PAPG: Private Aggregation Scheme based on Privacy-preserving Gene in Wireless Sensor Networks

Weini Zeng[1], Peng Chen[1], Hairong Chen[1] and Shiming He[2]
[1]The 716TH Research Institute, China Shipbuilding Industry Corporation
Lianyungang, Jiangsu, 222003, P.R. China
[e-mail: zengweini@jari.cn]
[2]School of Computer and Communication Engineering, Changsha University of Science and Technology
Changsha, Hunan, 410114, P.R. China
[e-mail: hsm@163.com]
*Corresponding author: Weini Zeng

Received January 3, 2016; revised June 29, 2016; accepted August 3, 2016; published September 30, 2016

Abstract

This paper proposes a privacy-preserving aggregation scheme based on the designed P-Gene (PAPG) for sensor networks. The P-Gene is constructed using the designed erasable data-hiding technique. In this P-Gene, each sensory data item may be hidden by the collecting sensor node, thereby protecting the privacy of this data item. Thereafter, the hidden data can be directly reported to the cluster head that aggregates the data. The aggregation result can then be recovered from the hidden data in the cluster head. The designed P-Genes can protect the privacy of each data item without additional data exchange or encryption. Given the flexible generation of the P-Genes, the proposed PAPG scheme adapts to dynamically changing reporting nodes. Apart from its favorable resistance to data loss, the extensive analyses and simulations demonstrate how the PAPG scheme efficiently preserves privacy while consuming less communication and computational overheads.

Keywords: Wireless Sensor Networks, Privacy-preserving Gene, Data Aggregation, Privacy, Distributed System

A preliminary version of this paper was presented at IEEE FiCloud 2016 , August 22-24, 2016, Vienna, Austria. This version includes a concrete analysis, formal security proof of the proposed scheme, and supporting implementation of simulations. This research was supported by the National Natural Science Foundation of China under Grant No. 61303045, by the Open Research Fund of Hunan Provincial Key Laboratory of Network Investigational Technology under Grant NO.2016WLZC016. We express our thanks to Prof. Xianghui Cao who checked our manuscript.

http://dx.doi.org/10.3837/tiis.2016.09.023 **ISSN : 1976-7277**

1. Introduction

A wireless sensor network is a collection of low-cost, small-size sensor nodes for sensing certain conditions or events. Given that sensor networks can sense the physical world and transmit their sensory data without any infrastructure, they have broad application prospects, such as in monitoring hospitals and critical facilities [1]. However, sensor nodes have a limited energy supply; therefore, in-network data aggregation by which sensors collaborate on in-network processing to reduce the amount of raw data has been widely adopted to prolong the system lifetime [2].

When sensor networks monitor the surrounding environment, the data produced by each sensor node must be known only to itself to ensure data privacy. Without proper privacy protection, sensor networks have impractical applications in civilian areas because the participating parties may disallow the tracking of their private data. Therefore, the privacy of each sensory data must be preserved during the data aggregation process [3].

However, data aggregation and data privacy protection contradict each other. To achieve data aggregation, any aggregator must view each data item they process in plaintext to prevent such items from being encrypted at all times, which can lead to data privacy violation. Therefore, end-to-end data encryption, a well-known security method, is inapplicable in this context. A new method that specifically addresses such contradiction must be devised. The existing approaches suffer either from low resistance to nodes, high communication overhead, high computational overhead or low adaptability to unreliable channels [3–17]. (Section 2 presents an in-depth discussion of related studies.) Therefore, in this paper, we revisit the aforementioned problem and then propose a scheme based on a newly designed data-hiding technique.

Data-hiding techniques can ensure data privacy and have been applied for aggregating private data [4, 7, 10]. When a sensor node has sensory data to report, the data with one or more secret items are hidden, and then the sensor node sends the hidden data instead of the original one. In this way, the other nodes/outsiders cannot obtain any private sensory data. However, obtaining the aggregation result remains an issue. To address this problem, a straightforward approach is to enable the secret data to be shared between the node and the base station (BS), and then allow the BS to recover the aggregation result. This method has been adopted in several existing works, such as the Fully-reporting Secret Perturbation-based Scheme (FSP)[10]. This method allows the BS to obtain the private data, but the aggregation result cannot be recovered within the network. Data loss may also affect the recovery of the aggregation result, and the BS cannot access the source information of the lost data.

To address these problems, we propose a new distributed private aggregation scheme based on the constructed privacy-preserving gene (PAPG). Similar to existing solutions, PAPG adopts data hiding to achieve private data aggregation. However, our erasable data-hiding technique is more suitable than the existing methods. The main contributions of this study are outlined as follows:

(1) This study constructs the P-Gene, a new data-hiding carrier, and then proposes the erasable data hiding technique. Unlike other data hiding carriers, in our P-Gene, each node can hide its sensory data independently using some simple calculation operations. Then, the hidden data is sent to the intra-network aggregator without encrypting. The aggregator that implements the aggregating operation can also erase all the P-Genes without knowing them. In

this way, each private datum is protected, and the in-network aggregator can obtain the aggregation values of the original data without additional data exchange.

(2) This paper proposes a method for secret P-Gene generation independent of cryptographic algorithms. In this method, each node independently and dynamically generates its P-Gene according to the dynamic reporting cluster members via some simple calculation operations. That is, during the generation of P-Genes, no extra message exchange is introduced even when some or all the network nodes are reporting. This method also demonstrates efficient communication and computation because of its simple and lightweight calculation operations.

(3) Compared with existing distributed approaches, the proposed PAPG scheme can efficiently preserve data privacy with low power consumption. Compared with existing centralized approaches, the proposed PAPG scheme can efficiently adapt to unreliable channels and avoid the single point problem. Specifically, compared with our previous work [7], the newly proposed method can work with other secure data aggregation schemes to ensure data integrity because each node can send its hidden data to the cluster head (CH) without encryption. The PAPG scheme also has a highly efficient computation because this method does not depend on cryptographic algorithms after the initialization process following the deployment of the network.

(4) Extensive analyses and simulations reveal that the proposed scheme outperforms the existing private data aggregation schemes [4,7,10] in terms of private data protection and power consumption. The novel erasable data-hiding technique and hiding-data generation method are also useful for ensuring the data privacy of other distributed systems.

The rest of this paper is organized as follows. Section 2 briefly reviews the related literature. Section 3 describes the system models, design goals, and basic idea of the erasable data-hiding technique. Section 4 defines the newly constructed P-Gene, its properties, and generation method. Section 5 elaborates the proposed PAPG scheme, while Sections 6 and 7 analyze its privacy-preserving and data aggregation efficacy, respectively. Section 8 evaluates the performance of the scheme. Section 9 concludes the paper.

2. Related Work

Many studies have been conducted on the privacy-preserving problem. According to whether the aggregation result could be retrieved in-network, all the private data aggregation schemes could be divided into two categories, namely, distributed and centralized, and they are analyzed in parts 2.1 and 2.2, respectively.

2.1 Distributed private data aggregation scheme

In the distributed private data aggregation scheme, nodes collectively retrieve the aggregation result in-network. Thus, this scheme can avoid the limitations of the centralized scheme such as data loss problem. However, this scheme still has weaknesses.

The pioneering distributed private data aggregation schemes are the cluster-based private data aggregation (CPDA) scheme and the slice–mix–aggregate (SMART) scheme proposed by He *et al*. [4]. In [4], a data-hiding technique is designed based on the algebraic properties of polynomials, and then a CPDA scheme of three rounds of intra-cluster nodes interactions is proposed. In SMART, each node slices its sensory data into J pieces, and then distributes $(J-1)$ of these pieces to its nearest $(J-1)$ nodes for aggregation. However, the privacy efficacy of both CPDA and SMART is restricted by the communication overhead. For a certain node, if all its cluster members in CPDA or all its interacted neighbor nodes in SMART are

compromised, its private data will be disclosed. Although the privacy efficacy can be raised by expanding the cluster size or increasing the number of slices, doing so rapidly increases the communication overhead.

Considering the message loss problem, Conti *et al*. proposed a robust privacy-preservation data aggregation scheme [5]. In this scheme, each pair of nodes establishes a twin key, and each node encrypts its sensory data by adding shadow values computed from the live twin keys it holds. Then, as the contribution of the shadow values for each twin key cancels each other out, the aggregation result is retrieved. Although this scheme can solve the message loss problem, the communication overhead remains expensive because each node within a cluster with size *n* has to send/receive *n* messages. Like Conti *et al*., Huang *et al*. proposed a scheme based on XOR and hash operation [6]. However, this scheme can only adapt to the scenario of fixed reporting nodes, while in reality, the reporting nodes can be changed dynamically.

Zeng *et al*. proposed a scheme based on the designed P-function set [7], where cluster members are divided into several P-classes, and each node generates its P-function according to the P-class membership. Each node hides its sensory data through the P-function value and sends the result to the cluster head in which the intra-cluster aggregation result is retrieved. Unfortunately, this scheme cannot work with the secure data aggregation schemes providing data integrity. Besides, each node needs to encrypt the data sent to the cluster head, thereby generating high computation overhead.

Jung *et al*. proposed a privacy-preserving data aggregation scheme based on the hop-by-hop and multi-polynomial encryptions [8]. This scheme can be used for additive and multiplication aggregation functions; but it suffers from high power consumption because each node needs to send several extra messages.

2.2 Centralized private data aggregation scheme

In the centralized private data aggregation schemes, the aggregation result can be retrieved only by the BS. Most of the centralized schemes adopt the idea of additively homomorphic encryption first proposed by Castelluccia *et al.* [9]. This idea is expressed as follows: each node *b* shares a secret data *k* with the BS and uses *k* to hide its sensory data *d* as $(d+k) \bmod M$ (*M* is a system parameter), which is further aggregated along the way to the BS. When receiving all the data, the BS retrieves the aggregation result by subtracting all the secret data *k*. Schemes that implement this basic idea have the following limitations: (1) if only part of the nodes report, the IDs of the reporting nodes need to be reported; (2) when all the nodes report, although the node ID does not need to be reported in theory, if any message loss occurs, then the aggregation result cannot be recovered; (3) if the attacker obtains *k* and the range of the hidden data, then obtaining the range of the private sensory data is also possible [10]. Thus, Castelluccia *et al.* proposed an enhanced scheme where the parameter *k* is generated dynamically [11]. However, this scheme still suffers from problem (2) [10].

Feng *et al.* also proposed a series of optimized schemes [10]. In the proposed FSP scheme, all the nodes are required to report their sensory data. However, plenty of extra communication overhead is introduced if fewer nodes exist to be reported. Thus, Feng *et al*. proposed the D-ASP scheme, where for a certain number of clusters, only partial cluster members report their data. For D-ASP, the communication overhead becomes unreasonable if plenty of nodes are reporting their IDs. Again, all these schemes suffer the message loss problem.

Several works are based on the idea of privacy homomorphism (PH), an encryption transformation technique that enables direct computation of encrypted data [12]. Using the symmetric PH technique, Girao *et al.* [13] proposed a concealed data aggregation (CDA) scheme in which all nodes participating in the aggregation share a secret datum and each node

encrypts its own sensory data. The aggregator then determines the sum of the sensory data without decrypting each received datum. However, the private data of one node can be accessed by its neighbors. Zhou *et al*.[14] proposed a secure data aggregation method based on ECC encryption and divide-and-conquer method. Yang *et al*. [15] proposed privacy-preserving data aggregation scheme that employs the EC-EG homomorphism encryption algorithm to provide end-to-end data privacy. However, the computation overhead of these asymmetric cryptography-based schemes is heavy, especially compared with non-encryption based schemes.

In the scheme proposed by Zhang *et al*. [16], the sensor nodes transmit a sample of the data complement to the BS. The BS then reconstructs a histogram of the original sensory data according to the negative samples. However, this scheme cannot provide accurate aggregation results.

Groat *et al*. [17] proposed a *k*-indistinguishable scheme that obfuscates data by adding a set of camouflage values. However, the private data of all the nodes are disclosed if a threshold number of nodes are compromised; raising the threshold results in a rapid increase of the communication overhead.

Currently, the privacy-preservation problem has attracted growing attention in other domains, such as cyber-physical systems, smart grids, and cloud computing [18-23]. Zhang *et al*. [18] address the problem of the cyber-physical systems domain by using differential privacy. For smart grids, Shi *et al*. [19] proposed a diverse grouping-based aggregation scheme with error detection by using differential privacy technique in grouping-based private stream aggregation. Bao *et al*. [20] proposed a secure data aggregation scheme that can achieve differential privacy and fault tolerance simultaneously. There are also other differential privacy technique-based schemes [21-22]. However, these works do not yield accurate aggregation results. In the cloud computing domain, Fu *et al*. [23] proposed a privacy data query scheme that uses vector space model to support searchable encryption. However, this technique is still unsuitable for the present study.

3. Assumptions and Design Goals

3.1 System Model

We consider static sensor networks composed of low-complexity sensor nodes, such as the Berkeley MICA mote, which has a 4 MHz processor and 4 KB RAM data storage. Therefore, each node has enough space to store bytes of information to protect its private data. The sensor nodes in a network are synchronized [24]. After network deployment, all sensor nodes form clusters [25], which are the minimum units for data aggregation. In a cluster, the CH aggregates the data of its cluster members. The role of CH may rotate among the cluster nodes according to appropriate criteria, such as remaining energy. The data aggregated by the CH are then further aggregated gradually as they are forwarded to the BS.

This study focuses on additive aggregation function because many other aggregation functions, including average, count, variance, standard deviation, and any other moment of the measured data, can be reduced to this function [10]. Similar to other privacy-preserving data aggregation schemes, the PAPG scheme also assumes that each sensory datum is an integer ranging from 0 to an upper bound $d_{\max}$. This assumption is reasonable because even if some sensory data are not integers in their original forms, they can still be transformed into integers.

3.2 Threat Model

The attackers may launch various security attacks to the sensor networks. However, these attacks have no one-for-all solution. Therefore, this study separately investigates such attacks and then proposes attack-specific defense techniques. We assume that the attackers aim to obtain private sensory data, and we address the conflict between in-network data aggregation and data privacy preservation. Similar to [4, 7, 10], we adopt the honest-yet-curious threat model, in which sensor nodes may attempt to break privacy but faithfully follow the protocol specification during data aggregation. This model is considered appropriate because sensors that are deployed by a common authority fulfill a certain task and can be trusted to follow the protocol. To obtain the private sensory data of interest, the attackers may launch the following attacks:

•**Eavesdropping**: The attacker may passively eavesdrop on the message transmissions in the network.

•**Node compromise**: The attacker may capture sensor nodes and read out all of the stored data.

•**Node colluding**: When compromising several sensor nodes, the attacker may combine all the information obtained from the compromised nodes to disclose the private data of the other sensor nodes.

3.3 Security Assumptions

In sensor networks, nodes always establish pairwise keys for confidential peer-to-peer communications. Given the rich literature on key management, this study does not investigate such topic. Three common security assumptions are as follows: ① two nodes can establish a pairwise key based on a key management scheme, ② any compromised node has no effect on the pairwise keys shared by other pairs of valid nodes, and ③ the compromised node is eventually detected by most of its neighbors within a certain period. These assumptions are reasonable because, for example, the pairwise key establishment schemes in [26-27] can be used to achieve these assumptions. The watchdog mechanism or several other collaborative intruder detection and identification schemes [28-30] can be used to detect the compromised nodes.

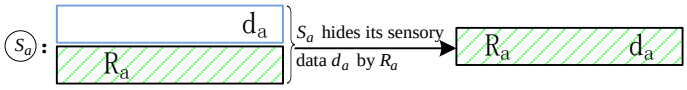
3.4 Design Goals

The new privacy-preserving data aggregation scheme aims to achieve the following qualities:
Data privacy: The sensory data collected by the sensor node must only be known to itself.
Efficiency: The proposed scheme must be as energy efficient as possible because additional overhead will be introduced for privacy protection and the sensor node has constrained resources.
Accuracy: The sensory data must have accurate aggregation results.
Flexibility: The proposed scheme must be adapted to complex network conditions. Sensor networks are naturally prone to data loss because of the vulnerable wireless channel. Nodes may fail or sleep, and new nodes may be added when many invalid nodes are present.

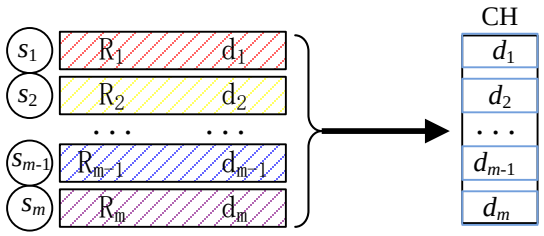
3.5 Basic Idea of Erasable Data hiding Technique

This paper proposes the erasable data hiding technique to achieve the aforementioned design goals. As shown in Fig. 1 (1), in this technique, each cluster member hides its private data using a novel constructed P-Gene (Section 4), and then sends the hidden data to its CH. In this way, data privacy can be ensured during the communication process. As shown in Fig. 1 (2),

after receiving the reporting data of its cluster members without obtaining each P-Gene, the CH can erase all the P-Genes from the hidden data and obtain the aggregation result. This process imitates the irradiation of seven colorful lights (red, yellow, green, blue, pink, brown, and purple) on an object. Specifically, when each of these lights irradiates on an object, the color of the object changes and its real color is hidden. However, when all lights simultaneously irradiate on the object, the color of the object remains the same because mixing these lights produce a white light.



(1) Random a node s_a ($1 \le a \le m$) hides its sensory data d_a by its P-Gene R_a.



(2) At the CH, the P-Genes are erased and the aggregation results are recovered.

Fig. 1. Basic idea of erasable data hiding technique

4. Preliminaries

We use a cluster C_a with size n to elaborate our method, in which each sensor node has a unique intra-cluster ID that is selected from {1, … , n} for intra-cluster communication. Not all nodes may have data to report in each session, which is defined as the interval between two data reporting periods. C_a' denotes the set of reporting nodes with size m ($m \le n$) in C_a. To achieve private data aggregation, a novel P-Gene is constructed for sensory data hiding. To ensure that goals (1) to (3) can be achieved, some concepts and properties related to the P-Gene are specified as follows.

Definition 1 (*P-list* and *P-seed*). *P-list* refers to the list of integers that are generated by node b ($b \in C_a'$) for P-Gene generation. These integers are denoted as *P-list* $\{p_c^b, c \in C_a'\}$, which satisfies

$$(\sum_{c \in C_a'} p_c^b) \bmod U = 0, \quad (1)$$

where $U \ge d_{\max} \times n$, $d_{\max}$ is the upper bound of the sensory data, and n is the maximum cluster size. The length of U is denoted as l bits. Each p_c^b named as ***P-seed*** is only shared between Nodes b and c.

Definition 2 (*P-Gene*). The P-Gene of a random node b ($b \in C_a'$) is denoted as R^b and $R^b = (\sum_{c \in C_a'} p_b^c) \bmod U$ (2).

We obtain Property 1 from the aforementioned definitions. For ease of description, d^b denotes the sensory data of b, while D_b denotes the hidden sensory data $(d_b + R^b) \bmod U$.

Property 1: 1) For a random cluster C_a', $(\sum_{b \in C_a'} R^b) \bmod U = 0$ (3).

2) If $(\sum_{b \in C_a'} d^b) \le U - 1$ (4), then $(\sum_{b \in C_a'} (d^b + R^b)) \bmod U = \sum_{b \in C_a'} d^b$ (5).

Proof: 1) For each cluster member $b \in C_a'$, we have

$$(\sum_{c\in C'_a} p_c^b) \bmod U = 0. \quad (1)$$

Therefore, $(\sum_{b\in C'_a} R^b) \bmod U$

$= [\sum_{b\in C'_a} (\sum_{c\in C'_a} p_b^c) \bmod U] \bmod U$

$= (\sum_{c\in C'_a} \sum_{b\in C'_a} p_b^c) \bmod U$

$= [\sum_{c\in C'_a} (\sum_{b\in C'_a} p_b^c) \bmod U] \bmod U$

$= (\sum_{c\in C'_a} 0) \bmod U = 0.$

2) As $U \geq d_{\max} \times n$, and $(\sum_{b\in C'_a} d^b) \leq U - 1$;

then $(\sum_{b\in C'_a} (d^b + R^b)) \bmod U$

$= (\sum_{b\in C'_a} d^b + \sum_{b\in C'_a} R^b)) \bmod U$

$= (\sum_{b\in C'_a} d^b) \bmod U + (\sum_{b\in C'_a} R^b) \bmod U$

$= \sum_{b\in C'_a} d^b.$

Property 1 shows that for a random C'_a, the sum of all hidden sensory data D_b $(b \in C'_a)$ is equivalent to that of all original sensory data d_b under the modular addition operation.

Numerical Example 1. We present a simple scenario where U=12626 and C'_a = {1, 2, 3} to illustrate property 1. **Table 1** shows the private data and P-seeds of nodes 1, 2, and 3. Take Node 1 for example. According to the P-seeds in **Table 1**, Node 1 calculates its P-Gene as follows (the P-seeds related to Node 1 are underlined in **Table 1**):

$$R^1 = (\underline{3654}+\underline{2379}+\underline{4717}) \bmod 12626 = 10750.$$

Similarly, Nodes 2 and 3 obtain R^2=11500 and R^3=3002, respectively. Thereafter, $(R^1+R^2+R^3) \bmod 12626 = 0$ satisfies Property 11), while $(D^1+D^2+D^3) \bmod 12626 = (d^1+d^2+d^3)$ satisfies Property 12).

Table 1. P-seeds and sensory data of node b

(d^b: sensory data of node b; R^b: the P-Gene generated by ode b; and D^b: hidden sensory data of Node b)

Node b	***P-seeds***	d^b	R^b	D^b
Node 1	{3654,2319,6653}	110	10750	10860
Node 2	{2379,5114,5133}	69	11500	11569
Node 3	{4717,4067,3842}	178	3002	3180

5. PAPG

This section presents PAPG, a distributive scheme that ensures data privacy in the additive data aggregation process based on the erasable data hiding technique. **Fig. 2** shows the entire PAPG process. In this scheme, each node perturbs its private data via its P-Gene without additional data exchange, and the aggregation result can be recovered from the hidden data in the CH. To implement this method, before deploying the nodes, a random node b is preloaded with ① its ID b, and ② t degree polynomial functions $T(x)$ and $W(x)$, which satisfy the condition that the range of the coefficients are [0, U'), where U' is a prime number with length $(l + L)$ bits. $T(x)$ is used for node b to generate the P-Gene R^b, and W(x) is used to update the seeds.

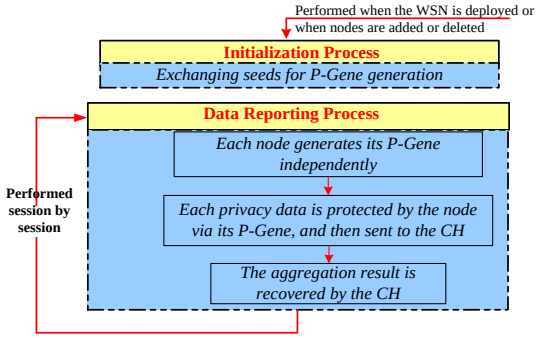


Fig. 2. PAPG process

Sections 5.1 and 5.2 present detailed descriptions of the seed table initialization and maintenance and the data collection process, respectively. **Table 2** presents the basic notations.

Table 2. Basic notations

Notation	*Signification*
C'_a	Set of reporting nodes in cluster C_a
r_c^b	Secret seed for P-seed generation that is generated by node b and only shared between nodes c and b
r_b^c	Secret seed for P-seed generation that is generated by node c and only shared between nodes b and c
p_c^b	Secret P-seed for P-Gene generation that is generated by node b and only shared between nodes b and c
d^b	Original sensory data held by node b
R^b	P-Gene held by node b
D^b	Hidden sensory data held by node b

5.1 Initialization Process for Seed Table Generation and Maintenance

The seed tables are implemented in the following cases:

CASE I: All sensor nodes are clustered after deployment.

In this case, the process is performed only once as follows:

Step 1 (seed exchange): Each node b randomly generates (n-1) data as seeds $\{r_c^b(c \neq b,\ 1 \le c \le n)\}$, where each $r_c^b(c \neq b,\ 1 \le c \le n)$ satisfies $r_c^b < U'$. As shown in **Fig. 3**, each encrypted $r_c^b\ (c \neq b,\ 1 \le c \le n)$ is sent to the corresponding cluster member c through the shared pairwise key $K_{b,c}$: $\{r_c^b\}_{K_{b,c}}$. Similarly, node b receives seed $r_b^c(c \neq b \text{ and } c = 1,2..n)$ from cluster member c.

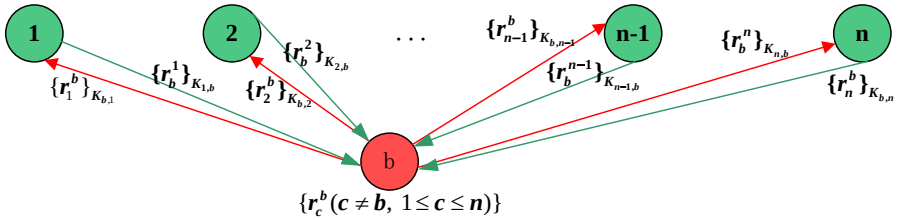


Fig. 3. Exchange of seeds between a certain node b and each of its cluster members ($\{r_c^b(c \neq b,\ 1 \le c \le n)\}$: seeds generated by node b; $r_b^c(c \neq b \text{ and } c = 1,..,n)$: seeds generated by node c; $K_{b,c}$: pairwise key shared between nodes b and c)

Step 2 (formation of seeds table): After the seed exchange, **Table 3** shows that each node *b* initializes its seed table T^b that contains all of its generated seeds $\{r_c^b (c \neq b,\ 1 \leq c \leq n)\}$ and those that are received from the other cluster members $\{r_b^c (c \in C_a, c \neq b)\}$.

Table 3. Seed table T^b of node b
(The seeds in the rectangular box represent those received from the other cluster members.)

c	1	2	…	*n-1*	*n*
r_c^b	r_1^b	r_2^b	…	r_{n-1}^b	r_n^b
r_b^c	$\boxed{r_b^1}$	$\boxed{r_b^2}$	…	$\boxed{r_b^{n-1}}$	$\boxed{r_b^n}$

CASE II: Node failure or compromise

When node *b* is notified that cluster member *c* fails or has been compromised, node *b* deletes r_c^b and r_b^c from T^b.

CASE III: Some nodes have been added.

When node *b* is notified that new cluster members have been added, for each added node *d*, node *b* generates Seed r_d^b, which is then added to T^b and sent to *d*: $\{r_d^b\}_{K_{b,d}}$. Similar to Case I, each added cluster member generates seeds for all other cluster members and then sends these seeds to the corresponding cluster members. Each added cluster member *d* also maintains its seed table T^d for P-Gene generation.

After this process, each pair of valid cluster members (*b*, *c*) only shares the two secret seeds, namely, $\{r_c^b, r_b^c\}$.

5.2 Data-reporting Process

In each data collection session, a random node collects sensory data, hides its data through its P-Gene, and sends the hidden data to its CH. The CH recovers the aggregation result after receiving all reports. For session *s* and C_a' with size *m* ($m \leq n$) (the set of reporting nodes in cluster C_a), only $m \geq 3$ is considered. If $m < 3$, each node slices and sends the data to its neighbors according to the SMART scheme [4], and none of the processes are repeated.

Step 1 (Original sensory data perturbation): According to seeds $\{r_c^b (c \in C_a', c \neq b)\}$, a random node *b* obtains all the P-seeds $\{p_c^b (c \in C_a', c \neq b)\}$, where each p_c^b is the lowest *l* bits of $T(r_c^b)$. Afterward, node *b* calculates the P-seed p_b^b as follows:

$$p_b^b = U - (\sum_{c \in C_a', c \neq b} p_c^b) \bmod U . \quad (6)$$

According to the corresponding seeds $\{r_b^c (c \in C_a', c \neq b)\}$, node *b* obtains all the P-seeds $\{p_b^c (c \in C_a', c \neq b)\}$, which have the lowest *l* bits of $T(r_b^c)$. Thereafter, node *b* calculates its P-Gene R^b according to $\{p_b^c (c \in C_a',\ )\}$ as follows:

$$R^b = (\sum_{c \in C_a'} p_b^c) \bmod U . \quad (7)$$

Node *b* hides its sensory data d^b with R^b as $D^b = (d^b + R^b) \bmod U$ (8), and then sends $\{D^b, b\}$ to its CH.

Step 2 (Data aggregation for each cluster): For each cluster, the CH checks if all nodes have sent their data.

(a) If so, CH calculates $D = (\sum_{b \in C_a'} D_b) \bmod U$ (9), which is equivalent to $\sum_{b \in C_a'} d_b$. CH then sends $\{D, m\}$ to the next hop node.

(b) Otherwise, by assuming that node *c* does not report, CH asks *c* to report. If node *c* responds, CH continues checking and calculating as (a). Otherwise, *c* is considered a failure, and CH sends this information to the cluster members. For each node *b*, go to *Step 1*.

Theorem 1: If all reporting nodes in cluster C_a implement the data-reporting process of PAPG, that is, each node *b* in C'_a implements step 1 mentioned, and the CH implements step 2, then the CH can obtain the aggregation result of C'_a without knowing each P-Gene R^b.

Proof: In Step 1, the $\{p_c^b(c \in C'_a)\}$ that is generated by each node *b* is a *P-list* because the *P-seed* p_b^b is generated according to all other *P-seeds* $\{p_c^b(c \neq b, c \in C'_a)\}$. Therefore, we obtain the following:

$$\begin{aligned}
&(\sum_{c\in C'_a} p_c^b) \bmod U \\
&= (p_b^b + \sum_{c\in C'_a, c\neq b} p_c^b) \bmod U \\
&= [(U - (\sum_{c\in C'_a, c\neq b} p_c^b) \bmod U) \bmod U + (\sum_{c\in C'_a, c\neq b} p_c^b) \bmod U] \bmod U \\
&= [(0 - (\sum_{c\in C'_a, c\neq b} p_c^b) \bmod U + (\sum_{c\in C'_a, c\neq b} p_c^b) \bmod U] \bmod U \\
&= 0
\end{aligned} \tag{10}$$

which satisfies the definition of *P-list*. Then, each R^b generated in Step 1 satisfies the definition of P-Gene. According to **Property 1**, by calculating equation (9), the aggregator can obtain $\sum_{b\in C'_a} d_b$, which is the aggregation result of all the reporting nodes C'_a.

Note: During each data-reporting process, the *P-seed* p_b^b of a random reporting node *b* is dynamically generated according to the other generated *P-seeds* $\{p_c^b(c \in C'_a, c \neq b)\}$ that are generated according to the seeds of the reporting cluster member $r_b^c(c \in C'_a, c \neq b)$. Therefore, (1) the PAPG scheme can adapt to dynamically changing reporting nodes, and (2) to achieve **Theorem 1**, each node *b* has no constraints on the seeds that are generated during the initializing process $\{r_c^b(c \neq b,\ 1 \leq c \leq n)\}$.

Numerical Example 2. We present a simple scenario to illustrate the aforementioned steps. Given *s*=2, *U*=31 (*l*=5), $U' = 1021$, $T(x) = 179x^2 + 839x$, and $C'_a = \{1,2,3\}$. **Table 4** shows the private data held by nodes 1, 2, and 3 and their corresponding seeds.

Table 4. Seeds and sensory data held by cluster members in session 2

Node	Generated seeds	Received seeds	Original sensory data
Node 1	$\{r_2^1 = 12, r_3^1 = 3\}$	$\{r_1^2 = 7, r_1^3 = 23\}$	$d^1 = 6$
Node 2	$\{r_1^2 = 7, r_3^2 = 398\}$	$\{r_2^1 = 12, r_2^3 = 821\}$	$d^2 = 9$
Node 3	$\{r_1^3 = 23, r_2^3 = 821\}$	$\{r_3^1 = 3, r_3^2 = 398\}$	$d^3 = 2$

1) According to $\{\{r_2^1 = 12, r_3^1 = 3\}, \{r_1^2 = 7, r_1^3 = 23\}\}$, node 1 calculates the following:

$$\begin{cases}
T(r_2^1) \bmod U' = (179 \times 12^2 + 839 \times 12) \bmod 1021 = 109 = (11\underline{01101})_2, \\
T(r_3^1) \bmod U' = (179 \times 3^2 + 839 \times 3) \bmod 1021 = 44 = (1\underline{01100})_2, \\
T(r_1^2) \bmod U' = (179 \times 7^2 + 839 \times 7) \bmod 1021 = 350 = (1010\underline{11110})_2, \\
T(r_1^3) \bmod U' = (179 \times 23^2 + 839 \times 23) \bmod 1021 = 657 = (1010010\underline{0001})_2,
\end{cases}$$

and then obtains the following:

$$p_2^1 = (01101)_2 = (13)_{10}, p_3^1 = (01100)_2 = 12, p_1^2 = (11110)_2 = 30, p_1^3 = (10001)_2 = 17.$$

Therefore, $p_1^1 = U - (p_2^1 + p_3^1) \bmod U = 6$,

and $R^1 = (p_1^1 + p_1^2 + p_1^3) \bmod U = 22$.

Node 1 obtains $D^1 = (d^1 + R^1) \bmod U = (6 + 22) \bmod 31 = 28$ and then sends {1, 28} to the CH.

2) According to { $\{r_1^2 = 7, r_3^2 = 398\}$, $\{r_2^1 = 12, r_2^3 = 821\}$ }, node 2 calculates the following:

$$\begin{cases} T(r_1^2) \bmod U' = (179 \times 7^2 + 839 \times 7) \bmod 1021 = 350 = (1010\underline{11110})_2, \\ T(r_3^2) \bmod U' = (179 \times 398^2 + 839 \times 398) \bmod 1021 = 180 = (101\underline{10100})_2, \\ T(r_2^1) \bmod U' = (179 \times 12^2 + 839 \times 12) \bmod 1021 = 109 = (11\underline{01101})_2, \\ T(r_2^3) \bmod U' = (179 \times 756^2 + 839 \times 756) \bmod 1021 = 987 = (11110\underline{11011})_2, \end{cases}$$

and then obtains the following:

$$p_1^2 = (11110)_2 = 30,\ p_3^2 = (10100)_2 = 20,\ p_2^1 = (01101)_2 = 13,\ p_2^3 = (11011)_2 = 27.$$

Therefore, $p_2^2 = U - (p_1^2 + p_3^2) \bmod U = 12$,

and $R^2 = (p_2^1 + p_2^2 + p_2^3) \bmod U = 21$.

Node 2 obtains $D^2 = (d^2 + R^2) \bmod U = (9 + 21) \bmod 31 = 30$ and then sends {2, 30} to the CH.

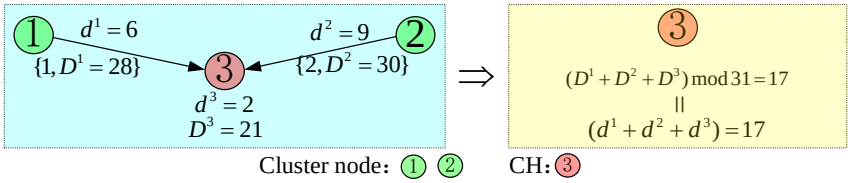


Fig. 4. Sample PAPG collection process

3) According to { $\{r_1^3 = 23, r_2^3 = 821\}$, $\{r_3^1 = 3, r_3^2 = 398\}$ }, node 3 calculates the following:

$$\begin{cases} T(r_1^3) \bmod U' = (179 \times 23^2 + 839 \times 23) \bmod 1021 = 657 = (1010\underline{10001})_2 \\ T(r_2^3) \bmod U' = (179 \times 756^2 + 839 \times 756) \bmod 1021 = 987 = (11110\underline{11011})_2 \\ T(r_3^1) \bmod U' = (179 \times 3^2 + 839 \times 3) \bmod 1021 = 44 = (\underline{101100})_2, \\ T(r_3^2) \bmod U' = (179 \times 398^2 + 839 \times 398) \bmod 1021 = 180 = (101\underline{10100})_2 \end{cases}$$

and then obtains the following:

$$p_1^3 = (10001)_2 = 17,\ p_2^3 = (11011)_2 = 27,\ p_3^1 = (01100)_2 = 12,\ p_3^2 = (10100)_2 = 20.$$

Therefore, $p_3^3 = U - (p_1^3 + p_2^3) \bmod U = 18$,

and $R^3 = (p_1^3 + p_2^3 + p_3^3) \bmod U = 19$.

Node 3 obtains $D^3 = (d^3 + R^3) \bmod U = (2 + 19) \bmod 31 = 21$ and then sends {3, 21} to CH.

As shown in **Fig. 4**, after obtaining the data of all its cluster members {1, 2, 3}, CH calculates (D^1+D^2+D^3) mod 31=17, which is equivalent to $(d^1 + d^2 + d^3) = 17$ according to Property 1. Then, CH sends {17, 3} to the next hop node.

Step 3 (Seeds updating): For each node *b*, each r_c^b $(c \neq b,\ 1 \le c \le n)$ is updated by $r_c^{b\prime}$, which is the lowest *l* bit of $W(r_c^b)$, and then each r_b^c $(c \neq b,\ 1 \le c \le n)$ is updated by $r_b^{c\prime}$, which is the lowest *l* bit of $W(r_b^c)$.

6. Privacy-preservation Efficacy

Pairwise keys are used in sensor networks for confidential peer-to-peer communications. In this study, we use pairwise keys only to distribute the seeds during the initialization process. In other distributed schemes such as CPDA, SMART, and PAPF, pairwise keys are used to

encrypt the data that are exchanged during the data-reporting process session by session. The following cases may disclose the encrypted communication link between nodes *b* and *c*: ① the attacker compromises node *b* or *c*, and ② the attacker obtains the pairwise key $K_{b,c}$ that is shared between nodes *b* and *c*. Given that any compromised node does not affect the pairwise keys that are shared by other pairs of nodes [26-27], if node *b* is secure, then obtaining $K_{b,c}$ is equivalent to compromising *c*. Therefore, only the node compromise attacker is considered.

Given that the attacker may eavesdrop, compromise, and collude nodes as well as obtain the sensory data through some other ways aside from breaking the PAPG system, the privacy-preservation efficacy of PAPG is analyzed in Section 6.1 and then compared with related distributed works in Section 6.2.

6.1 Privacy-preservation Efficacy Evaluation of PAPG

With PAPG, a random node *b* uses its secret P-Gene R^b to protect its private data d^b. The data that node *b* sent to its CH is the hidden data D^b ($D^b = (d^b + R^b) \bmod U$ (8)). Therefore, the outside eavesdropper, the other sensor node, or even the BS cannot obtain the private data d^b when R^b cannot be obtained. As shown by **Theorems 2 to 7**, the attacker cannot easily obtain the P-Gene R^b. Moreover, R^b is afterward- and backward-secure according to **Theorem 4**. Therefore, the PAPG scheme can efficiently preserve data privacy. Given that the attacker may compromise the node, collude the compromised nodes, obtain the sensory data through some other ways aside from breaking the PAPG system, eavesdrop and perform a brute force attack on the obtained data, and eavesdrop along with the compromised node, Sections 6.1.1 and 6.1.2 analyze the PAPG scheme in detail.

6.1.1 Analysis of Node Compromise Attack and Node Colluding Attack

In sensor networks, the attacker may compromise a node. Thus, the attacker can read out all the data stored in this node. Moreover, when several nodes are compromised, the attacker may collude with these nodes to break the system. However, as shown in **Theorem 3**, in a certain Session s_0, a random sensory d^b is secure against (*m*-1) compromised reporting cluster members, where *m* is the reporting cluster members of that session. Even when the attacker has compromised more than (*m*-1) nodes, if the compromised reporting cluster members of node *b* are less than (*m*-1), then the attacker still cannot obtain d^b. Furthermore, as proven in **Theorem 3**, even if the attacker compromised all the (*m*-1) reporting members of *b* in a certain Session s_0, in another Session s_0', the attacker still cannot obtain the private data d^b if the reporting nodes in Session s_0' are different from those in Session s_0. To obtain a random private data item of node *b*, the attacker must obtain its Seed Table T^b. Thus, as proven in **Theorem 2**, the attacker must compromise all of its (*n*-1) cluster members. Finally, as proven in **Theorem 4**, even if the attacker obtains the sensory data through other ways aside from breaking the PAPG system, it has no use for the privacy data collected in the other sessions.

Theorem 2: (1) A certain Seed table T^b belonging to node *b* is different from that of any other node, and T^b is not a subset of any other seed Tables. Thus, it is secure when any cluster member e ($e \neq b$) is compromised. (2) Seed table T^b is exposed only if node *b* is compromised or all the (*n*-1) cluster members of node *b* are compromised.

Proof: For a certain node *b*, the exposure of T^b means that all the Seeds in T^b have been discovered by the attacker.

(1) Each node *b* generates Seeds $\{r_c^b (c \neq b, c = 1,..., n)\}$ randomly and independently; thus, any other node cannot obtain $\{r_c^b (c \neq b, c = 1,..., n)\}$. For a certain member *e*, as the Seeds sent from *b* to *e* are r_e^b, all the other Seeds are clearly confidential to *e*. In this case, T^b cannot include the other Seeds except r_b^e. Therefore, T^b is certainly different from T^e. Moreover, as the Seeds shared between T^b and T^e are only r_b^e and r_e^b, in compromising *e*, the attacker can only learn the Seed table of this node and cannot obtain the Seed table of node *b*.

(2) As each pair of $\{r_e^b, r_b^e\}$ is only shared by *b* and *e*, the attacker must compromise node *e* to obtain $\{r_e^b, r_b^e\}$. To obtain T^b, the attacker must compromise all the other (*n*-1) cluster members.□

Theorem 3: In assuming that the adversary has obtained D^b, to obtain the sensory data d^b, the adversary must compromise all the (*m*-1) members of node *b* in C_a'.

Proof: In PAPG, the data D^b sent to CH from node *b* is generated by $D^b = (d^b + R^b) \bmod U$. When the adversary has obtained D^b, to obtain the sensory data d^b, the adversary has to obtain the P-Gene R^b. From the generation of R^b, we find that to obtain R^b, the adversary must obtain all the seeds that node *b* shared with the members in C_a'.

From the proven process of **Theorem 2**, we find that each pair of $\{s_e^b, s_b^e\}$ is shared only by Nodes *b* and *e*. In addition, no relationship exists between $\{s_e^b, s_b^e\}$ and the other seeds. If node *b* is secure, then the attacker must compromise node *e* to obtain $\{s_e^b, s_b^e\}$. Thereafter, to determine $\{s_c^b (c \in C_a')\}$ and $\{s_b^c (c \in C_a')\}$, the attacker must compromise all these corresponding nodes, which are $\{c \neq b, c \in C_a'\}$. Thus, if the adversary only obtains D^b, then to obtain d^b, it must compromise all the (*m*-1) members of node *b* in C_a'. □

Therefore, if the compromised nodes are less than (*m*-1), the attacker cannot obtain any private data. If the compromised nodes are no less than (*m*-1), the probability that the attacker obtains the private data d^b is $\sum_{n_c=m-1}^{n-1} q^{n_c} (C_{n_c-1}^{m-1} / C_{n-1}^{m-1})(n_c / n)$.

Theorem 4: We assume that in a certain Session s_0, the adversary has obtained d^b through a certain way aside from compromising the PAPG scheme and has obtained D^b through eavesdropping. Despite this scenario, the adversary still cannot obtain the sensory data of *b* in other sessions.

Proof: Under the attack assumption, as $D^b = (d^b + R^b) \bmod U$, the adversary can obtain the P-Gene R^b of Session s_0. However, as R^b is the sum of several P-seeds, from R^b, the attacker cannot obtain each of the secret P-seeds that generate R^b. In addition, as each P-seed is updated in each Session, from the way that R^b is generated, we find that the R^b used in a random Session s_t differs from those used in the other sessions. In addition, from the way that the seeds are updated, we find no relationship between the P-Genes used in different sessions. Thus, under this attack assumption, the adversary still cannot obtain the sensory data of *b* in other sessions, which means that the PAPG is secure under this attack assumption.

6.1.2 Analysis of Eavesdropping Attack with Node Compromise Attack

In PAPG, the hidden sensory data that each reporting cluster member sends to the CH are plain. Thus, the attacker can sniff all the reported hidden sensory data by eavesdropping. However, PAPG is efficient against this attack. In detail, as proven in **Theorem 5**, even if the attacker obtains all the hidden sensory data reported to the CHs, the attacker still cannot obtain

any private sensory data d_b and any P-Gene R_b even through a brute force attack. Furthermore, as proven in **Theorem 6,** for a random cluster C_a, even if the attacker compromised a random reporting cluster member, it still cannot guess any private sensory data d_b. Finally, as proven in **Theorem 7,** for a random cluster C_a, even if the attacker compromised the CH, it still cannot guess any private sensory data d_b even through a brute force attack. For ease of description, the ID of the CH is denoted as m, and all the reporting cluster members are denoted as $\{1,2,...,m-1\}$.

Theorem 5: The attacker cannot obtain any private sensory data d_b or P-Gene R_b from the reported hidden sensory data.

Proof: (1) We assume that the attacker obtains a random $D_b(b \in C_a', b \neq CH)$ by eavesdropping. As $D_b = (d_b + R^b) \mod U$ (8) and as R_b is confidential to the attacker, the attacker cannot determine d_b from D_b. In addition, as d_b is confidential to the attacker, the attacker cannot determine R_b from D_b. Furthermore, as the range of R_b and D_b is [0, U], the range of d_b is [0, $d_{\max}$], and $U \geq d_{\max} \times n$, guessing d_b from D_b or R_b from D_b is useless.

(2) Furthermore, even if the attacker obtains all the reported hidden sensory data, it is helpless in obtaining d_b. First, no relationship exists among the hidden data belonging to different clusters. Thus, in obtaining a random d_b collected by node $b \in C_a$, the attacker unnecessarily obtains the data belonging to the other clusters. Second, in cluster C_a, the reported hidden data are $\{D_b(b \in C_a', b \neq CH)\}$, which are generated as follows:

$$D_1 = (d_1 + R^1)\ ;\ D_2 = (d_2 + R^2)\ ;\ \cdots; D_{m-1} = (d_{m-1} + R^{m-1})\ .$$

Thus, a random $D_b(b \in C_a')$ is independent of all the other $\{D_e(e \in C_a', e \neq CH, e \neq b)\}$. Even if the attacker assigns each of the integers in the sensory data range $[0, d_{max})$ to d_b and obtains the corresponding R^b from Equation (8), it cannot determine d_b or R_b from $\{D_e(e \in C_a', e \neq CH, e \neq b)\}$. In addition, although $\{D_b(b \in C_a', b \neq CH)\}$ satisfies $[(\sum_{b \in C_a', b \neq CH} D_b) + (d_{CH} + R_{CH})] \mod U = D$ and $(\sum_{b \in C_a', b \neq CH} D_b)$ can be calculated, as the attacker cannot obtain D, d_{CH}, or R_{CH} by eavesdropping, the values of D, d_{CH}, and R_{CH} are still confidential to the attacker.

We conclude that the attacker cannot obtain any private sensory data from the hidden sensory data. □

Theorem 6: If the attacker compromised a random reporting cluster member c of cluster C_a and obtains $\{D_b(b \in C_a', b \neq CH)\}$, then it still cannot obtain the values of $d_b(b \in C_a', b \neq c)$ and $R_b(b \in C_a', b \neq c)$.

Proof: If the attacker compromised node c, then it can obtain d_c and R_c. However, as each $R_b(b \in C_a')$ is independent with R_c, obtaining R_b from R_c is useless. Similarly, obtaining d_b from d_c is useless. In addition, as $D_1 = (d_1 + R^1)\ ;\ D_2 = (d_2 + R^2)\ ;\ \cdots; D_{m-1} = (d_{m-1} + R^{m-1})$, each D_b is independent of the others. Therefore, even if the attacker also obtains $\{D_b(b \in C_a', b \neq CH)\}$, it is useless in obtaining any R_b or d_b from D_b, d_c, and R_c. □

Theorem 7: If the attacker compromised the CH of cluster C_a and obtains $\{D_b(b \in C_a', b \neq CH)\}$ by eavesdropping, then it still cannot obtain the values of $d_b(b \in C_a', b \neq CH)$ and $R_b(b \in C_a', b \neq CH)$.

Proof: (1) If the attacker compromised the CH of C_a and obtains $\{D_b(b \in C'_a, b \neq CH)\}$, then it can obtain $\sum_{b \in C'_a, b \neq CH} d_b = D'$ (11). Then, for $b = 1,...,m-2$, the attacker can assign values to each d_b such as integer 1, and then it can obtain the value of d_{m-1} according to Equation (11). As more than one variable are found in Equation (11), various groups of values of $d_b(b = 1,...,m-1)$ satisfy Equation (11). In addition, as the elements in $\{D_b(b = 1,...,m-1)\}$ are independent of each other and $R_b(b = 1,...,m-1)$ are confidential, according to Equation (11) and $\{D_b(b = 1,...,m-1)\}$, only the value range of each $d_b(b = 1,...,m-1)$, which may be shorter than the original value range $[0, d_{max})$, can be determined.

(2) In addition, according to $\sum_{b \in C'_a, b \neq CH} d_b = D'$ (11) and $\{D_b(b = 1,...,m-1)\}$ ($D_b = d_b + R_b$), the attacker can obtain $(\sum_{b \in C'_a, b \neq CH} R_b) \bmod U = R'$ (12). Similar to the equation in part (1), Equation (12) and the value of R' are useless in determining the values of any $d_b(b = 1,...,m-1)$. Similarly, the value of any $R_b(b = 1,...,m-1)$ cannot be determined.

Thus, the values of $d_b(b = 1,...,m-1)$ and $R_b(b = 1,...,m-1)$ cannot be determined through this attack.

6.2 Privacy-preservation Efficacy Comparison

As analyzed in the preceding sections, PAPG can protect the private data of a random node *b* against outside eavesdroppers, other sensor nodes, and even the BS efficiently. In addition, as the hidden data that each node sends to its CH is in plain text, PAPG can work with the other secure data aggregation schemes that provide data integrity [30-32]. The following analyzes and compares these security features with related works.

With centralized schemes such as Castelluccia's scheme [11], FSP, and D-ASP [10], each node protects its sensory data with the secret data shared with the BS. Thus, all the privacy data will be disclosed when the BS is compromised. Distributed schemes such as CPDA, SMART [4], Conti's scheme [5], and PAPF [7] protect private data through node collaboration, which can also protect private data against outside eavesdroppers, other sensor nodes, and the BS.

With the PAPF scheme, the private data of a random node *b* are cracked if all its P-class members are compromised; therefore, if the cluster size of PAPG is larger than the P-class size, the privacy-preserving efficacy of PAPF is more efficient than that of the PAPG scheme. In PAPF, the P-class size is recommended as 4 and 5 because the storage overhead of PAPF increases quickly with the increase of P-class size. Thus, the PAPG scheme is more efficient in privacy protection than PAPF if its cluster size is greater than 5.

With the CPDA scheme, the private data of a random node *b* are cracked if all the cluster members of the node are compromised. Therefore, if $m > m_c$, then PAPG is more efficient than CPDA. In the SMART scheme, a random node *b* slices its private data into *J* pieces. Thereafter, *b* keeps one piece to itself and sends (*J* -1) encrypted pieces to its neighbors. With SMART, on average, if the number of compromised nodes is greater than $3(J-1)/2$, then the data privacy of node *b* may be cracked. Therefore, if $m > 3(J-1)/2$, then PAPG is more efficient than SMART. For CPDA and SMART, a design tradeoff exists between the privacy protection and communication efficiency. Thus, m_c and *J* are recommended by [4] to have a value of 3. Therefore, the PAPG scheme is more efficient in protecting privacy than CPDA and SMART if *m* is greater than 3.

With Conti's scheme, in compromising two nodes, the attacker has the opportunity to obtain the private data of node *b*. However, with PAPG, the attacker cannot obtain any private data if the compromised nodes are less than (*m*-1). Moreover, by comparing the analysis and simulation results in Conti's scheme [5], we conclude that under the same condition (i.e., cluster size and number of compromised nodes in a cluster), even if the compromised nodes in a cluster are greater than (*m*-1), the probability that the attacker can obtain the private data of *b* in Conti's scheme is higher than that in PAPG.

The cluster size of PAPG, which is higher than 5, can be obtained easily. First, as the system overhead of PAPG increases gradually with the increase in cluster size (refer to Section 8), the increase of the cluster size of PAPG is not constrained by thesystem overheads; this characteristic makes PAPG different from the other distributed schemes. Thus, in PAPG, we can set the size of the cluster as large as possible. Secondly, although the initial cluster size of most clustering algorithms is affected by certain system parameters, the desired cluster size (such as larger than 5 or smaller than 10) can be easily obtained through cluster merging or division. Therefore, considering that the wireless link of sensor network is vulnerable, we recommend the cluster size of PAPG to be larger than 7 in general. In case nodes become compromised easily, the cluster size could be set larger. Then, under the recommended cluster size, the PAPG scheme is more efficient in privacy protection than PAPF, CPDA, and SMART.

To study the effect of the the probability that one node is compromised on the privacy-preservation efficacy of PAPG and to compare the related schemes with PAPG, several simulation experiments were conducted, where 1000 sensor nodes were uniformly deployed; *q*, the probability that one node is compromised, changes between 0.05 and 0.3. **Fig. 5** compares the privacy-preservation efficacy against node compromise of PAPG, CPDA, SMART, and PAPF, where the average degree of a node is 20, and the slice number in SMART is set to 3, which is the recommended parameter value in [4]. According to the recommended cluster size, which is 3, the cluster size of CPDA is equal to 3, 4, and 5. Note that if the size of all the clusters in CPDA is 3, then the probability that the private data are disclosed becomes higher. According to the recommended P-class size, which is 4 or 5, the P-class size of PAPF is equal to 4, 5. The cluster size of PAPG is set to 7, which is the recommended lowest parameter value. If the cluster size of PAPG is higher than7, then the probability that the private data are disclosed becomes lower.

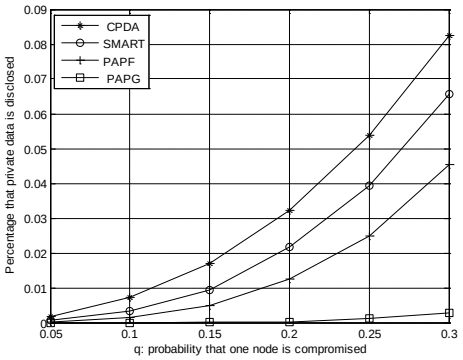


Fig. 5. Privacy Comparison under Collusion Attack with recommended parameter values (cluster size of CPDA is equal to 3, 4, and 5; the slice number in SMART is 3; the P-class of PAPF is equal to 4, 5; the cluster size of PAPG is 7)

As shown in **Fig. 5**, the larger q is, the greater is the percentage that private data are disclosed. As the theoretical analysis shows, given q, the percentage of disclosure of private data in PAPG is much lower than that of all the other schemes. This means that PAPG provides better protection for the private data than the other schemes. The reason is that given the q, for a random node b, the more nodes collaborate on private data protection, the lower is the probability that all its collaborating members are compromised. Furthermore, according to Section 8, the power consumption of PAPG is also efficient than that of the other schemes.

7. Data Aggregation Accuracy under Data Loss

When no data is lost, which is the ideal situation, all the data aggregation schemes mentioned in this paper can obtain 100% accurate aggregation results. However, these schemes react in different ways when messages are lost or delayed. In the PAPG scheme, during the intra-cluster communication, the lost messages can be detected by the CH through the received messages, so the data sent from the CH to the next hop node are an accurate aggregation result. Finally, the BS can obtain the accurate aggregation result of the received data, even if message loss occurs during the inter-cluster communication.

In SMART, FSP, and D-ASP, the aggregation result of the received data cannot be recovered when data loss occurs. No source node information is added to any message. Therefore, the lost messages cannot be detected by the CH or BS. This problem can be solved by adding the source node ID to each message. However, significant extra communication overhead is introduced.

The CPDA, PAPF, and Conti's scheme can also provide an accurate aggregation result of the received data even if message loss occurs. However, given the same message loss rate, compared with the CPDA and Conti's scheme, PAPG can transmit more data during the same limited duration because transmitting a message requires time and the number of messages to be exchanged in CPDA and Conti's scheme is several times greater than that of PAPG. In addition, cryptograph operation needs a significantly longer time than a simple calculation operations; thus, the computation time in PAPF is several times longer than that in PAPG. Thus, compared with the existing schemes, PAPG is more suitable for sensor networks with variable topology and vulnerable links.

8 Performance Evaluations

8.1 Communication Overhead

8.1.1 Communication Overhead Analysis

Only intra-cluster communication is analyzed in this study because the inter-cluster communication overhead of the PAPG scheme is the same as the one that cannot ensure data privacy preservation. The number of all the sensor nodes is denoted as N, and the globe node ID length is denoted as l_{glo} bits, which is equal to $\lceil \log N \rceil$ bits. The cluster size is denoted as n, and the intra-cluster node ID length is denoted as l_{clu} bits, which is equal to $\lceil \log n \rceil$. The length of the original sensory data item is denoted as L_{sen} bits, and the length of $list_u$ (containing reporting nodes IDs) in DASP scheme is denoted as $L_{\{ID\}}$ bits. The following analysis does not consider the packet head because we want to compare the expense of these schemes.

For convenience, similar to [7, 10], we assume that the distance between the heads of two neighboring clusters is one hop. If the distance between the heads of two neighboring clusters

is more than one hop, then a more efficient PAPG can be obtained than the others because as analyzed, with PAPG, each node only needs to send a data report, and the packet head of the report is shorter than those of the other schemes.

With PAPG, the message sent from a cluster member *b* to CH is $\{D^b, b\}$. As the length of D^b is $(L_{sen}+l_{clu})$ bits, the length of $\{D^b, b\}$ is $(L_{sen}+2l_{clu})$ bits. Based on this finding, the communication overhead of *b* increases with the original data length and the cluster size. In addition, as shown in the aforementioned analysis, when *n* increases, the privacy-preserving efficacy of PAPG improves. However, as $l_{clu}=\lceil \log n \rceil$, the communication overhead of PAPG increases slowly with the increase of *n*, while the privacy-preserving efficacy improves rapidly with the increase of *n*. Thus, the privacy-preserving efficacy of PAPG is not restricted with the communication overhead. The communication overhead of PAPF scheme is the same as that of PAPG, with no repetition occurring.

Table 5. Intra-cluster Communication Overhead

(*m:* number of reporting cluster members; *J*: number of data slices; l_{glo}: length of globe ID; l_{clu}: length of intra-cluster ID; |{M}|: length of message)

Scheme	Message sent to the CH / number	Message sent to the other nodes / number	Length of the message	Communication Overhead (bits)
CPDA	$\{F_b, ID\}$ / 1	$\{V_c^b, ID\}$ / $m-1$	$\lvert\{F_b, ID\}\rvert \geq (L_{sen}+l_{clu}+l_{glo})$ $\lvert\{V_c^b, ID\}\rvert \geq (L_{sen}+l_{glo})$	$\geq [m(L_{sen}+l_{glo})+l_{clu}]$
SMART	/	$\{S_{b,c}\}$ / $J-1$ $\{S_b\}$ / 1	$\lvert\{S_{b,c}\}\rvert = L_{sen}$, $\lvert\{S_b\}\rvert > L_{sen}$	$> (JL_{sen}+l_{clu})$
FSP	$\{\hat{D}_b, \hat{A}_b\}$ / 1	/	$\lvert\{\hat{D}_b, \hat{A}_b\}\rvert = 2\max\{L_{sen}, l_{glo}+1\}$	$2\max\{L_{sen}, l_{glo}+1\}$
D-ASP	$\{\hat{D}_b, \hat{A}_b, list_b\}$ or $\{\hat{D}_v, \hat{A}_v\}$ / 1	Control information	$\lvert\{\hat{D}_b, \hat{A}_b, list_b\}\rvert$ $= (2\max\{L_{sen}, l_{glo}+1\}+L_{(ID)})$	$(2\max\{L_{sen}, l_{glo}+1\}$ $+L_{(ID)})$
Cont's	$\{d_b + H(seed, k_i)\}$ / 1	$\{S, \{s_i, H(k_i)\}\}$ / 2	$\lvert\{d_b + H(seed, k_i)\}\rvert = L_{sen}$ $\lvert\{S, \{s_i, H(k_i)\}\}\rvert = AmL_{sen}/4$	$[1+(Am/2)]L_{sen}$
PAPF	$\{D_b, ID'\}$ / 1	/	$\lvert\{D_b, ID'\}\rvert = (L_{sen}+2l_{clu})$	$(L_{sen}+2l_{clu})$
PAPG	$\{D_b, ID'\}$ / 1	/	$\lvert\{D_b, ID'\}\rvert = (L_{sen}+2l_{clu})$	$(L_{sen}+2l_{clu})$

The communication overheads of all the other schemes are listed in **Table 5**. The table shows that the communication overhead of PAPG is the same as that of the PAPF scheme. In addition, as $J \geq 3$, $l_{glo} > l_{clu}$, $m \geq 3$, and $A \geq 2$, the communication overhead of PAPG is lower than those of the other distributed schemes. For FSP to be more efficient than PAPG in communication, the inequality $2\max\{L_{sen}, (l_{glo}+1)\} < (L_{sen}+2l_{clu})$ must be satisfied. This inequality is satisfied if and only if $L_{sen} < 2l_{clu}$. In satisfying $L_{sen} < 2l_{clu}$, even if $l_{clu}=5$, $d_{\max} <$ 1024 or $N <$ 512. Thus, the satisfaction of $L_{sen} < 2l_{clu}$ is limited to the system parameters. In other words, the PAPG scheme is also more efficient than the centralized schemes FSP and D-ASP in communication in general.

Numerical Example 3 ***(communication overhead of each node).*** **Table 6** shows some numerical results of the schemes mentioned, with the changing of the parameters L_{sen} and *n*. In this case, the length of the intra-cluster ID in PAPG also increases with the increase of *n*. **Table 6** shows that as analyzed, the communication overhead of PAPG is light and increases gradually with the system parameters because in this case, the communication overhead of PAPG only increases with l_{clu} and $l_{clu}=\log\lceil n \rceil$. Notably, when the cluster size is 36, as

$\lceil \log 36 \rceil$ is equal to $\lceil \log 22 \rceil$, the communication overhead of node *b* under this parameter value is equal to that of 22. Even when the cluster size increases to 64, node *b* needs to transmit only two extra bits. In addition, given l_{clu}, when L_{sen} increases, the communication overhead of PAPG increases linearly but slower than that of the other privacy-preservation schemes.

Table 6. Numerical results of intra-cluster communication overhead for each node (bits) (given m =3 and J=3, where m and J are the numbers of the collaboration nodes of CPDA and SMART, respectively)

Parameter values	CPDA	SMART	FSP	Conti's	PAPG/PAPF
n=8 ($l_{clu}=3$), $L_{sen}=11$	71	36	≥ 22	44	17
n=12 ($l_{clu}=4$), $L_{sen}=11$	71	37	≥ 22	44	19
n=16 ($l_{clu}=4$), $L_{sen}=11$	71	37	≥ 22	44	19
n=20 ($l_{clu}=5$), $L_{sen}=11$	71	38	≥ 22	44	21
n=20 ($l_{clu}=5$), $L_{sen}=12$	74	41	≥ 24	48	22
n=20 ($l_{clu}=5$), $L_{sen}=13$	77	44	≥ 26	52	23

The given example also shows that although the parameters that affect these privacy-preservation schemes are chosen to have the recommended smallest value, the communication overhead of PAPG is lower than those of these schemes, and the advantage is more significant when compared with the distributed schemes. Parameter *m* has no effect on the communication overhead of PAPG. Even when *m* increases from the minimum value to *n*, the communication overhead of PAPG remains the same. Meanwhile, the communication overheads of other schemes increase quickly if *m* is equal to *n*. Thus, compared with these distributed schemes, the PAPG is more efficient in preserving privacy and consumes less power.

8.1.2 Simulations

In this subsection, some simulation experiments were conducted to study the effect of the percentage of reporting nodes on the communication overheads of the schemes mentioned. Similar to [7, 10], this study assumes that all the sensor nodes are to be uniformly deployed in a 6×6 square area, which contains 36 cells. The sensor nodes in each cell form a cluster. The distance between two cells ranges from 1 ho to 3 hops. The size of the sensory data L_{sen} is 16 bits; the number of sensor nodes in each cell is set to 8; and the size of the node ID is set to 9 bits. We note that setting the node ID to 9 bits does not limit our scheme because according to the preceding analysis, in the PAPG scheme, the communication overhead of a random node *b* is related to L_{sen} and *n*. In addition, the cluster size in CPDA and the slice number in SMART are set to 3, which are the recommended parameter values in [4]. According to the analysis in Section 6, under these parameter values, the PAPG scheme is more efficient in privacy preservation than CPDA and SMART. The communication overheads of the PAPG scheme and the PAPF scheme are the same, so they are not compared in this study.

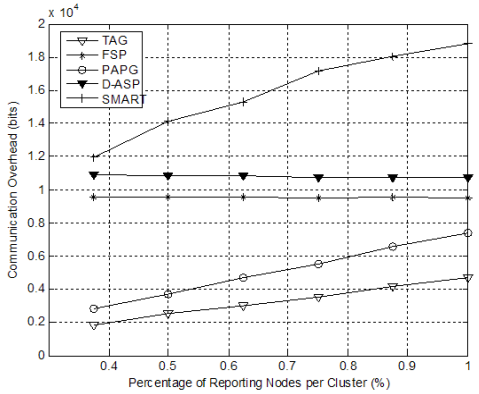


Fig. 6. Communications Overhead versus Percentage of Reporting Nodes

Fig. 6 shows the simulation results when the percentage of reporting nodes in each cluster changes between 37.5% and 100%. According to Fig. 5, the communication overhead of PAPG is higher than that of TAG but lower than that of all the other schemes. The simulation results in [4] show that the communication consumption of SMART is close to that of D-ASP when 25% of the nodes are reporting. The methods adopted by SMART are used when the number of reporting nodes in the cluster is less than m (m=4 in this study). Therefore, our simulation result and that in [4] show that the proposed PAPG is more efficient in communication than D-ASP. Thus, PAPG is more suitable than the other privacy-preserving schemes not only when all the nodes are reporting but also when the number of the reporting cluster nodes is changing.

8.2 Computational Overhead

In the PAPG scheme, during the data reporting process (performed per session), each node must generate its P-Gene by calculating a polynomial function value and modular addition operations. In addition, each node must update its Seed Table by calculating 2(n-1) function values and modular addition operations. Therefore, the computational overhead of this process is $2O(n)$ addition and multiplication operations. To aggregate the data of CH, it must perform $O(n)$ addition and multiplication operations. In addition, the computational overhead of the initialization process, which is only performed when the network is deployed or when nodes are added, is suitable. During this process, each node encrypts each of its generated Seeds and decrypts all of the received Seeds. Thus, the computational overhead of this process is 2(n-1) encryption/decryption. As shown, the computational overhead of PAPG is suitable.

In the CPDA scheme, the computational overhead of each cluster member is $(2n_c+1)$ encryption/decryption and involves $O(1)$ addition and multiplication operations. The computational overhead of each CH is $O(1)$ inversion of matrix, $2(n_c-1)$ encryption/decryption, and $2o(J)$ addition and multiplication operations. In the SMART scheme, the computational overhead of each node is $2J$ encryption/decryption and involves $O(1)$ addition and subtraction operations. In the FSP and D-ASP schemes, the computational overhead of each node is $O(m)$ hash operation and involves $o(1)$ modular addition and multiplication operations. The computational overhead of Conti's scheme is $2A$ hash operation, 4 encryption/decryption, and $O(A)$ modular addition and multiplication operations. In addition, each CH needs to perform m encryption/decryption and $O(m)$ addition and multiplication

operations. In the PAPF scheme, the communication of each node involves $O(n)$ addition and multiplication operations as well as one encryption.

According to the preceding analysis, the computation overhead of PAPG is lower than those of all other schemes.

8.3 Storage Overhead

Nodes in CPDA and SMART do not need to store any information at the expense of additional high communication overhead. Two seeds are used in FSP and D-ASP, and the storage overhead is $2\max(L_{sen}, l+1)$ bits. Nodes in Conti's scheme must store K keys; thus, the storage overhead of which is $K L_{sen}$ bits. In PAPF, each node b must store the seeds used to generate a μ-degree polynomial. Thus, the storage overhead is $(2n-1)(\mu+1)(L_{sen}+l_{clu})$ bits.

In PAPG, each node must store T^b. Thus, the storage overhead is $2(n-1)(L_{sen}+l_{clu})$ bits. Larger n, L_{sen}, and l_{clu} lead to larger storage overhead but also higher privacy-preservation efficacy. Thus, a design tradeoff exists between them. The storage overhead of PAPG is higher than those of all the other schemes except PAPF. However, the storage overhead of PAPG is still light. For example, if n = 20 (the corresponding l_{clu} is 5 bits) and $L_{sen}=16$ bits (the range of the sensory data is between 0 and 16383), then each node only needs to store 100 bytes of information. A cluster with a size of 20 is a large cluster. In addition, the range of sensory data between 0 and 16383, which contains most of the sensory data ranges, is a wide range, although the storage overhead is only 100 bytes. In summary, the storage overhead and computation overhead of PAPG are light and suitable for sensor networks.

8.4 Comparison of Power Consumption

The overall comparison in terms of system overheads is listed in **Table 7**.According to this table, compared with the distributed PAPF, the PAPG scheme consumes less computation overhead but has the same communication overhead. Compared with the distributed CPDA and SMART, the PAPG scheme consumes less communication and computation overheads. Compared with the centralized FSP and D-ASP, PAPG is more efficient in communication while having the same computation overhead. During the data-reporting process, energy-consuming operations include communication and calculation. Thus, PAPG is more efficient than the other schemes in terms of power consumption.

Table 7. Comparison between PAPG and other schemes in system overhead per node during data-reporting process (Note: $l_{clu} < l_{glo}$, $m \geq 3$, $J \geq 3$, $l_{clu} < L_{sen}$)

Scheme	Privacy-preservation category	Communication overhead (bits)	Unencrypted scheme	Storage overhead (bits)
PAPG	distributed	$(L_{sen}+2l_{clu})$	Yes	$2(n-1)(L_{sen}+l_{clu})$
PAPF	distributed	$(L_{sen}+2l_{clu})$	No	$(2n-1)(\mu+1)(L_{sen}+l_{clu})$
CPDA	distributed	$[m(L_{sen}+l_{glo})+l_{clu}]$	No	No
SMART	distributed	$>(JL_{sen}+l_{clu})$	No	No
FSP	centralized	$2\max\{L_{sen}, l_{glo}+1\}$	Yes	$2\max\{L_{sen}, (l+1)\}$
D-ASP	centralized	$(2\max\{L_{sen}, l_{glo}+1\}+L_{\{ID\}})$	Yes	$2\max\{L_{sen}, (l+1)\}$

9. Conclusions

In this study, an erasable data-hiding technique and a collaboration-based privacy-preservation scheme PAPG are proposed. Extensive analysis and simulations show that, compared with the centralized schemes FSP and D-ASP, PAPG not only avoids the single-point problem but also decreases the power consumption and is more resistant to vulnerable links. Compared with the distributed CPDA and SMART schemes, the PAPG scheme preserves privacy more efficiently while also consuming less power and ensuring suitable storage overhead. Compared with the distributed PAPF, the PAPG scheme preserves privacy more efficiently while also consuming less power and less storage overhead. Consequently, the proposed PAPG is more suitable for use in sensor networks. Our future work will involve designing private-preservation data aggregation schemes for general aggregation functions and robust private-preservation data aggregation schemes to protect against malicious attacks.

References

[1] I.F Akyildiz, W Su, Y Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp.393-422, Mar. 2002. Article (CrossRef Link)

[2] S Madden, M Franklin, J Hellerstein, and W Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. yst.Rev.*, vol. 36, pp.131-146, 2002. Article (CrossRef Link)

[3] J Xu, G Yang, Z Chen, Q Wang, "A survey on the privacy-preserving data aggregation in wireless sensor networks," *IEEE Journals & Magazines*, vol. 12, pp.162-180, May. 2015. Article (CrossRef Link)

[4] W He, X Liu, and H Nguyen, et al., "PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks," in *Proc. of IEEE INFOCOM 2007*, pp. 2045-2053, 2007. Article (CrossRef Link)

[5] M Conti, L Zhang, S Roy, *et al.*, "Privacy-preserving robust data aggregation in wireless sensor networks," *Security and Communication Networks*, vol. 2, n 2, pp. 195-213, March/April 2009. Article (CrossRef Link)

[6] S Huang, S Shieh, J Tygar, "Secure encrypted-data aggregation for wireless sensor networks," *Wireless Networks*, vol.16, pp. 915-927, Mar. 2010. Article (CrossRef Link)

[7] W Zeng, Y Lin, J Yu, S He and Lei Wang, "Privacy-preserving Data Aggregation Scheme Based on the P-Function Set in Wireless Sensor Networks," *Adhoc & Sensor Wireless Networks*, vol. 21, pp. 21-58, Jan/Feb. 2014. Article (CrossRef Link)

[8] T Jung, F Mao, X Li, et al., "Privacy-preserving data aggregation without secure channel: multivariate polynomial evaluation," in *Proc. of INFOCOM 2013: 32th IEEE International Conference on Computer Communications*, pp.2634-2642, 2013. Article (CrossRef Link)

[9] C Castelluccia, E Mykletun, G Tsudik, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks," in *Proc. of MobiQuitous 2005*, pp. 109-117, 2005. Article (CrossRef Link)

[10] T Feng, C Wang, and W Zhang, *et al.*, "Confidentiality Protection for Distributed Sensor Data Aggregation," in *Proc. of IEEE INFOCOM*, pp.131-146, 2008. Article (CrossRef Link)

[11] C Castelluccia, A Chan, E Mykletun, *et al.*, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, pp. 1-36, Mar. 2009. Article (CrossRef Link)

[12] R Rivest, L Adleman, M Dertouzos, "On data banks and privacy homomorphism. Foundations of Secure Computation," *New York: Academic Press*, pp.169-179, 1978. Article (CrossRef Link)

[13] J Girao, D Westhoff, and M. Schneider, "CDA: concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Proc. of 2005 IEEE International Conference on Communications, 2005. (ICC 2005)*, pp.3044-3049, 2005. Article (CrossRef Link)

[14] Q. Zhou, G. Yang, and L. He, "A Secure-Enhanced Data Aggregation Based on ECC in Wireless sensor Networks," *Sensors (Basel, Switzerland)*, vol.14, pp. 6701-6721, Apr.2014. Article (CrossRef Link)
[15] L Yang, C Ding, and M Wu, "RPIDA: Recoverable Privacy-preserving Integrity-assured Data Aggregation Scheme for Wireless Sensor Networks," *KSII Transactions on Internet and Information Systems*, vol. 9, pp. 5189-5208, Dec. 2015. Article (CrossRef Link)
[16] W Zhang, C Wang, T Feng, "GP2S: Generic privacy-preservation solutions for approximate aggregation of sensor data," in *Proc. of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, PerCom 2008, pp.179-184, 2008. Article (CrossRef Link)
[17] M Groat, W He, S Forrest, "KIPDA: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *Proc. of INFOCOM 2011: 30th IEEE International Conference on Computer Communications*, pp. 2024-2032, 2011. Article (CrossRef Link)
[18] H Zhang, Y Shu, P Cheng, and J Chen, "Privacy and Performance Trade-off in Cyber-Physical Systems," *IEEE Network*, vol. 30, pp. 62-66, March-April, 2016. Article (CrossRef Link)
[19] Z Shi, R Sun, R Lu, Le Chen, J Chen, X Shen, "Diverse Grouping-Based Aggregation Protocol With Error Detection for Smart Grid Communications," *IEEE Transactions on Smart Grid*, vol.6, no.6, pp.2856 - 2868, July, 2015. Article (CrossRef Link)
[20] H Bao and R Lu, "A New Differentially Private Data Aggregation With Fault Tolerance for Smart Grid Communications," *IEEE Internet of Things Journal*, vol.2, no.3, pp.248 - 258, June 2015. Article (CrossRef Link)
[21] J Won, C Ma, D Yau and N Rao, "Proactive fault-tolerant aggregation protocol for privacy-assured smart metering," in *Proc. of IEEE INFOCOM*, pp. 2804-2812, 2014. Article (CrossRef Link)
[22] J Zhao, T Jung, Y Wang and X Li., "Achieving differential privacy of data disclosure in the smart grid," in *Proc. of IEEE INFOCOM*, pp. 504-512, 2014. Article (CrossRef Link)
[23] Z Fu, K Ren, J Shu, X Sun, and F Huang, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. E98-B, pp.190-200, Jan.2015. Article (CrossRef Link)
[24] D Djenouri, M Bagaa, "Synchronization Protocols and Implementation Issues in Wireless Sensor Networks: A Review," *IEEE Systems Journal*, vol.10, No. 2, pp. 617-627, Feb. 2016. Article (CrossRef Link)
[25] X Guan, L Guan, X Wang, "A novel energy efficient clustering technique based on virtual hexagon for wireless sensor networks," *International Journal of Innovative Computing, Information and Control*, vol. 7, pp. 1891-1904, Apr. 2011. Article (CrossRef Link)
[26] W Zhang, M Tran, S Zhu, *et al.*, "A random perturbation-based scheme for pairwise key establishment in sensor networks," in *Proc. of MobiHoc'07: Proceedings of the Eighth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 90-99, 2007. Article (CrossRef Link)
[27] F Ali, B Mehdi, S Hossein, *et al.*, "A high performance and intrinsically secure key establishment protocol for wireless sensor networks," *Computer Networks*, vol. 55, pp. 1849-1863, Aug. 2011. Article (CrossRef Link)
[28] P Mukherjee, S Sen, "Comparing reputation schemes for detecting malicious nodes in sensor networks[J]," *Computer Journal*, vol. 54, pp. 482-489, Mar. 2011. Article (CrossRef Link)
[29] H Zhang, P Cheng, L Shi, and J Chen, "Optimal DoS Attack Scheduling in Wireless Networked Control System," *IEEE Transactions on Control System Technology*, Vol. 24, pp. 843-852, May 2016. Article (CrossRef Link)
[30] M Rezvani, A Ignjatovic, E Bertino, and S Jha, "Secure Data Aggregation Technique for Wireless Sensor Networks in the Presence of Collusion Attacks," *IEEE transactions on Dependable and Secure Computing*, vol. 12, pp.98-110, Jan. 2015. Article (CrossRef Link)
[31] S Zhu, S Setia, S Jajodia, *et al.*, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks," in *Proc. of IEEE Symposium on Security and Privacy*, pp.259-271, 2004. Article (CrossRef Link)

[32] L. Zhu, Z. Yang, M. Li, and D. Liu, "An Efficient Data Aggregation Protocol Concentrated on Data Integrity in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1-9, Jun. 2013. Article (CrossRef Link)

Weini Zeng is a Senior Engineer in the 716th Institute of China Shipbuilding Industry Corporation, China. She received her Ph.D. degree from Hunan University, China, in 2007 and 2011, respectively. Her current research interests include sensor networks and information security

Peng Chen is a Senior Engineer in the 716th Institute of China Shipbuilding Industry Corporation, China. He received his Ph.D. degree from National Defense Science and Technology University, China, in 2007. His current research interests include trust computing and information security.

Hairong Chen is a Senior Engineer in the 716th Institute of China Shipbuilding Industry Corporation, China. His current research interests include trust computing and information security.

Shiming He is a lectuer in School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China. She received her Ph.D. degree in computer application in 2013. Her current research interests include privacy preserving, wireless network and moblie computing.