foundation, Ministry of Education of China (20050614018), and US National Science Foundation (NSF) under grants CCF-0641177, CNS-0720595.

References

[1] Storage gateway (SPB400), Online. <http://www.xbow.com>.
[2] RISE project, Online. <http://www.cs.ucr.edu/~rise>.
[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (August) (2002) 102–114.
[4] M. Ilyas, I. Mahgoub, Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, CRC Press, 2004, pp. 235–256, ISBN: 0849319684 9780849319686.
[5] R. Kumar, V. Tsiatsis, M.B. Srivastava, Computation hierarchy for in-network processing, in: Proceedings of the 2nd International Workshop Wireless Networks Applications (WSNA'03), San Diego, CA, September 2003, pp. 68–77.
[6] O. Dousse, P. Thiran, M. Hasler, Connectivity in ad-hoc and hybrid networks, in: Proceedings of the IEEE Infocom2002, New York, USA, June 2002, pp. 1079–1088.
[7] B. Sheng, Q. Li, Verifiable privacy-preserving range query in two-tiered sensor networks, in: Proceedings of the IEEE Infocom2008, Phoenix, USA, 2008, pp. 46–50.
[8] L. Hu, D. Evans, Secure aggregation for wireless networks, in: Proceedings of the Workshop on Security and Assurance in Ad hoc Networks, Orlando, USA, January 2003, pp. 384–391.
[9] B. Przydatek, D. Song, A. Perrig, SIA: secure information aggregation in sensor networks, in: Proceedings of the Sensys'03, Los Angeles, USA, 2003, pp. 255–265.
[10] W. He, X. Liu, H. Nguyen, K. Nahrstedt, T.F. Abdelzaher, PDA: privacy-preserving data aggregation in wireless sensor networks, in: Proceedings of the IEEE Infocom2007, Anchorage, Alaska, USA, 2007, pp. 2045–2053.
[11] M. Shao, S. Zhu, W. Zhang, G. Cao, pDCS: security and privacy support for data-centric sensor networks, IEEE Transactions on Mobile Computing 8 (2009) 1023–1038.
[12] W. Su, I.F. Akyildiz, Time-diffusion synchronization protocol for sensor network, IEEE/ACM Transactions on Networking 13 (2005) 84–397.
[13] L. Lamport, R. Shostak, M. Pease, The byzantine general problem, ACM Transactions on Programming Languages and Systems (TOPLAS) 4 (1982) 382–401.
[14] A. Peering, R. Szewczyk, V. Wen, D. Cullar, J.D. Tygar, Spins: security protocols for sensor networks, Wireless Networks 8 (2002) 521–534.
[15] L. Lamport, Password authentication with insecure communication, Communications of the ACM 24 (1981) 770–772.
[16] R.L. Revist, The RC5 encryption algorithm, in: Proceedings of the 1st Workshop on Fast Software Encryption, 1995, pp. 86–96.
[17] M. Bellare, R. Canetti, H. Krawczyk, Keying hash functions for message authentication, in: Advance in Cryptology – CRYPTO'96, 1996, pp. 1–15.
[18] C.Y. Wan, A.T. Campbell, L. Krishnamurthy, PSFQ: a reliable transport protocol for wireless sensor networks, in: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, USA, 2002, pp. 28.
[19] OpenSSL, The Open Source Toolkit for SSL/TLS Project, Online. <http://www.openssl.org>.
[20] J. Hill, R. Szewczyk, S. Hollar, A. Woo, D. Culler, K. Pister, System architecture directions for networked sensors, in: Proceedings of the ACM ASPLOS IX, Cambridge, USA, 2000, pp. 93–104.
[21] S. Mauw, I.V. Vessem, B. Bos, Forward Secure Communication in Wireless Sensor Networks, Security in Pervasive Computing, vol. 3934, Springer, Berlin/Heidelberg, 2006, pp. 32–42, ISBN: 978-3-540-33376-0.

Yang Zhao is a Ph.D. Candidate at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests are in the area of networking security and e-commerce protocol.

Youtao Zhang is an Assistant Professor in the Computer Science Department, University of Pittsburgh. He completed his Ph.D. in Computer Science at the University of Arizona in 2002. His research interests are in the area of the computer security, program analysis, and computer architecture. He is the recipient of US NSF Career Award in 2005, distinguished paper award in ICSE 2003, most original paper award in ICPP 2003. He is a member of the ACM and the IEEE.

Zhiguang Qin is a Professor of School of Computer Science and Engineering, University of Electronic Science and Technology of China. He completed his Ph.D. in Computer Application at University of Electronic Science and Technology of China in 1996. His research interests are in the area of information security, e-commerce, and intelligent traffic system. He is a member of the IEEE.

Taieb Znati is a Professor in the Computer Science Department, University of Pittsburgh. He obtained the M.S. (Computer Science) from Purdue University, and the Ph.D. (Computer Science, 1988) from Michigan State University. His current research interests focus on the design of network level channel abstractions for real-time communication networks to support multimedia environments, the design and analysis of medium access control protocols to support distributed real-time systems, and the investigation of fundamental design issues related to distributed systems in the areas of machine learning, cognitive modeling, problem solving, and analogical reasoning.