Information Sciences 390 (2017) 82–94

Contents lists available at ScienceDirect

Information Sciences

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

An efficient privacy-preserving compressive data gathering scheme in WSNs

Kun Xie[a,b,*], Xueping Ning[a], Xin Wang[b], Shiming He[c], Zuoting Ning[a], Xiaoxiao Liu[d], Jigang Wen[e], Zheng Qin[a]

[a] *College of Computer Science and Electronics Engineering, Hunan University, Changsha, China*
[b] *Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA*
[c] *School of Computer and Communication Engineering, Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, China*
[d] *State Grid HuNan Electric Power Company Research Institute, Changsha, China*
[e] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*

ARTICLE INFO

Article history:
Received 16 November 2015
Revised 4 November 2016
Accepted 31 December 2016
Available online 31 December 2016

Keywords:
Homomorphic encryption function
Compressive sensing
Privacy-preserving
WSNs

ABSTRACT

Because of the strict energy limitation and the common vulnerability of Wireless Sensor Networks (WSNs), providing efficient and secure data gathering in WSNs becomes an essential problem. Compressive data gathering, which is based on the recent breakthroughs in compressive sensing theory, has been proposed as a viable approach for data gathering in WSNs at low communication overhead. Nevertheless, compressive data gathering is susceptible to various attacks in the presence of the open wireless medium. In this paper, we propose a novel Efficient Privacy-Preserving Compressive Data Gathering Scheme, which exploits homomorphic encryption functions in compressive data gathering to thwart the traffic analysis/flow tracing and realize the privacy preservation. This allows the proposed scheme to possess the two important privacy-preserving features of message flow untraceability and message content confidentiality. Extensive performance evaluations and security analyses demonstrate the validity and efficiency of the proposed scheme.

1. Introduction

Wireless Sensor Networks (WSNs) are increasingly deployed in critical security applications [4,5,40,53,56,60] such as environment monitoring, event detection, target counting and tracking. WSNs usually consist of a large number of low-cost sensor nodes that have extremely limited sensing, computation, and communication capabilities [29,59]. As sensor nodes are prone to attacks in remote and hostile environments, security issues such as data confidentiality are extremely important [20–22,28,32,39,47,48]. The efficiency and security of data transmission in WSNs is attracting more and more attention.

Conventionally, data gathering in WSNs [46,54,55] is done by in-network data compression in which the sensory readings are compressed by exploiting the spatial correlation of the sensed data at the sink node [1,41]. To gather data from N sources, in-network compression approaches need $O(N^2)$ single-hop transmissions in the worst case, which causes a high communication overhead.

[*] Corresponding author at: College of Computer Science and Electronics Engineering, Hunan University, Changsha, China
E-mail address: cskxie@gmail.com (K. Xie).

http://dx.doi.org/10.1016/j.ins.2016.12.050
0020-0255/© 2017 Elsevier Inc. All rights reserved.

As a ground-breaking signal processing technique developed in recent years, compressive sensing [15,26] can accurately reconstruct sparse signals with a relatively small number of random measurements. Compressive data gathering has been exploited to reduce the communication overhead in WSN [31,49,50,61]. Instead of letting each node send a message to the sink, the data sample from each node is multiplied with a measurement vector of M random values; the partial projected results at each non-leaf node are summed along the routing paths (tree) [30] to the sink, and the sink at last accurately reconstructs the original sensor readings based on the small number of received messages. The communication overhead is bounded by $O(MN)$, which is much smaller than $O(N^2)$ required in traditional in-network compression approaches. Nevertheless, the security of compressive data gathering is generally overlooked in current research. Because of the open wireless medium, WSNs are susceptible to various attacks, such as eavesdropping and node compromising. These attacks may breach the security of WSNs, including confidentiality, integrity, and authenticity. Particularly, some advanced attacks, such as controllable event triggering attack (CETA) and the random event triggering attack (RETA) aiming to obtain the measurement matrix of compressive sensing, can also to be launched in WSNs. These attacks seriously impact the privacy of compressive data gathering [23]. Despite much recent interest in applying CS theory to WSN, only the research in [23] tries to study secure compressive data gathering by protecting the measurement matrix of compressive sensing. However, the methods proposed may be vulnerable to the information leakage, which results in low confidentiality. Moreover, the computation and communication overhead involved to protect the measurement matrix is very high.

In this paper, based on compressive sensing and Homomorphic Encryption Functions (HEFs) [3,36], we propose an Efficient Privacy-Preserving Compressive Data Gathering Scheme for WSNs. Our objective is to achieve the sensory readings confidentiality by preventing traffic analysis and flow tracing in WSNs. To the best of our knowledge, this is the first research effort that exploits the Homomorphic Encryption Functions in compressive data gathering to thwart traffic analysis/flow tracing and achieve the privacy conservation. We have made following contributions in the proposed scheme.

- We employ HEFs to effectively guarantee the confidentiality of sensory readings by making them difficult for attackers to recover from the summed data. As only the sink knows the decryption key, adversaries cannot decrypt the sensory readings even if some intermediate sensor nodes are compromised, or when the adversaries obtain the information on the CS measurement matrix or the routing paths (tree) for data gathering. Moreover, the coding/mixing feature of compressive data gathering can also be exploited naturally to satisfy the requirements of privacy preservation against traffic analysis and flow tracing.
- Because of the homomorphism of HEFs, message recoding at intermediate sensor nodes can be directly performed on both the encrypted messages received and encoded sensory readings, without need for knowing the decryption keys or performing expensive decryption operations on each incoming message.
- We have conducted extensive performance evaluations and security analyses. The performance evaluations on computational complexity demonstrate the efficiency of the proposed scheme. Moreover, the security analysis demonstrates that the proposed scheme can not only resist attacks from both inside and outside the network but also resist brute force attack. Moreover, the influence of HEFs on the recovery performance for compressive sensing is negligible. Thus, the compressive sensing feature can be kept in our Efficient Privacy-Preserving Compressive Data Gathering Scheme.

The rest of the paper is organized as follows. In Section 2, we present the fundamentals of compressive sensing and discuss the related research. Section 3 introduces the network model and attack model. The proposed Privacy-Preserving Compressive Data Gathering Scheme is described in Section 4. We present the performance evaluations and security analysis of the proposed scheme in Section 5. Simulation results are presented in Section 6. Finally, Section 7 concludes the work.

2. Fundamentals and related research

We first introduce the fundamentals of compressive sensing and then summarize the most relevant existing research: data gathering in WSNs. We review related research and identify the differences between our research and existing research.

2.1. Fundamentals

Compressive sensing (CS) is a ground-breaking signal processing theorem developed in recent years. According to the CS theory [15,26], a sparse signal can be recovered with a high probability by solving an optimization problem from non-adaptive linear projections which preserves the structure of sparse signals. Suppose $\mathbf{x} \in R^N$ is an unknown sparse vector where $\|\mathbf{x}\|_0 = K$ and $K \ll N$. We call K the sparsity level of $\mathbf{x}$. Then, $\mathbf{x}$ can be reconstructed by a small number of measurements from the acquisition system by solving the following problem

$$\begin{aligned} &\min_{\mathbf{x}} \|\mathbf{x}\|_0 \\ &\text{subject to} \quad \mathbf{y} = \mathbf{\Phi x} \end{aligned} \tag{1}$$

where $\mathbf{\Phi}$ is an $M \times N$ measurement matrix and the number of measurements M satisfies:

$$M \geq cK \log \frac{N}{K} \tag{2}$$

where c is a constant value.

However, Eq. (1) is intractable because it is an NP-hard problem [13]. In recent research [7,24], it has been proven that the signal **x** can be recovered by solving the following minimum l_1-norm optimization problem with a very high probability

$$\begin{aligned} &\min_{\mathbf{x}} \|\mathbf{x}\|_1 \\ &\text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{x} \end{aligned} \tag{3}$$

with the measurement matrix $\mathbf{\Phi}$ satisfying the Restricted Isometry Property (RIP) [8], expressed as

$$(1-\delta_s)\|\mathbf{x}\|^2 \le \|\mathbf{\Phi}\mathbf{x}\|^2 \le (1+\delta_s)\|\mathbf{x}\|^2 \tag{4}$$

where δ_s is a constant and $\delta_s \in [0, 1)$. From [10] and [9], we know that the Bernoulli matrix and the Gaussian random matrix satisfy the Restricted Isometry Property when M satisfies (2). In this paper, we use the Bernoulli matrix as the measurement matrix.

Many approaches have been proposed to solve the above convex optimization problem in (3), such as Matching Pursuit (MP) [33], Orthogonal Matching Pursuit (OMP) [42], and Projection onto Convex Sets (POCS) [6]. In this paper, we use OMP to recover the original sensory reading data.

2.2. Related research

Compressive sensing is becoming a new paradigm for data gathering in WSNs as it can greatly improve communication efficiency. In [2], a universal compressive wireless sensing scheme was proposed, in which sensed data are collected and sent by synchronized amplitude-modulated analog transmissions to the fusion center in a single hop network. In [31], the authors presented the first complete design to apply compressive sensing to data gathering for large-scale WSNs, which is shown to be able to reduce the global communication cost. Xiang et al. [49,50] aimed to minimize the energy consumption in data collection with compressive sensing and formulate a mixed integer programming for data recovering. Zhao et al. [61] proposed a CS-based data aggregation scheme that adopts Treelet as a sparse transformation tool to efficiently address unordered sensory data. In [12], an alternative solution for unreliable transmissions is to take more samples at the sources, so that data recovery at the sink can still be performed in the face of data loss. The detection of data anomaly is formulated as a compressive sensing problem in [45]. In [16], network coding and compressive data gathering are jointly considered to greatly reduce the transmission in WSNs. Ebrahimi et al. [17] present a decentralized method to solve the joint problem of constructing forwarding trees and link scheduling for compressive data gathering in WSNs under the physical interference model, which can achieve the objective of energy efficient data gathering with the minimal collection latency. In addition to the above research, our recent studies [51,52] use vehicles as mobile sensors and propose road condition context gathering and sharing schemes based on compressive sensing to largely reduce the messages transmitted in the vehicle Delay Tolerant Networks (DTNs).

The above studies focus on compressive data gathering without considering the security of data transmission. Very few recent studies [23,35,37,43] have made efforts to protect the secrecy of compressive sensing, among which, Yaron Rachlin [37], Adem Orsdemir [35] and Ruslan Dautov [43] consider the CS measurement matrix as an encrypted representation of the original signal and provide different methods to protect the matrix from attackers. These methods can prevent outside attack, but not inside attack. Our scheme can prevent not only inside attack but also outside attack.

Recently, Kong et al. [25] proposed a Privacy Preserving Compressive Sensing scheme for crowdsensing based trajectory recovery, which combines the homomorphic obfuscation method KVP into the compressive sensing framework to accomplish recovery accuracy and privacy preservation simultaneously. This scheme encrypts a trajectory with several other trajectories while maintaining the homomorphic obfuscation property for compressive sensing. Although it can protect against stalkers and eavesdroppers, it is used in crowd-sensing recovery to reconstruct all users' trajectories based on their trajectory correlations and is not fit for data gathering in wireless sensor networks.

Only one recent study [23] tries to support the secure compressive data gathering in WSNs. The authors proposed two statistical inference attacks on compressive data gathering, which can estimate the measurement matrix $\mathbf{\Phi}$ when the eavesdroppers collect enough data from one or more nodes. They also proposed a new compressive data aggregation scheme SCDG to improve data confidentiality. In every monitoring round, new random seeds are generated and sent from the sink to the sensor nodes, according to which new measurement vectors are generated in sensor nodes. The computation and communication overhead is high. Moreover, the measurement vectors are individually generated in each node in every monitoring round, which results in a high error rate.

In contrast, this paper exploits Homomorphic Encryption Functions in compressive data gathering to thwart traffic analysis/flow tracing and realize privacy preservation. Taking advantage of the lightweight Homomorphic Encryption Functions, our scheme is efficient with low computational complexity. Moreover, the influence of HEFs on the recovery performance for compressive sensing is negligible. Thus, the good features of compressive sensing can be kept in our Efficient Privacy-Preserving Compressive Data Gathering Scheme.

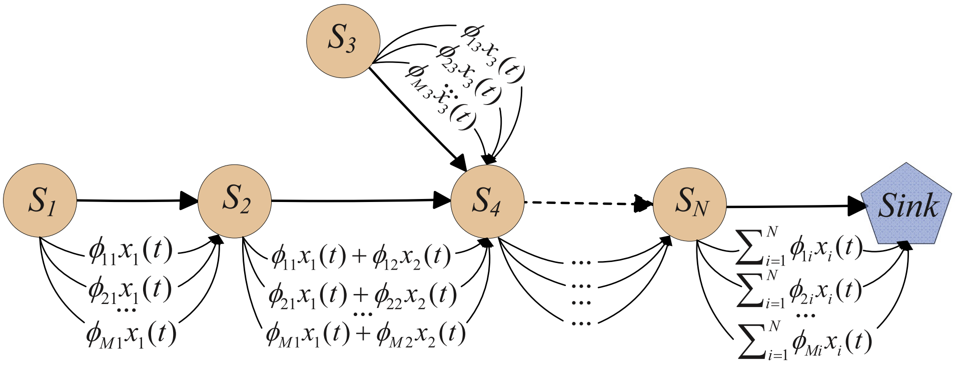

S_3
$\phi_{13}x_3(t)$
$\phi_{23}x_3(t)$
$\phi_{M3}x_3(t)$
S_1
S_2
S_4
S_N
Sink
$\phi_{11}x_1(t)$
$\phi_{21}x_1(t)$
$\phi_{M1}x_1(t)$
$\phi_{11}x_1(t)+\phi_{12}x_2(t)$
$\phi_{21}x_1(t)+\phi_{22}x_2(t)$
$\phi_{M1}x_1(t)+\phi_{M2}x_2(t)$
$\sum_{i=1}^{N}\phi_{1i}x_i(t)$
$\sum_{i=1}^{N}\phi_{2i}x_i(t)$
$\sum_{i=1}^{N}\phi_{Mi}x_i(t)$

Fig. 1. Compressive data gathering.

3. Network model and attack model

3.1. Network model

Consider a WSN that consists of one sink node and N randomly distributed sensor nodes. The ultimate goal of the WSN is to securely collect all data from sensor nodes at the sink with low cost. Without data aggregation, each node needs to send its sensory reading to the sink following a routing path; hence nodes around the sink will carry heavy traffic, as they are supposed to relay the data from the downstream nodes. To alleviate the bottleneck problem, we adopt compressive data gathering in which compressive sensing is applied to the data collection. The sink node collects data from various sensors along the aggregation paths, and the paths form a tree topology.

Let $\mathbf{x}(t)$ denote the sensory reading of the tth-round of the WSN with $\mathbf{x}(t) = (x_1(t), x_2(t), \cdots, x_N(t))^T$, where $x_i(t)$ ($i \in [1, N]$) corresponds to the reading of sensor S_i. Let $\mathbf{\Phi}$ denote an $M \times N$ measurement matrix, with the column vector ϕ_i assigned to the sensor S_i. After all nodes obtain their readings at the tth-round, each node S_i multiplies its reading $x_i(t)$ by its coefficient column vector $\boldsymbol{\phi}_\mathbf{i}$ to expand its reading to an M-dimension vector $\phi_i x_i(t)$ and transmits this encoded data vector in M messages rather than the raw data $x_i(t)$ to its upstream node. The aggregation is done by summing the coded vectors whenever they meet; therefore, the traffic load on the aggregation path is always M. After the sink collects the aggregated M-dimension vector (denote $\mathbf{y} \in R^{M \times 1}$ the aggregated M-dimension vector) rather than N raw sensory readings, a compressive sensing recovery algorithm can be used as a decoding algorithm to recover the original N sensory readings. The communication overhead is low and bounded by $O(MN)$.

Fig. 1 illustrates the basic idea of compressive data gathering. S_1 multiplies its reading $x_1(t)$ with the coefficient vector ϕ_1, and sends the encoded vector to S_2. Upon receiving the messages, S_2 multiplies its reading $x_2(t)$ with the coefficient vector ϕ_2 and then sends the sum $\phi_{j1}x_1(t) + \phi_{j2}x_2(t)$ ($j \in [1, M]$) to S_4. Similarly, each node S_i contributes to the relayed messages and re-encodes the messages by adding its own encoded data. Finally, the sink will receive a vector $\sum_{i=1}^{N} \phi_i x_i(t)$, M weighted sums of all the readings.

In the compressive data gathering scheme, the encoding process is done in a distributed fashion on each node, where each node simply performs some multiplications and summations whose computational cost can be negligibly small.

Without loss of generality, anonymous secure routing protocol [57] is deployed to assist sensor nodes to determine forwarding paths. The secure routing paths are only required to be established at the beginning and are not required to change or be re-established for each new monitoring round. The monitoring round of a packet can be hidden in the secure routing scheme, and the attackers cannot identify the monitoring round of a packet for their further analysis.

This aggregation procedure, however, may cause potential information leakage in network because the coefficient matrix $\mathbf{\Phi}$ can be estimated by an adversary through statistical inference. With a good estimation of the coefficient matrix, an attacker can easily recover the original sensory readings.

3.2. Attack model

We consider the following two attack models (as shown in Fig. 2) that can attack the confidentiality of data.

An outside attacker can be considered as a global passive eavesdropper that has the ability to observe all network links and thus all messages transmitted in the WSN. By analyzing and comparing the messages going into and out of a link, it is possible for a global outside attacker to trace flow packets in the WSNs. An inside attacker may compromise several intermediate nodes. If the intermediate nodes have the decryption keys, the message plaintext can be easily recovered.

We assume that the attacker has sufficient resources (e.g., in storage, computation and communication) to perform these advanced attacks. Both outside and inside attackers may perform more advanced traffic analysis/flow tracing techniques such as time correlation, size correlation, content correlation, and brute force.

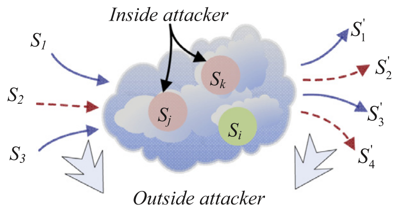


Fig. 2. Attack model.

4. Solution description

Because of the typically remote and hostile deployment environment, it is difficult to provide effective physical protection to sensors. Rather than requiring more external protection, it is essential to enforce secure compressive data gathering along the aggregation paths for high data fidelity.

There are two typical secure data aggregation categories: hop-by-hop encrypted data aggregation and end-to-end encrypted data aggregation.

In hop-by-hop encrypted data aggregation, security and data aggregation are achieved together in a hop-by-hop fashion. That is, data aggregators must decrypt every message received, aggregate the messages according to the corresponding aggregation function, and encrypt the aggregation result before forwarding it. If an inside attacker compromises an intermediate aggregator to get the decryption key, the data confidentiality is breached. Therefore, besides that the decryption/encryption process introduces higher latency, a hop-by-hop secure data aggregation protocol cannot provide data confidentiality at data aggregators.

To mitigate the drawbacks of hop-by-hop secure data aggregation, end-to-end encrypted data aggregation exploits symmetric cryptography or asymmetric key cryptography functions to provide end-to-end data confidentiality. Different from the hop-by-hop secure data aggregation, the data aggregators do not have the key to decrypt sensory data.

Therefore, to reduce the transmission delay and energy consumption, our Privacy-Preserving Compressive Data Gathering Scheme follows the end-to-end encrypted data aggregation. Particularly, we exploit the HEF to enhance security in the compressive data gathering. Before we give the detailed solution in Section 4.2, we first introduce the fundamentals of the Homomorphic Encryption Function.

4.1. Homomorphic encryption function

Homomorphic encryption is a form of encryption that allows operations on plaintext to be performed by operating on corresponding ciphertext. Let $E(x)$ denote the encryption of the message x. $E(\cdot)$ needs to satisfy the following properties:

(1) Additivity: Given the ciphertext, $E(x)$ and $E(y)$, there exists a computationally efficient algorithm $Add(\cdot,\cdot)$ such that $E(x+y)=Add(E(x),E(y))$
(2) Scalar Multiplicativity: Given $E(x)$ and a scalar t, there exists a computationally efficient algorithm $Mul(\cdot,\cdot)$ such that $E(t\cdot x)=Mul(E(x),t)$.

Benaloh [3] and Paillier [36] cryptosystems are such two additive HEFs, where the addition on plaintext can be achieved by performing a multiplicative operation on the corresponding ciphertext, i.e., $E(x_1+x_2)=E(x_1).E(x_2)$. Based on $E(t\cdot x)=E(\sum_{i=1}^{t}x)$, the following two equations can be easily derived.

$$\begin{aligned} &E(t\cdot x)=E^{t}(x)\\ &E\left(\sum_i t_i\cdot x_i\right)=\prod_i E^{t_i}(x) \end{aligned}\tag{5}$$

As Paillier cryptosystem [36] is one of the few practical homomorphic public-key cryptosystems, in this paper, we employ the Paillier cryptosystem as the HEF to apply encryption to enhance the security of compressive data gathering.

In the Paillier cryptosystem, the key can be generated following the steps below.

1. Choose two large prime numbers p and q randomly and independently of each other such that $\gcd(pq,(p-1)(q-1))=1$ (where gcd is the abbreviation of greatest common divisor). This property is assured if both primes are of equal length [36].
2. Compute $n=pq$ and $\lambda=lcm(p-1,q-1)$ (that is, λ is the lowest common multiple of $p-1,q-1$).
3. Select random integer g where $g\in\mathbb{Z}_{n^2}^{*}$ [34].
4. Function L is defined as $L(u)=\frac{u-1}{n}$.

The public (encryption) key is (n, g) and the private (decryption) key is λ. Based on the generated keys, the Paillier cryptosystem is described below.

$$\begin{aligned}&\textbf{Encryption}:\\ &\qquad\text{plaintext}\quad m(m<n),\text{random}\quad r(r<n)\\ &\qquad\text{ciphertext}\quad c=g^m\cdot r^n \bmod n^2\end{aligned} \tag{6}$$

$$\begin{aligned}&\textbf{Decryption}:\\ &\qquad\text{ciphertext}\quad c(c<n^2)\\ &\qquad\text{plaintext}\quad m=\frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n\end{aligned} \tag{7}$$

where r is a random factor in the Paillier cryptosystem.

In Paillier cryptosystem, the cost of both encryption and decryption is essentially that of one exponentiation in $\mathbb{Z}^*_{n^2}$ with the exponent roughly n(the product of two exponentiations in the encryption can be done in roughly the same time as one exponentiation).

As shown in Eq. (6), in the Paillier cryptosystem, given a message m and the public *key*(n, g), the encryption function is $E(m) = g^m \cdot r^n(mod\ n^2)$. It satisfies the following homomorphic property:

$$\begin{aligned}E(m_1)\cdot E(m_2) &= g^{m_1+m_2}\cdot(r_1\cdot r_2)^n(mod\ n^2)\\ &= E(m_1+m_2).\end{aligned} \tag{8}$$

4.2. Operations at sensor nodes and sink

Our scheme is designed based on HEF with the public-key encryption. Without loss of generality, the sensory readings would be encrypted by a public key that is known for all nodes including eavesdroppers. We assume that each sink acquires two keys, the encryption key (n, g) (public key) and the decryption key λ (private key), from an offline Trust Authority (TA). The encryption key (n, g) is published to all the other nodes. For security, the sink is required to negotiate the key pair in advance [11]. During the message transmission, we also assume that the encryption key (n, g) and the monitoring round of a packet are hidden by the secure routing scheme [27], and only authenticated intermediate nodes can obtain the information. To avoid transmitting measurement matrix $\mathbf{\Phi}$ from the sink to sensors, we adopt a simple strategy: before data transmission, the sink broadcasts a random seed to the entire network. Then, each sensor generates its own seed using this global seed and its unique identification. With a pre-installed pseudo random number generator, each sensor S_i is able to generate the corresponding column vector ϕ_i with the vector's entry values chosen from $\{-1, 1\}$. These vectors can be reproduced at the sink given that the sink knows the identifications of all sensors, and thus the sink node can obtain the measurement matrix $\mathbf{\Phi}$.

From Fig. 1, we know that the aggregated M-dimension message vector obtained at the sink using traditional compressive data gathering can be written as follows:

$$\mathbf{y}(t)=\sum_{i=1}^{N}\phi_i x_i(t),\ (\mathbf{y}(t)\in R^{M\times 1}). \tag{9}$$

which can be further written as follows.

$$\begin{pmatrix}y_1(t)\\ y_2(t)\\ \cdots\\ y_M(t)\end{pmatrix}=\begin{pmatrix}\phi_{11}&\phi_{12}&\phi_{13}&\cdots&\phi_{1N}\\ \phi_{21}&\phi_{22}&\phi_{23}&\cdots&\phi_{2N}\\ \cdots&\cdots&\cdots&\cdots&\cdots\\ \phi_{M1}&\phi_{M2}&\phi_{M3}&\cdots&\phi_{MN}\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\\ \cdots\\ x_N(t)\end{pmatrix} \tag{10}$$

$$\Rightarrow\begin{cases}\phi_{11}x_1(t)+\phi_{12}x_2(t)+\phi_{13}x_3(t)+\cdots+\phi_{1N}x_N(t)=y_1(t)\\ \phi_{21}x_1(t)+\phi_{22}x_2(t)+\phi_{23}x_3(t)+\cdots+\phi_{2N}x_N(t)=y_2(t)\\ \cdots\\ \phi_{M1}x_1(t)+\phi_{M2}x_2(t)+\phi_{M3}x_3(t)+\cdots+\phi_{MN}x_N(t)=y_M(t)\end{cases} \tag{11}$$

For secure compressive data gathering, we expect the sink to receive encrypted aggregated messages instead of the raw aggregating messages, that is,

$$\begin{cases}E_{ng}(\phi_{11}x_1(t)+\phi_{12}x_2(t)+\phi_{13}x_3(t)+\cdots+\phi_{1N}x_N(t))=y'_1(t)\\ E_{ng}(\phi_{21}x_1(t)+\phi_{22}x_2(t)+\phi_{23}x_3(t)+\cdots+\phi_{2N}x_N(t))=y'_2(t)\\ \cdots\\ E_{ng}(\phi_{M1}x_1(t)+\phi_{M2}x_2(t)+\phi_{M3}x_3(t)+\cdots+\phi_{MN}x_N(t))=y'_M(t)\end{cases} \tag{12}$$

where $E_{ng}()$ is the HEF encryption function using the public key (n, g), $y'_j(t)$ is the expected cryptal aggregating message value of $y_j(t)$ received at the sink. According to the properties of HEFs, Eq. (12) can be further written as

$$\begin{cases} E_{ng}(\phi_{11}x_1(t)) \cdot E_{ng}(\phi_{12}x_2(t)) \cdot \cdots \cdot E_{ng}(\phi_{1N}x_N(t)) = y'_1(t) \\ E_{ng}(\phi_{21}x_1(t)) \cdot E_{ng}(\phi_{22}x_2(t)) \cdot \cdots \cdot E_{ng}(\phi_{2N}x_N(t)) = y'_2(t) \\ \cdots \\ E_{ng}(\phi_{M1}x_1(t)) \cdot E_{ng}(\phi_{M2}x_2(t)) \cdot \cdots \cdot E_{ng}(\phi_{MN}x_N(t)) = y'_M(t) \end{cases} \tag{13}$$

$$\Rightarrow \begin{cases} E_{ng}^{\phi_{11}}(x_1(t)) \cdot E_{ng}^{\phi_{12}}(x_2(t)) \cdot \cdots \cdot E_{ng}^{\phi_{1N}}(x_N(t)) = y'_1(t) \\ E_{ng}^{\phi_{21}}(x_1(t)) \cdot E_{ng}^{\phi_{22}}(x_2(t)) \cdot \cdots \cdot E_{ng}^{\phi_{2N}}(x_N(t)) = y'_2(t) \\ \cdots \\ E_{ng}^{\phi_{M1}}(x_1(t)) \cdot E_{ng}^{\phi_{M2}}(x_2(t)) \cdot \cdots \cdot E_{ng}^{\phi_{MN}}(x_N(t)) = y'_M(t) \end{cases} \tag{14}$$

According to Eq. (14), our Privacy-Preserving Compressive Data Gathering Scheme is designed consisting of three algorithms: leaf node encoding, non-leaf node encoding and recoding, and sink node decoding.

Algorithm 1 shows the leaf node encoding algorithm which consists of two steps. In the first step, the leaf node S_i encrypts the message generated by itself, i.e., $E_{ng}(x_i(t))$, then performs the exponential operation on $E_{ng}(x_i(t))$ with the exponents being the M values in the vector $\phi_i = (\phi_{1i}, \phi_{2i}, \cdots, \phi_{Mi})^T$.

Algorithm 1 Leaf node encoding.

```
Input: the sensory message of leaf node $S_i$ at round $t$, denoted as $x_i(t)$, measurement vector $\phi_i = (\phi_{1i}, \phi_{2i}, \cdots, \phi_{Mi})^T$, public key $(n, g)$
Output: encrypted message vector at leaf node $S_i$, denoted as $\vec{x}'_i(t)$
Apply Eq. (6) to encrypt $x_i(t)$ using the public key $(n, g)$, then obtain the encrypted message $E_{ng}(x_i(t))$.
Perform exponential operations on $E_{ng}(x_i(t))$ with the exponents of $\phi_i$, obtain the encrypted vector $\vec{x}'_i(t) = (E_{ng}^{\phi_{1i}}(x_i(t)), E_{ng}^{\phi_{2i}}(x_i(t)), \cdots, E_{ng}^{\phi_{Mi}}(x_i(t)))^T$
```

Algorithm 2 shows the non-leaf node encoding and recoding algorithm. With HEFs, intermediate nodes are allowed to directly perform multiplication on the encrypted messages. In other words, because of the homomorphism of the HEF, encryption on the summation of messages in each node can be achieved by performing a multiplication operation on the corresponding ciphertext. Data forwarding can be achieved by operating on the encrypted messages without the need of knowing the decryption keys or performing the decryption operations.

Algorithm 2 Non-leaf node encoding and recoding.

```
Input: the sensory message of non-leaf node $S_i$ at round $t$, denoted as $x_i(t)$, the message vector received from its previous node $S_j$, denoted as $\vec{x}_j(t)$, measurement vector $\phi_i = (\phi_{1i}, \phi_{2i}, \cdots, \phi_{Mi})^T$, public key $(n, g)$
Output: encrypted message vector at non-leaf node $S_i$, denoted as $\vec{x}'_i(t)$
Apply Eq. (6) to encrypt $x_i(t)$, then obtain the encrypted message $E_{ng}(x_i(t))$.
Perform exponential operations on $E_{ng}(x_i(t))$ with the exponents of $\phi_i$, obtain $\vec{x}'_i(t) = (E_{ng}^{\phi_{1i}}(x_i(t)), E_{ng}^{\phi_{2i}}(x_i(t)), \cdots, E_{ng}^{\phi_{Mi}}(x_i(t)))^T$
According to Eq. (14), $\vec{x}'_i(t) = \vec{x}'_i(t) \circ \vec{x}_j(t)$ where $\circ$ is the Hadamard product operation defined as follows: $\forall \vec{a}, \vec{b} \in R^M$, $(\vec{a} \circ \vec{b})_i = (\vec{a})_i(\vec{b})_i$
```

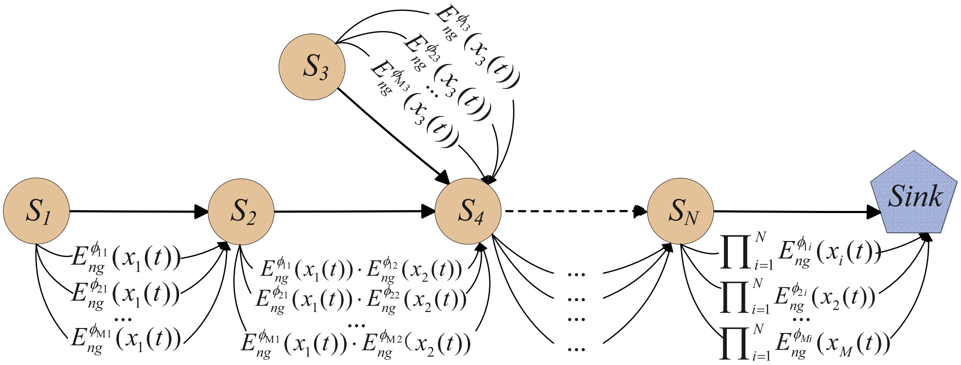
After receiving the aggregated messages of a monitoring round, the sink node can recover the raw sensory readings following Algorithm 3. In Algorithm 3, the sink node first decrypts the received message, then applies Orthogonal Matching Pursuit (OMP) [42] on the decrypted message to obtain the original sensory reading vector $\mathbf{x}(t)$.

Fig. 3 illustrates an example of the proposed scheme. Node S_1 first encrypts the message generated by itself, i.e., $E_{ng}(x_1(t))$, then performs the exponential operation on $E_{ng}(x_1(t))$ with the exponents being the M values in vector $\phi_1 = (\phi_{11}, \phi_{21}, \cdots, \phi_{M1})^T$. We can then have a vector of M encrypted data $(E_{ng}^{\phi_{11}}(x_1(t)), E_{ng}^{\phi_{21}}(x_1(t)), \cdots, E_{ng}^{\phi_{M1}}(x_1(t)))^T$. After the node S_2 receives the encrypted data vector from S_1, it should perform a multiplying operation on corresponding data in two vectors to obtain a new vector $(E_{ng}^{\phi_{11}}(x_1(t)) \cdot E_{ng}^{\phi_{12}}(x_2(t)), E_{ng}^{\phi_{21}}(x_1(t)) \cdot E_{ng}^{\phi_{22}}(x_2(t)), \cdots, E_{ng}^{\phi_{M1}}(x_1(t)) \cdot E_{ng}^{\phi_{M2}}(x_2(t)))^T$, and then forward this new vector to S_4. Finally, the sink node will receive the encrypted data vector $y'_j(t) = \prod_{i=1}^{N} E_{ng}^{\phi_{ji}}(x_i(t)))$ and recover the raw sensory data following Algorithm 3.

From the above algorithms, we can conclude that the influence of HEFs on the recovery performance for compressive sensing is negligible. Thus, the compressive sensing feature can be kept in our Efficient Privacy-Preserving Compressive Data Gathering Scheme.

Algorithm 3 Sink node decoding.

```
Input: the received encrypted message $y'_j(t)$, private key $\lambda$
1: Decrypt the received message according to Eq. (7) using the private key $\lambda$, then obtain the plaintext of $y'_j(t)$, denoted as $y_j(t)$
2: Solve the following compressive sensing recovery problem to recover the original sensory reading vector $\vec{x}(t)$ from the decrypted data $\vec{y}(t)$
       $$\begin{aligned} &\min_{\vec{x}(t)} \|\vec{x}(t)\|_1 \\ &\text{subject to} \quad \vec{y}(t) = \vec{\Phi}\vec{x}(t) \end{aligned} \qquad (14)$$
   through Orthogonal Matching Pursuit (OMP) [42].
```



S_3
$E_{ng}^{\phi_{13}}(x_3(t))$
$E_{ng}^{\phi_{23}}(x_3(t))$
...
$E_{ng}^{\phi_{M3}}(x_3(t))$
S_1
S_2
S_4
S_N
Sink
$E_{ng}^{\phi_{11}}(x_1(t))$
$E_{ng}^{\phi_{21}}(x_1(t))$
...
$E_{ng}^{\phi_{M1}}(x_1(t))$
$E_{ng}^{\phi_{11}}(x_1(t)) \cdot E_{ng}^{\phi_{12}}(x_2(t))$
$E_{ng}^{\phi_{21}}(x_1(t)) \cdot E_{ng}^{\phi_{22}}(x_2(t))$
...
$E_{ng}^{\phi_{M1}}(x_1(t)) \cdot E_{ng}^{\phi_{M2}}(x_2(t))$
...
$\prod_{i=1}^{N} E_{ng}^{\phi_{1i}}(x_i(t))$
$\prod_{i=1}^{N} E_{ng}^{\phi_{2i}}(x_2(t))$
...
$\prod_{i=1}^{N} E_{ng}^{\phi_{Mi}}(x_M(t))$

Fig. 3. Encryption model.

5. Performance evaluations and security analysis

In this section, we first evaluate the computational overhead of our proposed scheme, and then analyze its security.

5.1. Computational overhead

The computational overhead of the proposed scheme can be investigated from three aspects: leaf node encoding, non-leaf node's encoding and recoding, and sink node's decoding. As the computational overhead of the proposed scheme is closely related to the specific homomorphic encryption algorithm, in the following analysis, we will take the Paillier cryptosystem as the encryption method when necessary.

5.1.1. Computational complexity on a leaf node

To encode the sensory reading, a leaf node S_i needs one encryption operation and M exponentiations with the exponent corresponding to its associated ϕ_i. According to the Paillier cryptosystem, every encryption operation requires 2 exponentiations, 1 multiplication, and 1 modulus operation. When $n \gg m$ in the Paillier cryptosystem, the computational complexity of one encryption operation is $O(\log n)$, which is incurred for multiplication operations. Moreover, the measurement matrix adopted in this paper is a $\{-1, 1\}$ Bernoulli matrix. Therefore, the computational complexity of exponentiations in encoding is very small. The computational complexity on a leaf node is $O(\log n)$ in terms of multiplication operations.

5.1.2. Computational complexity on non-leaf node:

In addition to encoding its own sensory reading, the non-leaf node should perform M multiplication operations between the received data vector and the corresponding encoded values of their own encrypted data vector to complete the recoding procedure. Therefore, the computational complexity on non-leaf node is $O(\log n + M)$.

5.1.3. Computational complexity on sink:

According to the operations on sink, to recover the original sensory readings, the sink will first decrypt the encrypted M messages, and then apply OMP to reconstruct the original reading data. The sinks need M decryption operations and 1 OMP reconstruction operation. According to the Paillier cryptosystem, decrypting an element requires 1 exponentiation, 1 multiplication, and 1 division operation. The computational complexity of one decryption operation is $O(\log n)$ in terms of multiplication operations. Therefore, the computational complexity of decryption is $O(M \cdot \log n)$. According to OMP, the

Table 1
Comparison among our scheme and the other three schemes.

	Network coding [44]	Encrypt by HEF + Coding [18]	SCDG [23]	Our scheme
Preventing inside attack	√	√	√	√
Preventing outside attack	×	√	√	√
Computational overhead	$O(N^3)$	$O(N^3 \cdot \log n)$	$O(N^3)$	$O(M \cdot \log n) + O(K \cdot \log N)$
Space overhead	$N/(l+N)$	N^2	N^2	$M \cdot N$

computational complexity of using OMP to reconstruct the original message **x(t)** is $O(K \cdot \log N)$ where K is the sparsity of $\mathbf{x}(t)$. Therefore, the computational complexity of recovering the original sensory readings on the sink node is $O(M \cdot \log n) + O(K \cdot \log N)$.

5.2. Overhead comparisons

We compare the overhead of our scheme with three other data gathering schemes.

- SCDG proposed in [23], which encrypts data by dynamically changing the measurement matrix $\mathbf{\Phi}$ to prevent an attacker from obtaining the matrix. Dynamic matrix update will greatly increase the overhead of both the computation and communication.
- The scheme proposed by Vilela et al. [44], which exploits the mixing feature of the network coding. In this scheme, every intermediate node will receive a data packet that contains a random linear combination of all data from its former nodes. It also uses two types of coefficients to encode and decode the data in intermediate nodes. Therefore, the sink node needs two rounds of decoding processes, decrypting the coefficients and performing Gaussian elimination to recover the original data, which may reduce the algorithm efficiency. Moreover, a considerable space overhead will be incurred by the extra set of coding vectors. In comparison, we call this scheme “network coding”.
- The algorithm proposed by Fan et al. [18], which encrypts Global Encoding Vectors (GEVs) through the HEF to resist traffic analysis attack in network coding and protect the untraceability and confidentiality of the data package. Because of the homomorphism of HEFs, random linear network coding could be performed directly on the encrypted coding vectors. However, the computational complexity involved for multiplication operations is $(O(N^3 logn))$ for a GEM with N GEVs. In comparison, we call this scheme “Encrypt by HEF + Coding”.

Compressive sensing provides efficient in-network operations for data gathering in WSNs. As there are few studies on secure compressive sensing data gathering in WSNs, for fair comparison, we only compare our scheme with one CS-based scheme [23] that has a goal similar to ours. Moreover, as another good in-network technique, network coding (ONC) [14,58] has attracted much research interest because it can significantly improve the throughput and power efficiency of wireless networks with the mix of various traffic flows via algebraic operations. In addition to the scheme in [23], we compare our scheme with two other secure network coding based data transmission schemes [18,44].

Table 1 summarizes the overhead comparison results. In this table, N is the number of source nodes that collect environment data in WSNs, M is the number of rows in matrix $\mathbf{\Phi}(M \ll N)$, K is the sparsity of environment data vector $\mathbf{x} \in R^N$, n is the public key in HEF and l is the generation number of messages in the network coding scheme. Except that the Network coding [44] cannot prevent the outside attack, the other three schemes can prevent both inside and outside attack. The computational overhead of all the peer schemes reaches exponential order with N. Except for the Network coding scheme [44], as $M \ll N$, the space overhead of our scheme is much lower than Encrypt by HEF + Coding scheme [18] and SCDG [23], while the Network coding scheme cannot prevent outside attack. Therefore, it is fair to conclude that our scheme outperforms the other three schemes on thwarting inside and outside attacks at low space and computational cost.

5.3. Security analyses

5.3.1. Preventing inside attacks

To prevent inside attacks, the first step is to protect the monitoring round number. If an adversary attempts to launch a traffic-analysis attack, the adversary should first identify the packet’s monitoring round number. However, in our scheme, the monitoring round number is hidden in the secure routing scheme.

The second step is to resist attackers from analyzing the size correlation and content correlation, two widely used techniques in traffic analysis. In our scheme, the message received by an immediate node is the product of its previous nodes in topology, such that adversaries cannot obtain the size of the message. To launch the message content correlation attack, the adversary must intercept messages of the same monitoring round and determine if an intercepted message in a downstream link is a linear combination of some known messages. In our scheme, it is impossible for an adversary to achieve the content correlation between messages.

5.3.2. Preventing outside attacks

Our scheme can prevent timing attacks and protect route. According to Raymond [38], a timing attack needs to record the set of messages coming in and going out of the network as well as their respective arrival and departure times. As

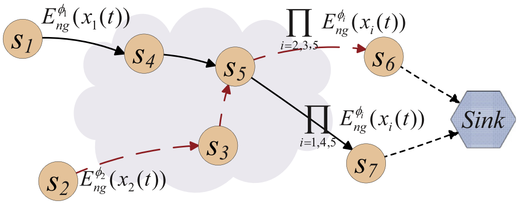

$E_{ng}^{\phi_1}(x_1(t))$
S_1
S_4
S_5
$\prod_{i=2,3,5} E_{ng}^{\phi_i}(x_i(t))$
S_6
S_3
$\prod_{i=1,4,5} E_{ng}^{\phi_i}(x_i(t))$
Sink
S_7
S_2
$E_{ng}^{\phi_2}(x_2(t))$

Fig. 4. Route protection.

shown in Fig. 4, S_1 and S_2 are the nodes where messages come from, and S_6 and S_7 are the nodes where messages go to. In our scheme, messages in every node are encrypted, and every receiving and forwarding message in the intermediate node is the mixed message of its previous hops. In Fig. 4, the messages received by nodes S_6 and S_7 are just vectors, and they are the products of nodes unknown to the adversary. Thus, it is impossible for adversaries to obtain the correlated route timing information by analyzing the messages in the two sets. Attackers cannot obtain the path of source messages and their next hop from by collecting messages from outside or recording the time of message forwarding. Then, we can prevent the attackers from inferring the route information of message forwarding and the network topology by timing attack.

In addition to preventing timing attack, we can also prevent adversaries from identifying the source of a message. For a global adversary, it is easy to trace the forwarding path of a message if the encrypted message remains the same during its forwarding. In our scheme, the encrypted message is changed after getting through every node, and so it is hard for adversaries to trace the path or find the source of a message.

5.3.3. Brute force attack

When adversaries adopt brute force attack to obtain the sensory readings in our scheme, the computational complexity in the brute force attack should include three parts.

First, one should know the monitoring round number of messages. Assuming that we have collected the messages from W monitoring rounds, then the complexity of determining that there are M messages from one monitoring round is $O(C_{MW}^{M})$.

Second, because the data gathering in our scheme is along aggregation paths, one should infer the topological structure to obtain the measurement matrix to further recover the original sensory data. Based on the messages found from the same monitoring round, if one wants to infer the topological structure to build the measurement matrix with N columns (where N is the number of sensor nodes), the complexity of identifying the first column is $O(N)$. Similarly, after identifying the first column, the complexity of identifying the second column is $O(N-1)$. For a WSN consisting of N sensor nodes, the overall complexity of inferring the correct topological structure to build the measurement matrix required in the compressive sensing theory is $O(N!)$.

Third, one should infer the real message under the condition of not knowing the secret key. Attackers can collect data from a compromised node, but the data on every node is the product of encrypted data from the compromised node and its previous nodes. If attackers want to infer messages of compromised node and its former nodes from the attacked node while not knowing the messages from former nodes, they need to estimate the maximum value of messages and determine the real data in the range of zero and the maximum. The number of nodes in network topological structure is usually large; thus, an attacker cannot compromise every node to collect data but can only estimate the real message in part from the compromised nodes. Assuming $q = max(x_i(t))$ is the estimated maximum, the complexity of finding the correct data of one node in the range of 0 and q is $O(q+1)$. There are N nodes in our topological structure; then the complexity of finding all the plaintext on every node is $O((q+1)^N)$.

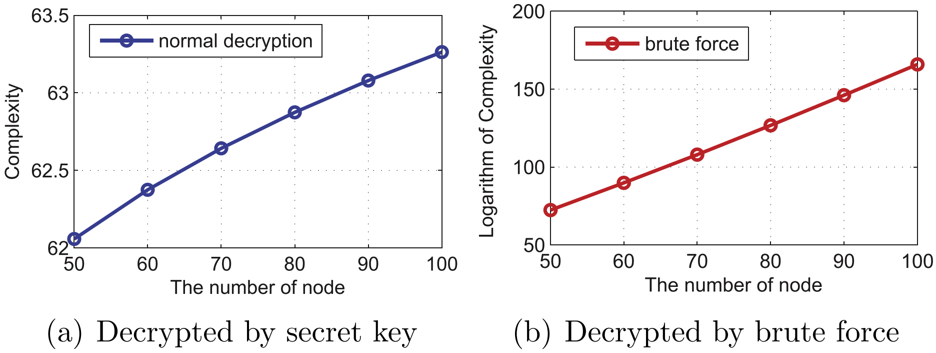
Based on the analysis above, the overall complexity for an attacker to infer the message by brute force is $O(C_{MW}^{M}(N!+(q+1)^N))$, which is in factorial and exponential order. Therefore, if one wants to infer our message by brute force attack, it is impossible with the huge computation power needed.

The complexities of decryption by secret key at the sink and brute force are related to the number of nodes in the topological structure. With increasing node number, the complexity increases almost linearly when decrypted by the secret key, and increases exponentially when decrypted by the brute force. As shown in Fig. 5, when the node number increases from 50 to 100, the complexity of normal decryption on the sink increases from 62 to 63.3, and the logarithm of complexity of brute force increases from 72 to 165.

6. Simulation

We implement our scheme and the Advanced Encryption Standard (AES) [19] for performance comparisons. **AES** is a symmetric block cipher that can encrypt and decrypt information. It is a standardized encryption algorithm and has become the default choice in numerous applications. In this comparison, our scheme is called CS-HE, and we implement it in the network for data gathering. Different from our scheme, we implement the AES encryption algorithm in the network where each intermediate node needs to make decryption and encryption operation upon a message's arrival.

We take the end-to-end delay as the performance metric to evaluate performance. The above two algorithms are evaluated through extensive simulations using NS-2. In the simulations, 100 nodes are generated randomly in a 600 m × 600 m



(a) Decrypted by secret key (b) Decrypted by brute force

Fig. 5. Complexity of decryption when K=4, W=5, q=10, M=8 and n=1000.

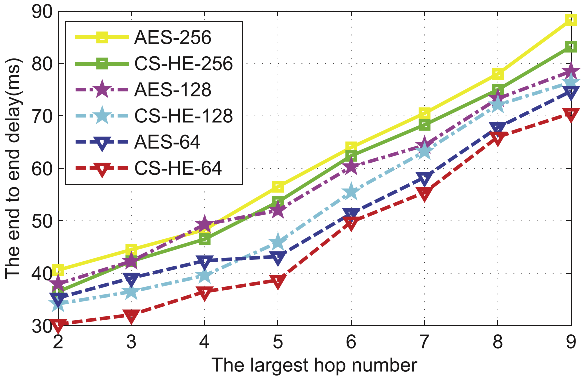


Fig. 6. End to end delay under different hop numbers.

area. The sink node is located in the center of the area. The maximum communication range of each node is set to 70 m. All the simulation results are obtained by averaging over 20 runs of simulations. We use the Bernoulli matrix as the measurement matrix for compressive sensing which is generated following the descriptions in Section 4.2. To evaluate the performance of the proposed scheme under different packet sizes, we vary the simulated packet size from 64 to 256.

Fig. 6 shows the simulation results. The end-to-end delay of our scheme is correlated with the packet size and the largest hop number from all sensor nodes to the sink. As expected, the end-to-end delay increases with the increase of the number of hops, as more node participation in data relays would invoke more homomorphic operations. Meanwhile, the delay increases with increasing packet size.

Compared with AES algorithm, our scheme has much lower end-to-end delay. AES algorithm requires every intermediate relay node to decrypt the messages received before making arithmetical operation on them; and then encrypts the operation results before forwarding. Both encryption and decryption operations introduce extra network latency. In contrast, our scheme frees the intermediate relay nodes from the complicated encryption and decryption operations. Intermediate nodes carry out arithmetical operation on the ciphertext as if they were plaintext, which saves much time for the whole process.

7. Conclusion

In this paper, we propose an Efficient Privacy-Preserving Compressive Data Gathering Scheme to protect against traffic analysis and flow tracing in WSNs. With the lightweight homomorphic encryption exploited, the proposed scheme offers two significant privacy preserving features, packet flow untraceability and message content confidentiality, which can efficiently thwart traffic analysis/flow tracing attacks. Moreover, with homomorphic encryption, the proposed scheme keeps the essence of compressive data gathering, and each sink can recover the original sensory reading through CS reconstruction algorithm after the sink decrypts the messages received. We have performed extensive performance evaluations and security analysis, which demonstrates that our scheme not only has good security features to protect privacy but also has a low computation and communication overhead.

Acknowledgments

The research is supported by the Prospective Research Project on Future Networks (Jiangsu Future Networks Innovation Institute) under Grant No. BY2013095-4-06, the National Natural Science Foundation of China under Grant Nos. 61572184, 61300219, 61472283, 51575167, 61272546, and 61472131, the Science and Technology Key Projects of Hunan Province under Grant Nos. 2015TP1004, 2016JC2075, and U.S. NSF CNS 1526843.

References

[1] S. An, H. Yang, J. Wang, N. Cui, J. Cui, Mining urban recurrent congestion evolution patterns from gps-equipped vehicle mobility data, Inf. Sci. (Ny) 373 (2016) 515–526.

[2] W. Bajwa, J. Haupt, A. Sayeed, R. Nowak, Compressive wireless sensing, in: Proceedings of International Conference on Information Processing in Sensor Networks,IPSN, 2006, pp. 134–142.

[3] J. Benaloh, Dense probabilistic encryption, in: Proceedings of the Workshop on Selected Areas of Cryptography, 1994, pp. 120–128.

[4] M.Z.A. Bhuiyan, G. Wang, J. Cao, J. Wu, Sensor placement with multiple objectives for structural health monitoring, ACM Trans. Sensor Netw. (TOSN) 10 (4) (2014) 68.

[5] M.Z.A. Bhuiyan, G. Wang, A.V. Vasilakos, Local area prediction-based mobile target tracking in wireless sensor networks, IEEE Trans. Comput. 64 (7) (2015) 1968–1982.

[6] E.J. Cands, J.K. Romberg, Signal recovery from random projections, in: Proceedings of Electronic Imaging, 2005, pp. 76–86.

[7] E.J. Cands, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.

[8] E.J. Cands, T. Tao, Decoding by linear programming, IEEE Trans. Inform. Theor. 34 (4) (2004) 435–443.

[9] E.J. Cands, T. Tao, Decoding by linear programming, IEEE Trans. Inform. Theor. 51 (12) (2005) 4203–4215.

[10] E.J. Cands, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inform. Theor. 52 (12) (2006) 5406–5425.

[11] Y. Challal, H. Seba, Group key management protocols: a novel taxonomy, Int. J. Inf. Technol. 2 (1) (2005) 105–118.

[12] Z. Charbiwala, S. Chakraborty, S. Zahedi, Compressive oversampling for robust data transmission in sensor networks, in: Proceedings of IEEE INFOCOM, 2010, pp. 1–9, doi:10.1109/INFCOM.2010.5461926.

[13] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 33–61.

[14] W. Chen, K. Letaief, Z. Cao, Opportunistic network coding for wireless networks, in: Proceedings of IEEE ICC, 2007, pp. 4634–4639.

[15] D. Donoho, Compressed sensing, IEEE Trans. Inf. Theor. 52 (4) (2006) 1289–1306.

[16] D. Ebrahimi, C. Assi, On the benefits of network coding to compressive data gathering in wireless sensor networks, in: Proceedings of IEEE SECON, 2015, pp. 55–63, doi:10.1109/SAHCN.2015.7338291.

[17] D. Ebrahimi, C. Assi, On the interaction between scheduling and compressive data gathering in wireless sensor networks, IEEE Trans. Wireless Commun. 15 (4) (2016) 2845–2858, doi:10.1109/TWC.2015.2512272.

[18] Y. Fan, Y. Jiang, H. Zhu, An efficient privacy-preserving scheme against traffic analysis attacks in network coding, in: Proceedings of IEEE INFOCOM, 2009, pp. 2213–2221.

[19] N. Fips, Announcing the advanced encryption standard (aes), Nat. Ins. Standards Technol. (NIST) (2001) 29 (8) (2001) 2200–2203.

[20] Z. Fu, K. Ren, J. Shu, X. Sun, F. Huang, Enabling personalized search over encrypted outsourced data with efficiency improvement, IEEE Trans. Parallel Distrib. Syst. 10.1109/TPDS.2015.2506573.

[21] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. 98 (1) (2015) 190–200.

[22] Z. Fu, X. Wu, C. Guan, X. Sun, K. Ren, Towards efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement, IEEE Trans. Inf. Foren. Secur. 10.1109/TIFS.2016.2596138.

[23] P. Hu, K. Xing, X. Cheng, Information leaks out: attacks and countermeasures on compressive data gathering in wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2014, pp. 1258–1266.

[24] M.A. Iwen, Simple deterministically constructible rip matrices with sublinear fourier sampling requirements, in: 43rd Annual Conference on Information Sciences and Systems, CISS, 2009, pp. 870–875.

[25] L. Kong, L. He, X.Y. Liu, Y. Gu, M.Y. Wu, X. Liu, Privacy-preserving compressive sensing for crowdsensing based trajectory recovery, in: Proceedings of IEEE ICDCS, 2015, pp. 31–40, doi:10.1109/ICDCS.2015.12.

[26] Y. Li, K. Xie, X. Wang, Pushing towards the limit of sampling rate: adaptive chasing sampling, in: Proceedings of IEEE MASS, 2015, pp. 398–406.

[27] X. Lin, R. Lu, H. Zhu, Asrpake: An anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks., in: Proceedings of IEEE ICC, 2007, pp. 1247–1253.

[28] X. Liu, Q. Liu, T. Peng, J. Wu, Dynamic access policy in cloud-based personal health record (phr) systems, Inf. Sci. (Ny) (2016), doi:10.1016/j.ins.2016.06.035.

[29] X. Liu, S. Zhang, K. Bu, A locality-based range-free localization algorithm for anisotropic wireless sensor networks, Telecommun. Syst. 62 (1) (2016) 3–13.

[30] J. Luo, J. Hu, D. Wu, R. Li, Opportunistic routing algorithm for relay node selection in wireless sensor networks, IEEE Trans. Ind. Inf. 11 (2015) 112–121.

[31] S.J. Luo Chong Wu Feng, Compressive data gathering for large-scale wireless sensor networks, in: Proceedings of ACM MobiCom, 2009, pp. 145–156.

[32] T. Ma, J. Zhou, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, S. Lee, Social network and tag sources based augmenting collaborative recommender system, IEICE Trans. Inf. Syst 98 (4) (2015) 902–910.

[33] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.

[34] T. Okamoto, S. Uchiyama, A new public-key cryptosystem as secure as factoring, in: Advances in Cryptology-EUROCRYPT 98, 1998, pp. 308–318.

[35] A. Orsdemir, H.O. Altun, G. Sharma, M.F. Bocko, On the security and robustness of encryption via compressed sensing, in: Proceedings of IEEE MILCOM, 2008, pp. 1–7.

[36] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology-EUROCRYPT 99, 1999, pp. 223–238.

[37] Y. Rachlin, D. Baron, The secrecy of compressed sensing measurements, in: Proceedings of IEEE 46th Allerton Conference on Communication, Control, and Computing, 2008, pp. 813–817.

[38] J.-F. Raymond, Traffic analysis: Protocols, attacks, design issues, and open problems, in: Designing Privacy Enhancing Technologies, 2001, pp. 10–29.

[39] Y. Ren, J. Shen, J. Wang, J. Han, S. Lee, Mutual verifiable provable data auditing in public cloud storage, J. Internet Technol. 16 (2) (2015) 317–323.

[40] J. Shen, H. Tan, J. Wang, J. Wang, S. Lee, A novel routing protocol providing good transmission reliability in underwater sensor networks, J. Internet Technol. 16 (1) (2015) 171–178.

[41] Y. Tang, B. Zhang, T. Jing, Robust compressive data gathering in wireless sensor networks, IEEE Trans. Wireless Commun. 12 (6) (2013) 2754–2761.

[42] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theor. 53 (12) (2007) 4655–4666.

[43] G. Tsouri, R. Dautov, Establishing secure measurement matrix for compressed sensing using wireless physical layer security, in: Proceedings of International Conference on Computing, Networking and Communications (ICNC), 2013, pp. 354–358.

[44] J.P. Vilela, L. Lima, J. Barros, Lightweight security for network coding, in: Proceedings of IEEE ICC, 2008, pp. 1750–1754.

[45] J. Wang, S. Tang, B. Yin, Data gathering in wireless sensor networks through intelligent compressive sensing, in: Proceedings of IEEE INFOCOM, 2012, pp. 603–611.

[46] D. Wu, D.I. Arkhipov, M. Kim, C.L. Talcott, A.C. Regan, J.A. McCann, N. Venkatasubramanian, Addsen: adaptive data processing and dissemination for drone swarms in urban sensing, IEEE Trans. Comput. (2016), doi:10.1109/TC.2016.2584061.

[47] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, K. Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing, IEEE Trans. Inf. Forensics Secur. (2016), doi:10.1109/TIFS.2016.2590944.

[48] Z. Xia, N.N. Xiong, A.V. Vasilakos, X. Sun, Epcbir: an efficient and privacy-preserving content-based image retrieval scheme in cloud computing, Inf. Sci. (Ny) (2016).

[49] L. Xiang, J. Luo, C. Rosenberg, Compressed data aggregation: energy-efficient and high-fidelity data collection, IEEE/ACM Trans. Netw. 21 (6) (2013) 1722–1735.

[50] L. Xiang, J. Luo, A. Vasilakos, Compressed data aggregation for energy efficient wireless sensor networks, in: Proceedings of IEEE SECON, 2011, pp. 46–54.

[51] K. Xie, W. Luo, X. Wang, L. Wang, J. Wen, Road condition gathering with vehicular dtn, in: Proceedings of IEEE INFOCOM Workshops, 2015, pp. 576–581.

[52] K. Xie, W. Luo, X. Wang, D. Xie, J. Cao, J. Wen, G. Xie, Decentralized context sharing in vehicular delay tolerant networks with compressive sensing, in: Proceedings of IEEE ICDCS, 2016, pp. 169–178.

[53] K. Xie, L. Wang, X. Wang, J. Wen, G. Xie, Learning from the past: intelligent on-line weather monitoring based on matrix completion, in: Proceedings of IEEE ICDCS, IEEE, 2014, pp. 176–185.

[54] K. Xie, X. Wang, X. Liu, J. Wen, J. Cao, Interference-aware cooperative communication in multi-radio multi-channel wireless networks, IEEE Trans. Comput. 65 (5) (2016) 1528–1542.

[55] K. Xie, X. Wang, J. Wen, J. Cao, Cooperative routing with relay assignment in multi-radio multihop wireless networks, IEEE/ACM Trans. Netw. (TON) 24 (2) (2016) 859–872.

[56] S. Xie, Y. Wang, Construction of tree network with limited delivery latency in homogeneous wireless sensor networks, Wireless Personal Commun. 78 (1) (2014) 231–246.

[57] K. Xing, X. Cheng, F. Liu, Location-centric storage for safety warning based on roadway sensor networks, J. Parallel Distrib. Comput. 67 (3) (2007) 336–345.

[58] L. Zhang, S. Xiao, N. Cai, J. Du, An opportunistic network coding algorithm based on the queue state and network topology, in: Proceeding of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2011, pp. 290–293.

[59] S. Zhang, X. Liu, J. Wang, J. Cao, G. Min, Accurate range-free localization for anisotropic wireless sensor networks, ACM Trans. Sensor Netw. (TOSN) 11 (3) (2015) 51.

[60] Y. Zhang, X. Sun, W. Baowei, Efficient algorithm for k-barrier coverage based on integer linear programming, China Commun. 13 (7) (2016) 16–23.

[61] C. Zhao, W. Zhang, X. Yang, A novel compressive sensing based data aggregation scheme for wireless sensor networks, in: Proceedings of IEEE ICC, 2014, pp. 18–23, doi:10.1109/ICC.2014.6883288.