2008 14th IEEE International Conference on Parallel and Distributed Systems

Time-based Privacy Protection for Multi-attribute Data in WSNs

Baowei Wang[1], Xingming Sun[1], Xinbing Wang[2], Bin Xiao[3]
[1]*School of Computer & Communication, Hunan University, Changsha, China*
[2]*Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China*
[3]*Department of Computing, Hong Kong Polytechnic University, Hong Kong*
{wbw.first, sunnudt}@163.com, xwang8@sjtu.edu.cn, csbxiao@comp.polyu.edu.hk

Abstract

Wireless sensor networks become ubiquitous to collect people's information in many people-centric applications, such as, health care, smart space and public safety. Because any misusage of these personal data might result in the leakage of privacy, it is expected that the data requesters can only access to the data what they are entitled to read. Based on a revised hash chain technique, we proposed a novel Time-based Privacy Protection (TPP) scheme for multi-attribute data in WSNs. In the scheme, all the personal data are divided into 2-D subspaces representing data attribute and generation time. Data in each subspace is encrypted with a sub-key before its transmission to the sink. Anyone who wants to read data attribute at a particular time must get the corresponding sub-key from the sender node. TPP can generate a sub-key for data in each subspace in an efficient manner in terms of less sub-key generation time and low memory space usage. The simulation results show that the schemes can be applied to the resource limited WSNs efficiently.

1. Introduction

Wireless sensor network technology has the potential to change the way we live, work and do business [1, 2]. It offers economically viable solutions for a variety of people-centric applications, such as, health care, working space automation and public safety [3, 4, 5]. Wireless sensor networks even might be quickly becoming a vital part of our infrastructure to enrich lives and make processes easier. However, there are some potentially negative aspects of these networks, and one of them is the prospective loss of individual privacy.

In the people-centric wireless sensor networks, we refer to the data privacy as 'informational self-determination'. The data generated by sensors have the potential to threaten people's privacy if misused or mishandled. Hence, privacy protection is a critical issue which might impact the acceptance of a WSN project from a large community. The deployment of sensor network in the convenience service applications must meet increasingly stringent security and privacy require-ments. In current wireless sensor networks, once the personal data are transmitted to the sink, it will be actually out of the owner's control. It might be unacceptable to most people and might impact the deployment of a WSN system. People expect that only the permitted privacy information in a given time slot to be read by others. Thus, strict privacy mechanisms must always be in place to prevent discretionary use of the personal data.

Although some efforts have been carried out to mitigate the risks of privacy exposure [7, 13], privacy protection remains to be a challenging issue in current WSNs when we consider it in both the attribute and time context. In traditional wireless sensor networks, all privacy data use a completely identical protective measure. If a person can query part of data, he is able to query the others. This might result in many problems. Actually, many kinds of data are collected continuously by a WSN. Different kinds of data may have different usage. Meanwhile, the same kind of data collected in different time should be used with different permissions.

In this paper, we proposed a Time-based Privacy Protection (TPP) scheme for multi-attribute data in WSNs. From an initial key and a chosen hash chain function, TPP scheme can efficiently generate 2-dimensional keys to provide privacy in WSNs with guaranteed security. To the best of our knowledge, there are no similar researches so far. A side effect of the scheme is that we can carry out the access control in the sink as well for queries from ordinary sensor nodes. We also propose improved schemes to make it adaptable to different application scenarios. The simulation results show that TPP can be applied to the

1521-9097/08 $25.00 © 2008 IEEE
DOI 10.1109/ICPADS.2008.67

resource limited WSNs efficiently, requiring less key computation time and small storage space. The storage space consumption is extremely low, even only requiring a single seed key stored on each sensor node.

The remainder of this paper is organized as follows. The next section discusses the problem statement. Section 3 describes the basic design of Time-based Privacy Protection scheme, and further shows the improved schemes. Performance evaluation is presented in Section 4. Section 5 presents some related works. We conclude the work in Section 6.

2. Problem statement

Wireless sensor networks are used for collecting people's information in many people-centric applications, such as home automation, health care and public safety. Because sensor nodes have not enough storage space, it might be unrealistic that all the data are stored on the sensor nodes in many applications. The alternative feasible approach is that all of the privacy data collected by WSNs are transmitted to the sink and stored on it. But this might result in another critical problem.

In traditional wireless sensor networks, the sink is considered as the core of the whole sensor network. It can query and access to any kinds of data from any sensor nodes at any time and handle them discretionarily. That is, the personal data collected by sensor nodes are actually out of the owner's control. If any of these personal data are misused or mishandled, it might result in the leakage of privacy. It might be unacceptable to most people and might impact the deployment of a WSN system. People might expect only the right privacy information at the right time to be queried by any persons who have been mandated firstly. To the best of our knowledge, there are still no methods that have been proposed to restrict the data access competence of the sink to protect people's privacy for ensuring the acceptance of the application systems from a larger community.

Thus, above existing challenges pose the problem to design a more secure system for the purpose of privacy protection in people-centric WSNs, where data should be read according to their given *attribute* and *time slot*. Future privacy protection schemes should consider the feelings and requirements of the users' who are the owners of the privacy data. This is essential to ensure users acceptance to the systems. They should focus on the 2-D characteristics of private data, which will be formulated in Subsection 3.1. The owners of the privacy data could control which data can be accessed by the sink. In addition, they must be adaptive to the resource limited sensor nodes. It is equally important to consume as little energy for key generation and small storage space.

3. Time-based privacy protection for multi-attribute data

In order to address the above issue, we argue that the time factor should be considered in future privacy protection schemes. Thus, we propose the 2-D characteristic of the multi-attribute private data. Based on that, we present the Time-based Privacy Protection scheme in detail.

3.1 The 2-D characteristic of the private data

We formulate the 2-dimensional characteristic of the attribute and time for the privacy data collected by WSNs now. Many kinds of data are collected continuously by a WSN. Data with different attributes may have different usage. Meanwhile, data with the same attribute but collected in different time should be treated differently. Thus, we argue that the data collected by WSNs have the 2-D characteristic in terms of attribute and time.

Attribute**:* Data collected by a WSN can be regarded as a data set $D=(d_1,d_2,\cdots,d_x)$, where each element indicates one kind of data. Partition the data set D to obtain a set partition $P=\{D_1,D_2,...,D_m\}$, then rearrange all elements in P by a given rule, we can obtain a sequence $A=\{a_1,a_2,...,a_m\}$. Each element of sequence A is named as an ***attribute. The partition and rearrange rule here can be made according to the specific function of the WSN.

Time**:* Data collected in WSNs can be differentiated by the time that it is collected. Given a fixed time slot T_Δ, the time from the start of a WSN system to a given moment can be partitioned into a time sequence $T=\{t_1,t_2,...,t_n\}$, which is in chronological order. Each element of sequence T is named as ***time slot.

If we deem all data collected by a WSN in a 2-D plane which is synchronized by attribute and time, the plane is divided into many subspaces in terms of attribute and time. If we use s_{ij} to indicate the ith attribute of data collected at *time slot j,* the data collected by a sensor node can be indicated using a matrix S.

$$S=\begin{pmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mn} \end{pmatrix}$$

3.2 Overview of TPP

We present the working model of TPP firstly. We only consider the case that all the collected personal data are transmitted and stored on the sink in a people-centric WSN. In order to provide adequate security, the data collected by a sensor node are divided into many subspaces by the 2-D characteristic of attribute and time and are encrypted with different sub-keys before they are transmitted to the sink.

When time elapses, the number of sub-keys will rapidly increase. It is infeasible to store all these encryption keys in a single sensor node due to very limited storage space. Thus, we need design a lightweight Key Generation Scheme (KGS) to generate a sub-key for the data in each subspace. KGS works synchronously with data collection on sensor nodes. When the privacy data have been collected in a *time slot*, corresponding sub-keys for different attributes are generated at the same time. The sensor node encrypts the data using the sub-keys and then transmits them to the sink. The sub-keys are discarded when new sub-keys are generated in the next *time slot*.

Because all the data are encrypted, nobody can read the data stored on the sink. If someone is plan to read any part of the data, he has to send an attribute-time pair *(a, t)*, which is encrypted using the traditional secure transmission technologies, such as the one described in literature [7], to the corresponding sensor node first. On the sensor node there is a mechanism which can be designed to satisfy different requirements to decide whether the corresponding sub-key is returned. If so, then the sensor node generates the corresponding sub-key and sends it to the sink. Only when the one has got the sub-key, he could read the needed data. We will present the Data Abstraction Scheme (DAS) and 3 different Key Update Modes in Subsection 3.4 and 3.5 in detail.

3.3 Key Generation Scheme (KGS)

The goal of KGS is to generate a distinct encryption key for the data in each subspace. That is, to generate a sub-key matrix *K,* in which k_{ij} indicates the sub-key for the data in subspace s_{ij}, for the data matrix *S*.

$$K = \begin{pmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mn} \end{pmatrix}$$

Sensor nodes have extremely limited power resources and computational capabilities, so traditional cryptographic algorithms are considered too bulky, complex and power hungry for them. So we select the lightweight hash function for KGS, which has been proved to adapt to wireless sensor networks [17, 18, 19]. We utilize the traditional one-way hash chain technology and propose a key generation scheme which is illustrated in **Figure 1**.

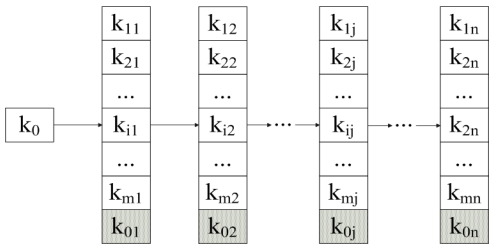

Figure 1: Key generation scheme

First of all, we should introduce the symbols used in the algorithm. Assume that our selected hash function is $H = Hash(k)$. Each sensor node stores one single seed key named k_0 which is used as the initial input of the hash chain. The length of each sub-key is $|k_{ij}| = l$, and the length of reserved key is *L*, so the length of each hash value is $|H_i| = l \times m + L$.

The KGS works as follows. At the start moment of each term t_i, the hash value H_{i-1} is used as the input to generate the corresponding hash value $H_i = Hash(H_{i-1})$. Then a hash chain $H = \{H_1, H_2, ..., H_n\}$ is generated.

For each hash value H_i, we divided it into $m+1$ parts named as $k_{0i}, k_{1i}, ..., k_{mi}$ respectively in which $|k_{0i}| = L$ and $|k_{1i}| = |k_{2i}| = \cdots = |k_{mi}| = l$. $k_{1i}, k_{2i}, ..., k_{mi}$ are used as the needed sub-keys. And the remained part k_{i0} is named as ***the reserved key*** which will not be used in any time and is also never sent to the sink.

Algorithm 1 Key Generation Algorithm

Input: the seed key k_0, attributes number *m*, sub-key length *l,* reserved key length *L*
Output: sub-key matrix *K*
Algorithm:

Step1: Compute the hash value $H_1 = Hash(k_0)$, the length of which is $|H_1| = l \times m + L$.

Step2: Divide H_1 into $m+1$ parts named as $\{k_{01}, k_{11}, ..., k_{m1}\}$, in which $|k_{01}| = L$ and $|k_{11}| = |k_{21}| = \cdots = |k_{m1}| = l$; select $k_{11}, k_{21}, ..., k_{m1}$ as the 1st column of the sub-key

matrix;

Step3: For each $i \in \{i \mid 1 < i \le n, i \in N\}$, use H_i as the input of hash function, compute the hash value $H_{i+1} = Hash(H_i)$, and $|H_{i+1}| = l \times m + L$;

Step4: Divide H_i into m+1 parts named as $\{k_{0i}, k_{1i}, ..., k_{mi}\}$, in which $|k_{0i}| = L$ and $|k_{1i}| = |k_{2i}| = \cdots = |k_{mi}|$, select $k_{1i}, k_{2i}, ..., k_{mi}$ as the ***i***th column of the sub-key matrix;

Step5: Return the sub-key matrix $K = \begin{pmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mn} \end{pmatrix}$

- **Discussion**

It is well known that traditional one-way hash chain has the following properties: (i) If, given a hash value H_i, it is computationally infeasible to find the hash value H_{i-1}. (ii) To compute the hash value H_{i+1}, one has to know the value of H_i. In KGS, *the reserved key* is not be used in any time and is also never sent to the sink. So even someone has collected all the sub-keys $k_{1i}, k_{2i}, ..., k_{mi}$, it is impossible to compute the hash value H_i. So it is also impossible to infer H_{i+1}. So it ensures the security of all sub-keys. That is, given a group of sub-keys, it is computationally infeasible to find any other sub-keys.

The security of our scheme only depends on the security level of our selected hash chain and the length of *the reserved key*. Actually, there is a trade off between the security level and the energy consumption. But the length of the reserved key could be set to adapt to the different security requirements. Actually, there is a more complex hierarchical one-way chain which might be used in our scheme. But compare with our revised hash chain, it is more complex and consumes more energy.

There are several reasons for KGS could be applied to the resource limited WSNs. First of all, hash function has been proved that it is adapt to WSNs. And multi-hashing in the hash chain also increases the security level without increasing its footprint. Secondly, because the KGS works synchronously with the data collection and the sub-keys are generated only at the beginning of each *time slot*, the computation time for generating the sub-keys is also not a critical problem. Further more, the sub-key is discarded when new sub-keys are generated in the next term. So the storage space consumption is extremely low.

3.4 Data Abstraction Scheme (DAS)

The core of DAS is that: When a sensor node receives a request, an attribute-time pair *(a, t)*, how it generates the corresponding sub-key k_{ij} and returns it to the sink.

We use the same symbols as described in Subsection *3.3* in the algorithm. In addition, we assume that the starting time of TPP works is T_1 and the length of each *time slot* is T_Δ.

DAS works as follows. Given a request moment ***t,*** it is easy to compute which *time slot* it is in. Assume it is j, then $j = \left\lceil \frac{(t - T_1)}{T_\Delta} \right\rceil$. So hash value in *time slot* j is $H_j = Hash^j(k_0)$, which means that k_0 is used as the seek key and is hashed j times. The length of H_j is $|H_j| = l \times m + L$.

For a given attribute ***a***, the corresponding sub-key in term j is $k_{ij} = Subkey(H_j, l \times i - l, l)$, which means that retrieve a substring from H_j. And the substring starts at the $(l \times i - l)\,th$ character and its length is l.

Algorithm 2 Data Abstraction Algorithm

Input: an attribute-time pair *(a, t)*

Output: the corresponding sub-key k_{ij}

Algorithm:

Step1: Compute the coordinate (i, j) of the subspace which the request data is in by the following expressions. i = a ; $j = \left\lceil \frac{(t - T_1)}{T_\Delta} \right\rceil$

Step2: Compute the hash value $H_j = Hash^j(k_0)$ and $|H_j| = l \times m + L$.

Step3: Compute the requested sub-key is $k_{ij} = Subkey(H_j, l \times i - l, l)$

Step4: Return k_{ij} to the sink.

- **Improvement of the scheme**

From the expressions $H_j = Hash^j(k_0)$ and $j = \left\lceil \frac{(t - T_1)}{T_\Delta} \right\rceil$, it is easy to see that the sensor node's computation consumption for responding each request depends on the number of update terms from the

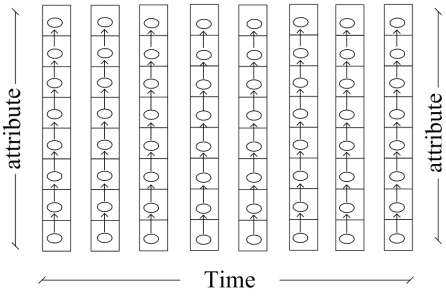

Figure 2: Transverse update mode (TUM)

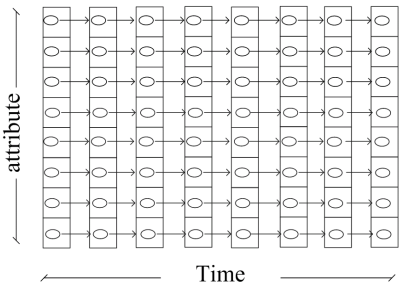

Figure 3: Longitudinal update mode (LUM)

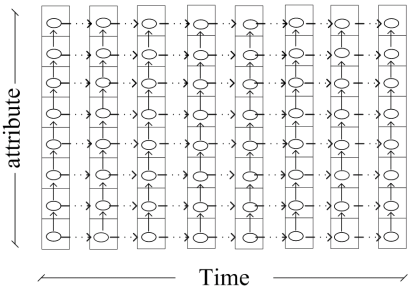

Figure 4: Massive update mode (MUM)

starting time to the time in the attribute-time pair. So if j is very huge, the total amount of computation for responding each request might be also too huge for the energy limited sensor node.

To avoid always using the seed key k_0 as the input to compute the needed sub-key for each request, we make a tradeoff between the computation overhead and the memory space usage. We store a hash value H_{yN} on the sensor node in every N *time slots*, which can be set to adapt to different applications. And all of the hash values can be indicated by H_{yN}.

Then we describe how the sensor node generates the corresponding sub-key for the request from the sink again. When a sensor node received an attribute-time pair *(a, t)*, it can easily compute $j = \left\lceil \frac{(t-T_1)}{T_\Delta} \right\rceil$, and $yN < j < (y+1)N$, so the hash value H_j can be computed using the expression: $H_j = Hash^{j-yN}(H_{yN})$.

Our simulation results in Section 4 show that the computation consumption is greatly decreased.

3.5 Update Modes

In DAS, we can see that each request sent by the sink can only access to one sub-key. Actually, someone might want to access to more sub-keys in one request. For example, in some applications, some people often read all the data of with the same attribute. But in other applications, people might always want to access to the entire data collected at the same term. So the sensor node needs have different key update modes to satisfy different applications. So we design three update modes for our privacy protection scheme. Each update mode is used in different situation. The main purpose is to decrease the consumption of computation. Due to the page limitation, we only introduce the main idea of each mode.

3.5.1 Transverse update mode (TUM). TUM is the mode we have used in the previous description. The data collected in the same *time slot* use the sub-keys generated from the same hash value; and the hash value is updated for each *time slot*. Using this mode, the sink can request all the required sub-keys for all kinds of data collection in the same *time slot* in one request. **Figure 2** shows the Transverse Update Mode. In the figure only the first eight update cycles are marked.

3.5.2 Longitudinal update mode (LUM). As shown in **Figure 3**, in LUM, there is a corresponding seed key k_i for each attribute of data. And they generate sub-keys using the same scheme we design in Subsection 3.3. The difference is that the data with the same attribute use the sub-keys generated by their corresponding seed key. It is updated in every **m** *time slots*. Using this mode, the sink can request multiple sub-keys with the same attribute in continuous *time slots* within one request packet.

3.5.3 Massive update mode (MUM). In MUM there are two seed keys on each sensor node. And they generate sub-keys using the scheme we design in Subsection 3.3. In an arbitrary *time slot n*, they generate a group of sub-keys respectively, named as $ka = \{ka_0, ka_1, ka_2, \cdots, ka_m\}$ and $kb = \{kb_0, kb_1, kb_2, \cdots, kb_m\}$, in which m is the total number of attributes. ka_0 and kb_0 are both ***reserved keys***. We select ka_i, the ith sub-key of *ka,* and kb_j, the jth sub-key of *kb*, where $i, j \neq 0$, and then compute $k_{ij} = ka_i \oplus kb_j$ which is the corresponding sub-key for the data in subspace *(i, j)*. So we can obtain a sub-key matrix K_n. The data in matrix S_n, which is a sub matrix of data matrix S, use the corresponding sub-keys in K_n. Using this mode, the sink can send many attribute-time pairs in one

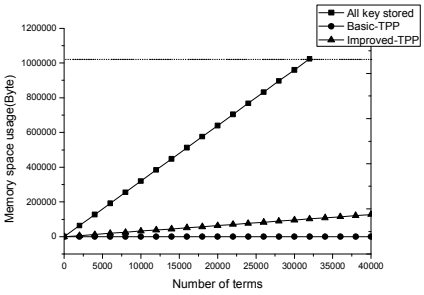

Figure 5: Memory space usage

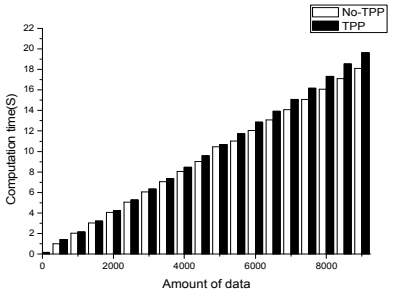

Figure 6: Computation overhead

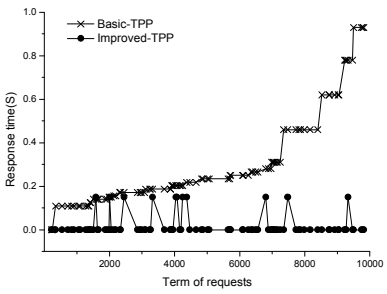

Figure 7: Computaion time for single sub-key requests

request to request the corresponding sub-keys of the data in a matrix, i.e., S_n. **Figure 4** shows this mode, where only the first update cycle is marked.

4. Performance evaluation

In this section, we evaluate the performance of our proposed scheme. Our proposed TPP scheme can efficiently generate 2-dimensional keys with guaranteed security. A side effect of the scheme is that we can carry out the access control in the sink as well for queries from ordinary sensor nodes. To the best of our knowledge, there are no similar researches for the 2-dimension keys generation to provide privacy in WSNs so far. Hence we don't provide comparison work with others. However, we perform extensive simulations in various scenarios to evaluate the performance of our proposed scheme with respect to different application requirements.

4.1 Performance Metrics

We define the following performance metrics:

Memory space usage: It includes the memory space consumption for storing the seed key and the generated sub-keys on a single sensor node.

Computation overhead: It includes two parts: (i) the overhead for generating all the sub-keys for encrypting the collected data before they are transmitted to the sink and (ii) the overhead for generating the corresponding sub-keys requested by the sink.

Response time of a sub-key request is one of the parameters concerned by users. To highlight the computation time, we ignore the transmission time. We define the *response time* of a request as a time slot from when a sensor receives a request to when it generates the corresponding sub-key.

4.2 Simulation Setup

We evaluate the performance of TPP using a custom simulator. In the simulation setup, we randomly deploy 1000 sensor nodes into a (600 m $\times$ 600 m) square sensing field, and the average communication range is set to 40 m. The memory space for each sensor node is 1MB. All sensor nodes collect 10 attributes of data continuously, encrypt and send them to the sink by a Shortest Path Tree. Also the sink can send requests to every sensor node.

4.3 Experiment results

4.3.1. The memory space usage. To evaluate the *memory space usage*, we generate different network topologies with 1000 nodes. On each topology, we implement three models of our TPP scheme. Apart from the basic time-based privacy protection model (Basic-TPP) and the improved time-based privacy protection model (Improved-TPP), which are designed in Section 3, we also implement the all keys stored model (All keys stored). In this model, all the generated sub-keys are stored on the sensor nodes. All the sensor nodes collect data which are partitioned into 10 attributes continuously, and then encrypt them and transmit to the sink. We run the network more than 50000 terms. The length of each sub-key is 16bit.

As shown in **Figure 5**, the memory space usage is extremely low in the basic model, because only one single seed key and the using sub-keys are required to be stored. For the all keys stored model, the memory space consumption increases rapidly, and at the 32768th term the sensor node broken down. The reason is that there is no more memory space to store the sub-keys. That is, it is infeasible to store all sub-keys on a sensor node. For the improved TPP scheme, the memory space consumption increase with time elapses, but its increase rate is much lower. It can satisfy normal operations of a WSN system.

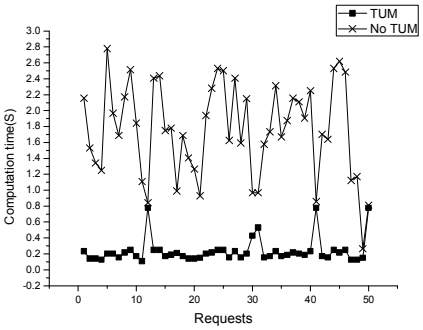

Figure 8: Average response time of TUM.

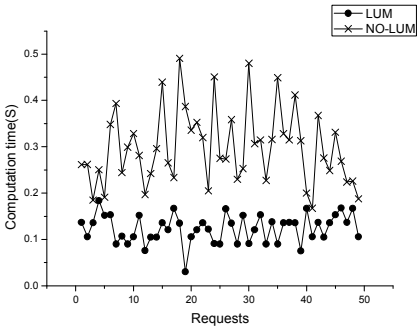

Figure 9: Average response time of LUM

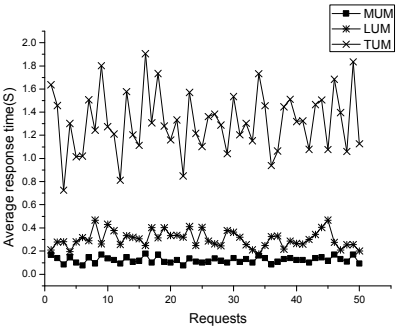

Figure 10: Average response time of MUM

4.3.2. The computation overhead. To evaluate the *computation overhead* (i) described in Subsection 4.1, we generate different network topologies with 1000 sensor nodes. On each topology, we implement the ordinary data transmission model, in which there is no privacy protection scheme (No-TPP), and the model with the TPP scheme (TPP), respectively. In both models, to ensure the security, all data require to be encrypted. The only difference is that all data are encrypted using the same key in No-TPP model, while different sub-keys are used in TPP model. As shown in **Figure 6**, there is about 5% overhead increases.

The *computation overhead* (ii) is entirely introduced by our TPP scheme. So we have to decrease it as much as possible. We firstly run the networks with the Basic-TPP and Improved-TPP 10000 *terms*, respectively. And then the sink randomly sends 20 attribute-time pairs to the sensor node. The response time to compute the corresponding sub-keys in both models are shown in **Figure 7**. We can see that the response time in Improved-TPP is very low.

4.3.3 The average response time in different update modes. To evaluate the performances of our proposed three Key Update Modes, we design 3 groups of experiments. We firstly generate three networks using the same topology, and they runs TUM, LUM and MUM respectively. In each group of experiment, we run the networks 10000 terms firstly. Then the sink sends 500 attribute-time pairs to sensor nodes using different models. We run this ten times compute the average response time. In the first group, the sink requests 10 continuous sub-keys in a same *time slot* using TUM, in which the sink sends 10 attribute-time pairs in a request packet, and No-TUM model respectively. As shown in **Figure 8**, the average response time of TUM is only about 10% of the NO-TUM's. In the second group, the sink requests 10 sub-keys with the same attribute in 10 continuous *time slots*. **Figure 9** suggests that LUM, in which the sink sends 10 attribute-time pairs in a request packet, decreases the response time significantly. Here, No-TUM and No-LUM mean that the sink sends only one contribute-time pair in each request packet. In the last group, the sink requests 100 sub-keys in a 10×10 sub-keys matrix. Compare with TUM and LUM, we can see that the performance of MUM, in which the sink sends these 100 attribute-time pairs in a request packet, is best in this case, which is shown in **Figure 10**.

5. Related works

Many efforts have been made to mitigate the risks of privacy exposure in WSNs. The privacy protection solutions that have been proposed may be categorized into two groups: content-oriented solutions and contextual privacy protection solutions [13].

Content-oriented privacy threats are issues that arise due to the ability of the adversary to observe and manipulate the exact content of packets. Current research is mainly concentrated on the transmission security and the defense of multifarious attacks. And many solutions have been proposed for these threats. One approach is to create secure platforms in which provide link layer cryptographic primitives or libraries. TinySec [6], SecureSense [8] and SenSec [10] are the examples. There are many other solutions, including secure information routing protocols such as SPINS [7] and LEAP [9], security aware middleware services such as secure localization [11] and secure time synchronization [12].

The issue of contextual privacy is concerned with protecting the context associated with the measurement and transmission of sensed data, such as, the location of the source node(s) that observed the target, the time when the source node(s) observed the target. Source location privacy in WSNs is extensively studied in [13, 14], where phantom routing, which uses a random

walk before commencing with regular flooding/single-path routing, protects the source location. In [15], Deng proposed randomized routing algorithms and fake message injection to prevent an adversary from locating the network sink based on the observed traffic patterns. Temporal privacy in WSN is formulated in literature [16], and an adaptive buffering algorithm, RCAD, is proposed.

6. Conclusions

In this paper, we addressed the privacy problem to multi-attribute data in a 2-dementional plane in people-centric WSNs. The privacy data are divided into many subspaces linking to their attribute and generation time. In order to provide controllable security, we proposed a TPP scheme for multi-attribute data in WSNs based on a revised hash chain technology. The data in each subspace is encrypted with different sub-keys before the transmission to the sink. If someone wants to read any part of these data, he has to send an attribute-time pair *(a, t)* to the sensor node firstly. On the sensor node, there is a mechanism to determine whether to respond the request. If so, then the sensor node generates the corresponding sub-key and sends it to the sink. Only when the one has got the sub-key, he could decrypt and read the needed data. TPP scheme could generate these sub-keys using limited calculating consumption. The storage space consumption is also extremely low; even only one single seed key is stored on each sensor node. We do some improvements to adapt to different application scenarios. The simulation results show that it applies to the resource limited WSNs.

Future works on TPP will lead in two directions. One is to improve the security of the sub-keys by using other key generation algorithms. The other one is the design of a comprehensive mechanism that the owners of the data can manage their privacy data freely.

Acknowledgments

This work is supported in part by the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, the National High Technology Research and Development Program of China (863 Program) under grant 2007AA01Z180, NSFC Key Project grant No. 60736016, and NSFC grants No. 60573045 and 60873198.

References

[1] S. Kumar, T. H. Lai, and A. Arora, "Barrier Coverage with Wireless Sensors," in Proceedings of MobiCom, 2005.

[2] M. Li and Y. Liu, "Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes," in Proceedings of ACM MobiCom, 2007.

[3] M. Li and Y. Liu, "Underground Structure Monitoring with Wireless Sensor Networks," IEEE IPSN, 2007.

[4] R. Szewczyk, A. M. Mainwaring, J. Polastre, J. Anderson, and D. E. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," ACM SenSys, 2004.

[5] Z. Yang, M. Li, and Y. Liu, "Sea Depth Measurement with Restricted Floating Sensors," IEEE RTSS, 2007.

[6] C. Karlof, N.Sastry, D. Wagner, "TinySec: Link Layer Encryption for Tiny Devices," ACM SenSys, 2004

[7] A. Perrig, R. Szewczyk, V. Wen, D. culler, "SPINS: Security Protocols for Sensor Networks," ACM CCS, 2003

[8] Q. Xue, A. Ganz, "Runtime Security Composition for Sensor Networks (SecureSense)," IEEE Vehicular Technology Conference, 2003

[9] S. J. S. Zhu, S. Setia, "LEAP: Efficient security mechanism for large-scale distributed sensor networks, " ACM CCS, 2003

[10] T. Li, H. Wu, F. Bao, "SenSec Design," Institute for InfoComm Research, Tech. Rep. TR-I2R-v1.1, 2005

[11] S.Capkun, J.P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," IEEE Infocom, 2005

[12] S. Ganeriwal, S. Capkun, C. C. Han, M. B. Srivastava, "Secure time synchronization service for sensor networks," ACM WiSe, 2005

[13] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," IEEE ICDCS '05, 2005.

[14] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," ACM SASN, 2004.

[15] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in 1st IEEE/CreateNet Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005.

[16] P. Kamat, W. Xu, W. Trappe, Y. Zhang, "Temporal Privacy in Wireless Sensor Networks," IEEE ICDCS, 2007.

[17] M. Brown, D. Cheung, D. Hankerson, J. Hernandez, M. Kirkup, and A. Menezes, "PGP in constrained wireless devices," in Proceedings of the 9th USENIX Security Symposium, pages 247-261. USENIX, August 2000.

[18] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in Network and Distributed System Security Symposium, NDSS '03, pages 263-276, February 2003.

[19] D. Coppersmith and M. Jakobsson, "Almost optimal hash sequence traversal," in Proceedings of the Fourth Conference on Financial Cryptography (FC'02), Lecture Notes in Computer Science, 2002.