Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 962925, 11 pages
http://dx.doi.org/10.1155/2014/962925

Research Article

An Efficient Secure Data Aggregation Based on Homomorphic Primitives in Wireless Sensor Networks

Qiang Zhou,[1,2,3] Geng Yang,[1,2] and Liwen He[1]

[1] *School of Computer Science & Technology, Nanjing University of Posts & Telecommunications, Nanjing 210046, China*
[2] *Key Laboratory of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts & Telecommunications, Nanjing 210046, China*
[3] *School of Computer & Information Engineering, Chuzhou University, Chuzhou 239000, China*

Correspondence should be addressed to Liwen He; helw@njupt.edu.cn

Received 17 September 2013; Revised 22 November 2013; Accepted 7 December 2013; Published 6 January 2014

Academic Editor: Yun Liu

Copyright © 2014 Qiang Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data aggregation is an important method to reduce the energy consumption in wireless sensor networks (WSNs); however, it suffers from the security problems of data privacy and integrity. Existing solutions either have large communication and computation overheads or only produce inaccurate results. This paper proposes a novel secure data aggregation scheme based on homomorphic primitives in WSNs (abbreviated as SDA-HP). The scheme adopts a symmetric-key homomorphic encryption to protect data privacy and combines it with homomorphic MAC synchronically to check the aggregation data integrity. It compares the scheme with the previously known methods such as SIES, iPDA, and iCPDA in terms of the data privacy protection efficiency, integrity performance, computation overhead, communication overhead, and data aggregation accuracy. Simulation results and performance analysis show that our SDA-HP requires less communication and computation overheads than previously known methods and can effectively preserve data privacy, check data integrity, and achieve high data transmission efficiency and accurate data aggregation rate while consuming less energy to prolong network lifetime. To the best of our knowledge, this is the first work that provides both integrity and privacy based on homomorphic primitives.

1. Introduction

Currently, wireless sensor networks (WSNs) have many popular applications, such as real-time accident reporting, environment monitoring, and military investigation. In WSNs, sensors are deployed to gather different kinds of data within a certain range and send them to the base station (BS). Sensors are restricted by energy consumption due to battery supply and computational capacity; therefore, energy saving technologies must be considered. Data aggregation [1] is one of the important approaches to facilitate the utilization of WSNs. However, WSNs are often deployed in an open and hostile environment; the inherent characteristics of WSNs and data aggregation algorithms make WSNs data aggregation face many security and performance challenges, such as data integrity; privacy protection, and how to enhance the security and performance becomes the key issues for practical applications.

In recent years, some schemes [2–5] have been proposed focusing on guaranteeing the data privacy during data aggregation phase, but these do not protect the integrity of aggregation data sent to the BS. A compromised aggregator may arbitrarily forge aggregation data and let the BS accept them. Other schemes [6–8] are proposed to guarantee the data integrity, but these lead to the leakage of data privacy due to decryption at aggregator. In this paper, we aim to bridge this gap in secure data aggregation and focus on preserving data privacy and integrity simultaneously through data aggregation phase in WSNs.

Many innovative secure data aggregation schemes have been proposed, and a survey of these works is presented in the literature [1]. These solutions fall into two main categories: hop-by-hop and end-to-end secure data aggregation. The hop-by-hop schemes are vulnerable to attacks because the data will be decrypted on aggregators and often have to enhance the security by using the expensive encryption and decryption algorithms; thus, the communication and computation overheads are large. The end-to-end schemes seem more secure; the data are transparent to the aggregators which can aggregate and transfer the encrypted data without decrypting them, and the end-to-end privacy is achieved by using homomorphic encryption (HE). HE allows the ciphertext to be aggregated directly, then the decrypted aggregation result matches the result of aggregation operations performed on plaintext. HE has been widely used for data aggregation in WSNs [9, 10]. However, the existing HE schemes suffer from the data integrity issue.

Since the incoming packets have been aggregated by aggregators, data integrity cannot be checked by the conventional MACs method. Therefore, we propose a novel integrity protection scheme for data aggregation based on homomorphic MAC (HM). HM is an energy-efficient symmetric approach, and it is first designed to check the integrity of network coded data [11]. However, it cannot be used directly in WSNs due to its complicated set of parameters and steps. Therefore, we remove an unnecessary step and some parameters in the original scheme and then adapt it to protect the integrity of data aggregation for energy constraint WSNs, and combine it with the symmetric-key HE [12] to protect data privacy. Then, we originally propose a novel secure data aggregation scheme based on homomorphic primitives (SDA-HP). This scheme relies on the combination of a revised version of the HM and the new symmetric-key HE [12] synchronically for WSNs. To the best of our knowledge, our proposed method is the first work that addresses data aggregation supporting both integrity and confidentiality based on homomorphic primitives. The proposed scheme can significantly improve the transmission efficiency and energy efficiency, shorten the communication delay, and achieve accurate data aggregation.

The rest of this paper is organized as follows. Section 2 presents the existing approaches to privacy protection and integrity in WSNs. Section 3 discusses the background and assumptions about the problem we are trying to solve. Section 4 presents a new secure data aggregation scheme based on homomorphic primitives. Section 5 analyzes the security performance and experimental results to prove the effectiveness and efficiency of our scheme. Section 6 gives conclusions and some future work.

2. Related Work

To solve both integrity and privacy issues for data aggregation phase in WSNs, Ozdemir and Çam [13] propose an authentication protocol to integrate false data detection with data aggregation and confidentiality; the monitoring nodes of every aggregator also perform data aggregation and compute the MACs for data verification; then, the sensors between two consecutive aggregators verify the integrity of the encrypted data. However, it has some flaws: (i) topological constraints that require at least T nodes on the path between any two consecutive data aggregators, (ii) nonneighboring sensors which need to spend longer time to establish pairwise keys, which may incur attacks for adversary, and (iii) compromised nodes which are likely to obtain group key through group key establishment process, which causes the data privacy leakage.

He et al. present two new data aggregation schemes, named iPDA [14] and iCPDA [15], which piggyback on SMART [2] and CPDA [2] schemes respectively. iPDA achieves privacy protection through data slicing and assembling technique as SMART and achieves integrity through redundancy by constructing disjoint aggregation trees. In iCPDA protocol, cluster members can detect data pollution attacks through monitoring the cluster leaders, so iCPDA spends a little more message overhead to achieve data integrity. However, both schemes need much more communication and computation overheads.

Chan et al. [16] present the first provably secure hierarchical data aggregation scheme based on aggregation-commit-verify approach, which forces the adversary to commit to its choice of aggregation results and then allows the sensors to verify whether their aggregation contributions are correct or not. Although this secure aggregation scheme can be used for arbitrary topologies and multiple malicious nodes, the communication and computation overheads are still very large. Then, Frikken and Dougherty [17] improve Chan's scheme by reducing the maximum communication per node from $O(\Delta \log^2 n)$ to $O(\Delta \log n)$, where Δ is the maximum degree of the aggregation tree and n is the number of nodes in WSNs.

In order to mitigate the drawbacks of the hop-by-hop schemes, some end-to-end protocols are proposed. Chen et al. [18] propose a recoverable concealed data aggregation scheme for data integrity in WSNs, named RCDA. It integrates the aggregate signature scheme to ensure data integrity and authenticity and can recover all sensing data even these data which have been aggregated; however, it is not practical for large scale network deployment due to its high costs. Castelluccia et al. [19] present a simple and provably secure scheme based on an extension of the one-time pad encryption technique. This scheme allows efficient additive aggregation of encrypted data, and only one modular addition over modulo n is necessary for aggregation. The privacy and integrity of the scheme are based on the indistinguishability of a pseudorandom function, but its aggregation authentication scheme is only against outsider attacks. Papadopoulos et al. [20] present a scheme, named SIES, that provides both integrity and confidentiality through combination of homomorphic encryption and secret sharing. It can cover numerous aggregates and return exact results. Although this scheme only introduces a small amount of bandwidth consumption, the data transmission efficiency is low due to the oversize space of secret keys.

Garofalakis et al. [21] and Roy et al. [22] propose some lightweight secure data aggregation schemes which return approximate aggregation results securely based on synopsis diffusion approach. It can tolerate the malicious activities in the compromised nodes that contribute false subaggregate values. Though secure approximate aggregation is an interesting domain, it is orthogonal to our work.

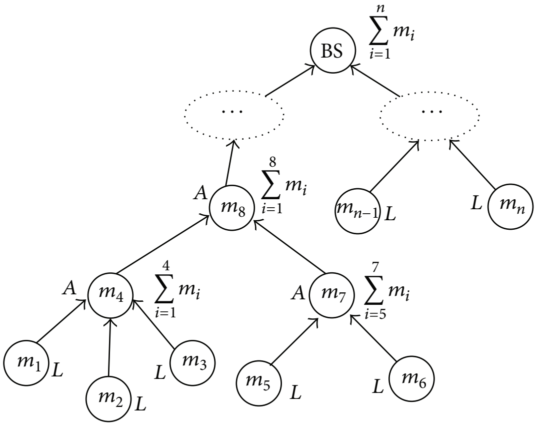


Figure 1: Aggregation tree.

3. Background and Assumptions

Section 3.1 describes in-network aggregation techniques. Section 3.2 provides some useful homomorphic primitive tools.

3.1. In-Network Aggregation

3.1.1. System Architecture. We assume a large number of sensors to form a query-based WSN, where a multihop routing protocol can be applied. Sensors will be organized as a tree topology where BS locates at the root, as shown in Figure 1. Each sensor acts as either a leaf sensor (L) or an aggregator (A), or both. Aggregation tree is constructed by using the TAG [23] protocol, and the BS is assumed to broadcast an authenticated query before aggregation. Each node collects environmental readings and an aggregator performs sum aggregation due to resource constrains. Without loss of generality, we focus on additive aggregation. It is not a too restrictive assumption, in respect of that it serves as the base of many other statistics aggregations, such as count, average, and variance.

3.1.2. Attack Model. We assume there are some polynomially bounded adversary that can perform attacks to break the privacy and integrity of aggregation results.

In this paper, we focus on two types of attacks:

(i) eavesdropping attacks: overhearing the sensors transmission data over its neighboring wireless links, the privacy of system can be compromised;

(ii) stealthy attacks: it is an intelligent attack to disinform a sensor network in a manner that to escape from attack discovery; that is, the BS accepts false sensors aggregation data.

Our goal is to propose a secure data aggregation scheme, which is robust against eavesdropping and stealthy attacks, efficient in keeping the additional overhead as small as possible, and effective in achieving accurate aggregation results.

3.2. Homomorphic Primitive Tools

3.2.1. Homomorphic Encryption. HE allows direct addition and multiplication of ciphertexts. Let m_1 and m_2 be two plaintexts and let $\otimes$, $\times$ be the homomorphic operations on the ciphertexts and plaintexts, respectively; we have $\text{Enc}(m_1)\otimes\text{Enc}(m_2) = \text{Enc}(m_1\times m_2)$, where $\text{Enc}(m)$ represents the ciphertext of m. For example, original Rivest's scheme [24] is homomorphic, supposing that p and q are large primes and $n = pq$, where n is a public key, and (p, q) is a private key. The encryption function is $E_{(p,q)}(m) = (m \bmod p,\ m \bmod q)$. We can reduce the two components mod p and mod q through applying the Chinese remainder theorem (CRT) [25]. Component-wise multiplications and additions of ciphertexts result in the corresponding multiplications and additions of plaintexts.

If $E_{(p,q)}(m_1) = (x_1, y_1)$ and $E_{(p,q)}(m_2) = (x_2, y_2)$, then

$$\begin{aligned} E_{(p,q)}(m_1 + m_2) &= \text{Add}\left(E_{(p,q)}(m_1), E_{(p,q)}(m_2)\right) \\ &= \left(x_1 + x_2 \bmod n,\ y_1 + y_2 \bmod n\right) \\ E_{(p,q)}(m_1 m_2) &= \text{Multi}\left(E_{(p,q)}(m_1), E_{(p,q)}(m_2)\right) \\ &= \left(x_1 x_2 \bmod n,\ y_1 y_2 \bmod n\right). \end{aligned} \tag{1}$$

3.2.2. Homomorphic MAC. Homomorphic MAC [11] should satisfy the following properties.

(i) Homomorphism: given (message, tag) pairs (m_1, t_1) and (m_2, t_2), we can compute a valid tag $t_a = w_1 t_1 + w_2 t_2$ for an aggregated message $m_a = w_1 m_1 + w_2 m_2$, where w_1 and w_2 are weights of m_1 and m_2, respectively.

(ii) Security against the chosen message attack: it is infeasible for the adversary to create a valid tag for an aggregated message even under a chosen message attack.

The modified homomorphic MAC includes three polynomial-time algorithms.

(i) Sign algorithm: $t_i = \text{sign}(k, \text{rid}, m_i, \text{id}_i)$, where id_i and m_i are the id and raw message of node i, respectively, rid is the id of report, and k is key.

(ii) Aggregation algorithm: $t = \text{aggregate}((m_1, t_1, w_1), \ldots, (m_j, t_j, w_j))$, that is; we can compute a tag $t = \sum_{i=1}^{j} w_i t_i$ for the aggregated message $m = \sum_{i=1}^{j} w_i m_i$ in the absence of key.

(iii) Verify algorithm: $\text{verify}(k, \text{rid}, m, t)$; that is, BS can verify the integrity of aggregation result by using k, rid, and tag t.

Definition 1. Let $\mathcal{T}$ be a homomorphic MAC scheme and let $\text{Adv}(\mathcal{A}, \mathcal{T})$ be the probability that adversaries $\mathcal{A}$ wins the security game [11]; if $\text{Adv}(\mathcal{A}, \mathcal{T})$ is negligible for all polynomial time adversaries $\mathcal{A}$, then the scheme $\mathcal{T}$ is secure.

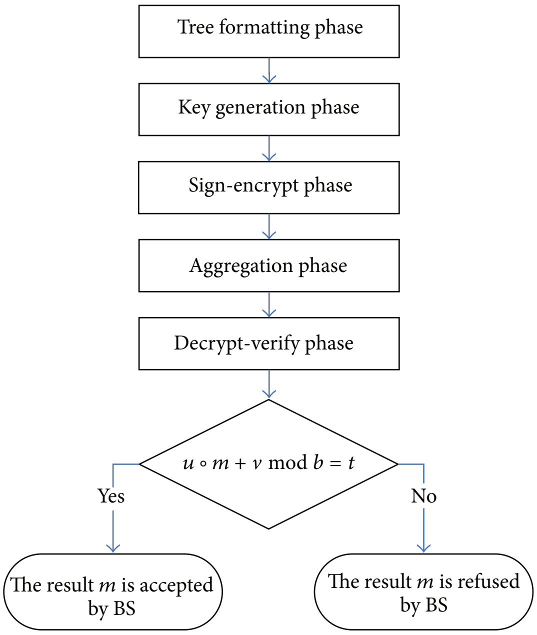


Figure 2: Sequence flow diagram of SDA-HP.

4. Secure Data Aggregation Based on Homomorphic Primitives

This section describes the details of SDA-HP scheme. The sequence flow diagram of SDA-HP consists of tree formatting phase, key generation phase, sign-encrypt phase, aggregation phase, and decrypt-verify phase, as shown in Figure 2.

4.1. Tree Formatting Phase. An aggregation tree is constructed according to the standard aggregation protocol TAG [23]; the formation method of the aggregation tree is illustrated as follows.

Step 1. BS is appointed to be the root, which broadcasts a message requesting sensors to generate an aggregation tree. In that message, it contains its own ID and its level information L_v (e.g., zero).

Step 2. When receiving the message from BS, any sensor not in the aggregation tree should assign its own level L_v to be the level in the message plus one L_v+1 and select the sender node as its parent, through which the sensor will route messages to the root.

Step 3. Each node of the aggregation tree rebroadcasts the message containing its own ID and level. When it receives the message, if any node has already been in the tree, it will reject the message, otherwise, the node also assigns its level L_v to be the level in the message plus one $L_v + 1$. When all nodes have been covered and their level and parent information are generated, an aggregation tree is constructed.

Step 4. During the aggregation phase, each sensor listens for messages from its children and then computes a partial state record which is the aggregate of children values with its own sensor readings. Eventually, the partial state record is transmitted upward along the tree, until the complete aggregated result converges at the root.

4.2. Key Generation Phase. This scheme uses $2d + 2$ private keys, which are denoted by $K = (p, q, r_1, \ldots, r_i, \ldots, r_d, s_1, \ldots, s_i, \ldots, s_d)$, where p and q are large primes, $1 \le i \le d$, and uses one public key, which is $n = pq$. We can preload these keys into sensors so that they can work correctly after being spread out, or the sensors can load the keys shared with BS.

4.3. Sign-Encrypt Phase. The encryption function is of the form:

$$E_K(\cdot) : \mathbb{Z}_n \longrightarrow (\mathbb{Z}_n \times \mathbb{Z}_n)^d. \tag{2}$$

To formally present our scheme, message m_i is formed as d arbitrary numbers of l bits. Let $b = 2^l$; then the message space is $\mathbb{F}_b^d$. In other words, message m_i can be represented as a vector of d numbers $(m_{i1}, m_{i2}, \ldots, m_{id})$, where $m_i = \sum_{j=1}^{d} m_{ij} \bmod n$ and $m_{ij} \in \mathbb{F}_b$.

To generate and verify tags, there is one global MAC key that consists of (k_1, k_2), which is shared between contributors and verifiers. Let $\mathscr{K}_1$ and $\mathscr{K}_2$ denote the key spaces of k_1 and k_2, respectively, let $\mathscr{I}$ denote the space of node identities, and let $\mathscr{R}$ denote the space of report identifies. We implement the pseudorandom generator $G : \mathscr{K}_1 \rightarrow \mathbb{F}_b^d$ and the pseudorandom function $F : (\mathscr{K}_2 \times \mathscr{I} \times \mathscr{R}) \rightarrow \mathbb{F}_b$ using AES [26]. The details of this phase are given as follows.

Sign-encrypt $(k, \mathrm{rid}, m_i, \mathrm{id}_i, r_i, s_i, p, q)$, by a contributor node i.

(1) $u = G(k_1) \in \mathbb{F}_b^d$.

(2) $v_i = F(k_2, \mathrm{id}_i, \mathrm{rid}) \in \mathbb{F}_b$.

(3) $t_i = u \circ m_i + v_i \in \mathbb{F}_b$//, where $\circ$ stands for the inner product of vectors u and m_i over finite field $\mathbb{F}_b$.

(4) Randomly generate $r_i < p$ and $s_i < q$, $\forall i \in [1, d]$ which are secret.

(5) Consider

$$\begin{aligned} E_K(m_i) = {} & ((m_{i1} r_1 \bmod p,\ m_{i1} s_1 \bmod q), \\ & (m_{i2} r_2 \bmod p,\ m_{i2} s_2 \bmod q), \ldots, \\ & (m_{id} r_d \bmod p,\ m_{id} s_d \bmod q)) \\ = {} & ((x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{id}, y_{id})) \in \mathbb{F}_b^{2d}. \end{aligned} \tag{3}$$

4.4. Aggregation Phase. Aggregate $((E_K(m_1), t_1, w_1), \ldots, (E_K(m_i), t_i, w_i))$, by an aggregator.

(1) $y = \sum_{i=1}^{j} w_i E_K(m_i) \bmod n \in \mathbb{F}_b^{2d}$//, where the additive operation is over $\mathbb{F}_b$ and w_i is weight of m_i.

(2) $t = \sum_{i=1}^{j} w_i t_i \in \mathbb{F}_b$.

4.5. Decrypt-Verify Phase. Decrypt-verify $(k, \mathrm{rid}, m_i, t_i)$, by the BS with the knowledge of contributor identities and weights.

(1) Suppose $y = \sum_{i=1}^{j} w_i E_K(m_i) \bmod n = ((x_1, y_1), (x_2, y_2), \ldots, (x_d, y_d))$,

$$\begin{aligned} y' &= \left(\left(x_1 r_1^{-1} \bmod p,\ y_1 s_1^{-1} \bmod q\right),\right. \\ &\quad \left(x_2 r_2^{-1} \bmod p,\ y_2 s_2^{-1} \bmod q\right), \ldots, \\ &\quad \left.\left(x_d r_d^{-1} \bmod p,\ y_d s_d^{-1} \bmod q\right)\right) \\ &= \left(\left(x_1', y_1'\right), \left(x_2', y_2'\right), \ldots, \left(x_d', y_d'\right)\right). \end{aligned} \tag{4}$$

//Multiply each component with the corresponding r_i^{-1} and s_i^{-1} in mod p and mod q, respectively.

(2) Use CRT to find $m = (m_1, m_2, \ldots, m_d (\bmod n))$

$$\begin{aligned} m_1 (\bmod n) &= x_1' q q^{-1} + y_1' p p^{-1} (\bmod n) \\ m_2 (\bmod n) &= x_2' q q^{-1} + y_2' p p^{-1} (\bmod n) \\ &\vdots \\ m_d (\bmod n) &= x_d' q q^{-1} + y_d' p p^{-1} (\bmod n). \end{aligned} \tag{5}$$

(3) $m = \sum_1^d m_i$.

(4) $u = F(k_1) \in \mathbb{F}_b^d$.

(5) $v = \sum_{i=1}^{j} [w_i \cdot F(k_2, \mathrm{id}_i, \mathrm{rid})] \in \mathbb{F}_b$.

(6) If $u \circ m + v \bmod b = t$, BS accepts the result m, else refuses the result m.

4.6. An Example of SDA-H. Let $p = 11, q = 13$; then, $n = 143$. For simplicity, let $d = 2$; that is, each plaintext message is split into 2 separated parts. Let $r_1 = 5, r_2 = 7, s_1 = 6, s_2 = 8$.

Suppose there are two plaintexts in $\mathbb{Z}_{143}$: $m_1 = 7$ and $m_2 = 10$; corresponding weights are $w_1 = 4$ and $w_2 = 3$, respectively. Suppose $u = (2, 4) \in \mathbb{F}_{64}^2$, $v_1 = 3$, $v_2 = 5 \in \mathbb{F}_{64}$, which are computed by the pseudorandom function using AES. The scheme runs as follows.

Step 1. Sign-encrypting m_1 and m_2.

Decompose m_1 into $m_{11} = 2$ and $m_{12} = 5$:

$$\begin{aligned} t_1 &= u \circ m_1 + v_1 = (2,4) \circ (2,5) + 3 = 27 \\ E_K(m_1) &= ((2 \times 5 \bmod 11,\ 2 \times 6 \bmod 13), \\ &\quad (5 \times 7 \bmod 13,\ 5 \times 8 \bmod 13)) \\ &= ((10, 12), (2, 1)). \end{aligned} \tag{6}$$

Decompose m_2 into $m_{21} = 3$ and $m_{22} = 7$:

$$\begin{aligned} t_2 &= u \circ m_2 + v_2 = (2,4) \circ (3,7) + 5 = 39 \\ E_K(m_2) &= ((2 \times 5 \bmod 11,\ 2 \times 6 \bmod 13), \\ &\quad (5 \times 7 \bmod 13,\ 5 \times 8 \bmod 13)) \\ &= ((10, 12), (2, 1)). \end{aligned} \tag{7}$$

Step 2. Aggregating $(E_K(m_1), t_1, w_1), \ldots, (E_K(m_i), t_i, w_i)$:

$$\begin{aligned} y &= (w_1 E(m_1) + w_2 E(m_2)) \\ &= ((4 \times 10 + 3 \times 4, 4 \times 12 + 3 \times 5), \\ &\quad (4 \times 2 + 3 \times 5, 4 \times 1 + 3 \times 4)) \bmod 143 \\ &= ((52, 63), (23, 16)), \\ t &= \sum_{i=1}^{j} w_i t_i = 4 \times 27 + 3 \times 39 \bmod 64 = 33. \end{aligned} \tag{8}$$

Step 3. Decrypt verifying:

$$\begin{aligned} & r_1^{-1} = 5^{-1} \equiv 9 \bmod 11, \qquad r_2^{-1} = 7^{-1} \equiv 8 \bmod 11 \\ & s_1^{-1} = 6^{-1} \equiv 11 \bmod 13, \qquad s_2^{-1} = 8^{-1} \equiv 5 \bmod 13 \\ & p^{-1} = 11^{-1} \equiv 6 \bmod 13, \qquad q^{-1} = 13^{-1} \equiv 6 \bmod 11 \\ y' &= ((52 \times 9 \bmod 11,\ 63 \times 11 \bmod 13), \\ &\quad (23 \times 8 \bmod 11,\ 16 \times 5 \bmod 13)) \\ &= ((6, 4), (8, 2)). \end{aligned} \tag{9}$$

Using CRT, the two components of the result m are

$$\begin{aligned} m_1 &= 6 \times 13 \times 6 + 4 \times 11 \times 6 (\bmod 143) = 17 \\ m_2 &= 8 \times 13 \times 6 + 2 \times 11 \times 6 (\bmod 143) = 41 \\ m &= m_1 + m_2 = 58. \end{aligned} \tag{10}$$

For computing homomorphic MAC, as we know from the above:

$$\begin{aligned} & u = (2, 4) \\ & v = (4 \times 3 + 3 \times 5) = 27 \\ & u \circ m + v = (2, 4) \circ (17, 41) + 27 \bmod 64 = 33 = t. \end{aligned} \tag{11}$$

Hence; the result m is accepted.

Actually, we may set $r_i = s_i = c, \forall i$, where c is a random number, to reduce the temporary key storage requirement of each sensor.

In SDA-HP, for the convenience of performance analysis, we let $d = 2$, $l = 32$. Considering the over flow during aggregation phase, we add the extra $\log_2 N$ bits in m_i. Figure 3 depicts the data format of $E_K(m_i)$ and the tag of m_i. The extra bits required cannot be more than $\log_2 N$ when N numbers are aggregated. Since the padding in m_i can be up to 2 bytes, our scheme can support up to $N = 2^{16}$ sources. We also can moderately expand the extra bits to meet the requirement.

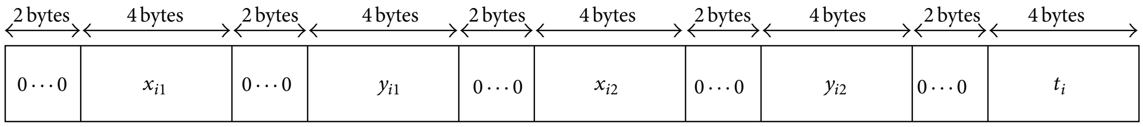


Figure 3: Data format of $E_K(m_i)$ and m_i' tag.

Table 1: Simulation parameters.

Radio parameters		Topology parameters	
White Gaussian noise	Noise floor	Number of nodes	Terrain dimensions
4 dB	−105 dB	600	400 meters × 400 meters

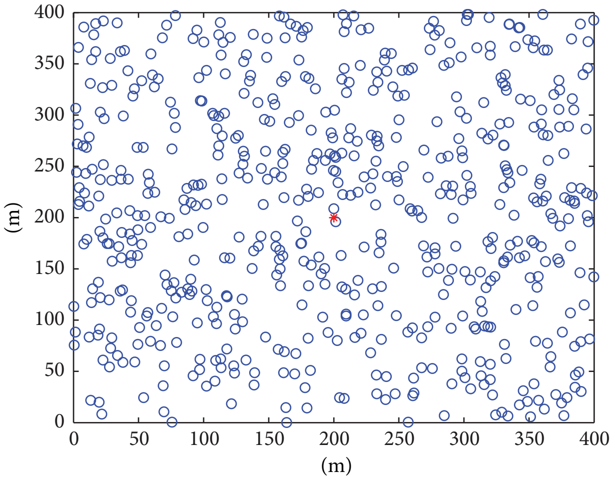


Figure 4: Nodes distribution.

5. Simulation and Analysis

In this section, we evaluate performances of SDA-HP scheme in terms of security properties, power analysis, and aggregation accuracy. The simulation is conducted in TOSSIM system operated by TinyOS 2.0. Table 1 shows the parameters setting in the simulation, and Figure 4 depicts the topology of nodes, where the BS coordinate is (200,200).

5.1. Privacy Protection. Secure to ciphertext-only attacks: SDA-HP is secure as long as factoring integer composite of large primes is difficult. In addition, test for encrypted zero in SDA-HP is unlikely according to the literature [12]. It is hard for an adversary to breach all encryption keys or find the plaintext m_i if it knows only the ciphertext $E_K(m)$.

We denote a as the probability that one key is broken and $p(a)$ as the probability that plaintext m_i is disclosed for a given a, and then take $p(a)$ as a confidentiality performance metric. When the message is broken down into d numbers, the number of keys is $2(d+1)$, so $p(a) = a^{2(d+1)}$. In the predistribution phase, a large key pool of K keys is generated and $2(d+1)$ keys are randomly selected from the key pool independently; therefore, the probability that an adversary breaches all keys can be approximated by $p(a) = (1/K)^{2(d+1)} = a^{2(d+1)}$. Figure 5 describes the system confidentiality performance under different values of d. Obviously, the larger the value of d, the better data confidentiality, can be achieved. However, a larger d will also require larger temporary storage space and computation and communication overheads, so there is a design tradeoff between the confidentiality protection, storage space, computation overhead, and communication overhead.

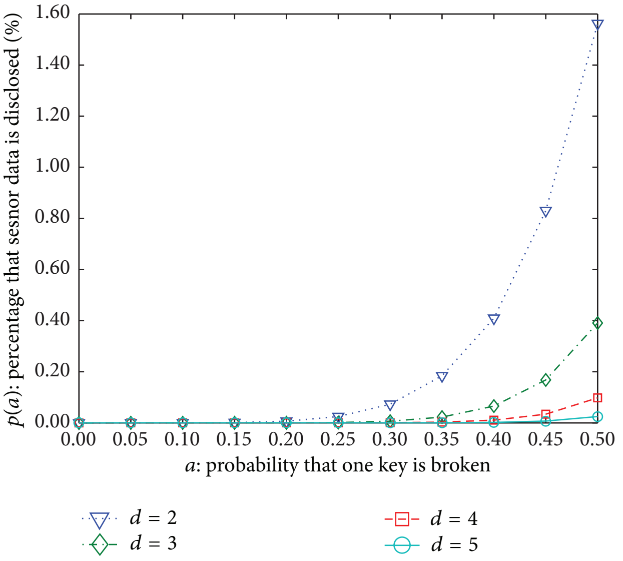


Figure 5: $p(a)$ under SDA-HP.

In the scheme, we proposed to limit d to a value in the range of 2–4 to balance the security protection and energy consumption.

The scheme SDA-HP uses homomorphic encryption to achieve end-to-end data confidentiality, so addition and scalar multiplication of the ciphertexts are just componentwise vector addition and scalar multiplication of the corresponding d-tuples. Even if the aggregation data is disclosed, the adversary can only get the aggregation result but not sensor data.

5.2. Integrity Performance

Definition 2. Assuming F and G are secure, $\mathscr{B}_1$ is a F adversary, and $\mathscr{B}_2$ is a G adversary. One defines $\mathrm{Adv}(\mathscr{B}_1, F)$ is $\mathscr{B}_1'$ advantage in winning the F security game, and $\mathrm{Adv}(\mathscr{B}_2, F)$ is $\mathscr{B}_2'$ advantage in winning the G security game.

Theorem 3. *Assuming F and G are secure, the homomorphic MAC scheme is secure for any fixed b and n. Also, for all* *homomorphic MAC scheme $\mathscr{T}$ adversaries $\mathscr{A}$, there are the F adversary $\mathscr{B}_1$ and the G adversary $\mathscr{B}_2$ which suffice for the inequation, which is $Adv(\mathscr{A}, \mathscr{T}) \leq Adv(\mathscr{B}_1, F) + Adv(\mathscr{B}_2, G) + (1/b)$.*

Proof. We use three games to prove the theorem. Supposing E_i is the event, the $\mathscr{A}$ wins the homomorphic MAC security game in Game i for $i = 1, 2, 3$.

Game 1 is the same to Attack Game 1 [11] applied to the scheme $\mathscr{T}$. So there is

$$\Pr(E_1) = \text{Adv}(\mathscr{A}, \mathscr{T}). \tag{12}$$

In Game 2, if we use a truly random string instead of the output of G in the scheme $\mathscr{T}$, that is, the challenger computers $u \xleftarrow{R} \mathbb{F}_b^d$ instead of $u \leftarrow G(k_1)$ in the process of Sign-encrypt, then Game 2 is the same as Game 1. So there is an G adversary $\mathscr{B}_2$ such that

$$\left|\Pr(E_1) - \Pr(E_2)\right| = \text{Adv}(\mathscr{B}_2, G). \tag{13}$$

In Game 3, if we use a truly random string instead of F in the scheme $\mathscr{T}$, that is, the challenger computers $v_i \xleftarrow{R} \mathbb{F}_q$ instead of $v_i \leftarrow F(k_2, \text{id}_i, \text{rid})$ in the process of Sign-encrypt, then the Game 3 is the same to the Game 2. So there is a F adversary $\mathscr{B}_1$ such that

$$\left|\Pr(E_2) - \Pr(E_3)\right| = \text{Adv}(\mathscr{B}_1, F). \tag{14}$$

The challenger in Game 3 works as follows:

$$\text{initialization}: u \xleftarrow{R} \mathbb{F}_b^d. \tag{15}$$

The adversary submits MAC queries (m_j, rid_i), and the challenger responds to rid_i as follows.

For $j = 1, 2, \ldots, n$ do

$$v_j \xleftarrow{R} \mathbb{F}_b, \quad T_j \longleftarrow \left(u \cdot m_j\right) + v_j \in \mathbb{F}_b \tag{16}$$

send $(T_1, T_2, \ldots, T_n)$ to $\mathscr{A}$.

Then, the adversary $\mathscr{A}$ outputs (rid^*, T^*, m^*). We first compute the v_j^* as follows.

If $\text{rid}^* = \text{rid}_i$, then;

set $(v_1^*, v_2^*, \ldots, v_n^*) \leftarrow (v_1, v_2, \ldots, v_n)$//forgery type 2;

else for $j = 1, 2, \ldots, n$ set $v_j^* \xleftarrow{R} \mathbb{F}_b$//forgery type 1.

Let $m^* = (m_1^*, m_2^*, \ldots, m_n^*)$. The adversary wins if $T^* = (u \cdot m^*) + \sum_{j=1}^{n} v_j^*$. For adversary type 2, $m^* \notin m_j$.

Let E^* be the event that the adversary outputs an adversary type 1. According to the literature [11], we can obtain

$$\begin{aligned}\Pr(E_3) &= \Pr(E_3 \wedge E^*) + \Pr(E_3 \wedge \neg E^*) \\ &= \frac{1}{b}\left(\Pr(E^*) + \Pr(\neg E^*)\right) = \frac{1}{b}.\end{aligned} \tag{17}$$

All the above equations prove the theorem. □

Table 2: Computation effort for SDA-HP.

	Computation overhead (at S)			Computation overhead (at A)		
d	+	×	%	+	×	%
2	3	6	5	8	0	4
3	5	9	7	12	0	6
4	7	12	9	16	0	8
5	9	15	11	20	0	10
8	15	24	17	32	0	16
10	19	28	19	40	0	20

5.3. Computation Overhead. We evaluate the computation overhead of SDA-HP for the Micaz Mote with an Atmega 128 CPU and compare it with the known algorithms such as SIES, iPDA and iCPDA. Both SDA-HP and SIES adopt symmetric-key homomorphic encryption, and both iPDA and iCPDA use simple hop-by-hop encryption with RC5. Therefore, the computational overhead can be determined by the following equation:

$$E_j = N_c\text{Cal} + N_e\text{Enc} + N_d\text{Dec}, \tag{18}$$

where Enc and Dec are the energy consumptions for one encryption and decryption process of 10 bits value, and Cal represents the energy consumption of one computational operation. N_c, N_e, and N_d denote the number of operations of computation, encryption, and decryption, respectively.

In SIES, it only needs some basic procedures, which involve three HMAC operations, one modular addition and one modular multiplication for every sensor.

In SDA-HP, we suggest that the keys and the parameters generation phase can be performed by the manufacturer before the WSN is deployed; furthermore, the pseudorandom generator is performed using AES regularly, so the energy consumption for the preconfiguration and pseudorandom function are not considered here. The computational overhead for an encryption and integrity protection at the sensor nodes depends on the choice of d. We illustrate the number of major operations with different d for every sensor in Table 2. Apparently, the computation and communication overheads increase with the increasing of d. Considering the security performance and the data overhead, we propose to use a moderate value of d, which is $d \leq 4$. SDA-HP retains a comparable performance to SIES, which is in the order of hundreds of milliseconds. As expected, the overhead of all values increases linearly with d.

However, iPDA and iCPDA involve generating an excessive number of sketches and performing some hop-by-hop RC5 encryptions and decryptions. iPDA requires in total six encryptions and decryptions and seven computational operations. iCPDA includes two encryptions and decryptions and seventeen computational operations for each leaf sensor.

To determine energy consumptions and computational time of encryption operations for each sensor, we use the cost functions for common operations listed by Groat et al. [27], which are given in Tables 3 and 4. Table 3 depicts the costs of the transmission and reception of 1 bit of data and computes for 1 CPU clock cycle. Table 4 shows time spent to encrypt and

Table 3: Energy consumption of common operations [27].

Operation	MICAz	TelosB
Computer per clock tick	3.5 nJ	1.2 nJ
Transmit 1 bit	0.6 μJ	0.72 μJ
Receive 1 bit	0.67 μJ	0.81 μJ

Table 4: Cost to encrypt/decrypt 10 bits of data [27].

Method, architecture	Time (ms)	Energy (μJ)
IDEA Enc, MICAz	2902.12	74.86
IDEA Dec, MICAz	8350.80	215.41
IDEA Enc, TelosB	2673.58	12.83
IDEA Dec, TelosB	7693.27	36.93
RC5 Enc, MICAz	7037.25	181.53
RC5 Dec, MICAz	7035.89	181.49
RC5 Enc, TelosB	6483.06	31.12
RC5 Dec, TelosB	6481.81	31.11
RC4, MICAz	2018.00	52.05
RC4, TelosB	1859.08	8.92

decrypt 10 bits of data on the TelosB and MICAz architectures with RC5, RC4, and IDEA.

In Figure 6, we assess the computational time for each leaf sensor. In order to control the relative error within 10%, we ran every experiment over 25 times and reported the average cost value. The computational time in SIES and SDA-HP is in the order of few milliseconds, which outperforms iPDA and iCPDA by approximately three orders of magnitude, since the computation-expensive RC5 encryptions and decryptions are performed in iPDA and iCPDA during sending or receiving data slice. Furthermore, the computational cost of SIES, iPDA and iCPDA is independent of d. Therefore, the comparison result for computation overhead is SIES < SDA-HP < iPDA (l = 2) < iCPDA (p_c = 0.3). Actually, since the SDA-HP needs to compute the tags of every message to check the data integrity, it has to spend more computation time than SIES. Although SIES also achieves data integrity by attaching the secret sharing in message, the secret sharing occupies 28 bytes space, and the data transmission efficiency is only $4/32 = 12.5\%$. However, as we can see from Figure 3, the primary sensor data of SDA-HP occupies 12 bytes, so the data transmission efficiency is $12/30 = 40\%$, which outperforms SIES by 27.5%.

5.4. Communication Overhead. We compare the communication overhead among SDA-HP, SIES, iPDA, and iCPDA. The above four schemes adopt TAG to construct the aggregation tree. The data packet size for SDA-HP, iPDA, and iCPDA is 30 bytes, and the packet size for SIES is 32 bytes. In order to make the comparison fairly, we evaluate the communication overhead through the sum of sending bytes of all sensors during one aggregation tree construction and aggregation phase for these four schemes. In order to reduce the relative error, we run every experiment over 25 times and record the average value. Figure 7 shows that the communication overhead for the four schemes is independent of epoch duration, and the comparison result is SDA-HP < SIES < iCPDA (p_c = 0.3) < iPDA (l = 2).

The theoretical analysis is as the following.

SDA-HP uses the homomorphic encryption to provide data confidentiality preserving and homomorphic MAC to check the aggregation data integrity. The number of data packets sent from every sensor is the same as the TAG, including one "hello" packet and one aggregation packet. Accordingly, the number of packets sent from all sensors is $O(2N)$, and the total bytes are 3.6×10^4, s.t. $N = 600$.

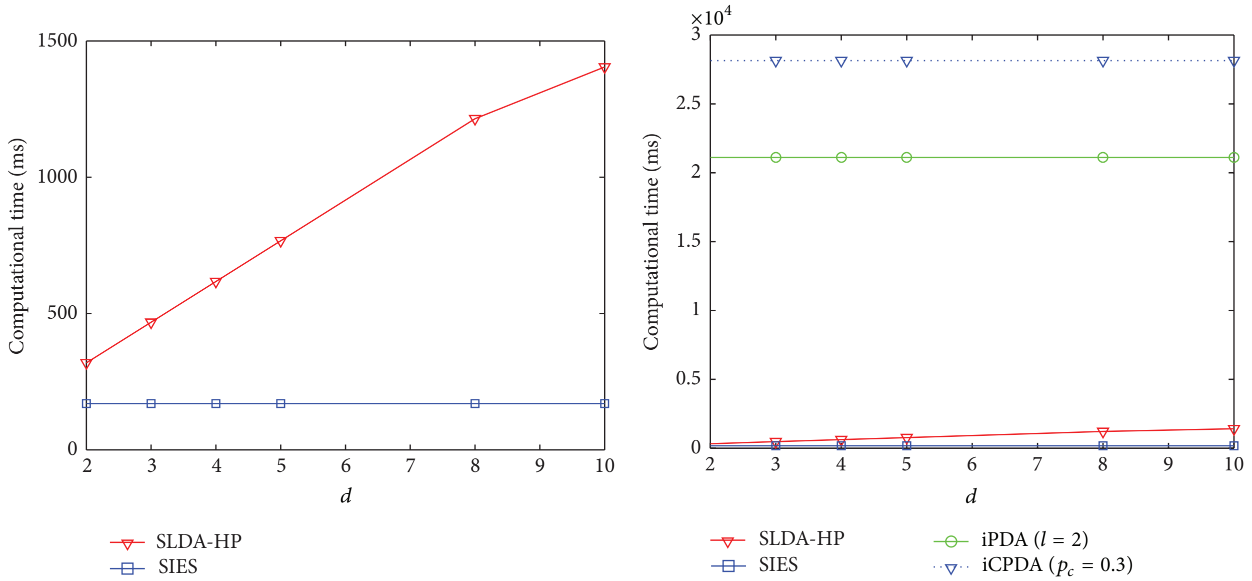
SIES adopts the homomorphic encryption to protect data confidentiality and attaches the secret sharing in message to provide data integrity. The number of data packets sent from every sensor is the same as SDA-HP. Hence the number of packets sent from all sensors is also $O(2N)$, and the total bytes are 3.8×10^4.

iPDA is built on data slicing techniques to achieve both data confidentiality and integrity. The number of data slicing (l) is at least 2 to ensure the data security, accordingly the number of packets sent from all sensors is $O((2l + 1)N)$, and the total bytes are 9×10^4.

iCPDA adopts secure multiparty computation method to protect data privacy and uses the sensor surveillance method to achieve integrity. Figure 8 shows the data packet interaction phase. Each member sensor needs to send three packets, and every cluster leader should send four packets, moreover, in order to check integrity, the cluster leaders also need to send the aggregation result of sublevel leaders and themselves to the neighbor monitors; hence, the number of packets sent from leaders amounts to six. Accordingly, the number of packets sent from all sensors is $O(3(1 - p_c)n + 6p_c n) = O((3 + 3p_c)n)$, where p_c is the probability of the sensors which choose themselves as leaders, and the total bytes are about 7×10^4.

5.5. Data Accuracy. In WSNs, message may be delayed and even dropped due to the processing time and collisions over wireless channels, so the aggregation accuracy is one of important performance metrics. Here, we define the accuracy metric as the ratio of the actual aggregation result collected by BS to the sum of the data sent by the all individual sensors.

Figure 9 shows the accuracy for SDA-HP, SIES, iPDA (l = 2) and iCPDA (p_c = 0.3). From the simulations, we can make three conclusions. Firstly, the accuracy of every scheme increases as the epoch duration increases. The reason is that the data packets sent within the large epoch duration will be easy to go through without collision. Secondly, the accuracy diagram for SDA-HP and SIES is almost the same, and their accuracy can reach 80% when epoch duration is about 20 seconds, which outperforms that of iPDA and iCPDA. This is due to the fact that is without the communication, overhead introduced by sending data slicing, so there will be less data collisions. Finally, the accuracy of iCPDA is slightly better than that of iPDA, because the packets sent by the iCPDA scheme are less than those of the iPDA scheme; therefore, the iCPDA scheme will have smaller data loss probability than the iPDA scheme.



(a) Comparison of computational time between SDA-HP and SIES

(b) Comparison of computational time among SDA-HP, SIES, iPDA and iCPDA

Figure 6: Computational time comparison.

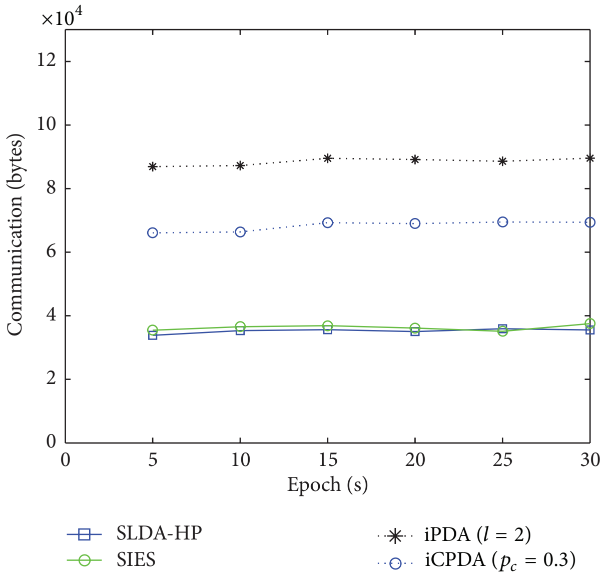


Figure 7: Communication overhead comparison.

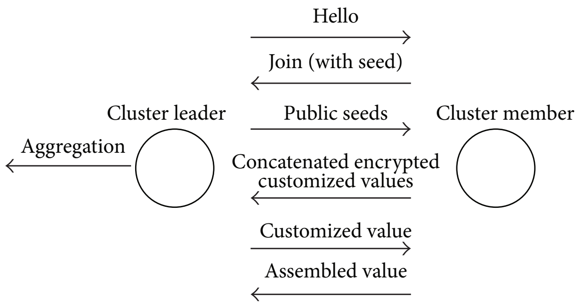


Figure 8: The message interaction in cluster of iCPDA.

6. Conclusion

Protecting data privacy and integrity during data aggregation at the same time is challenging in hierarchical WSNs. We originally present SDA-HP, a novel and efficient secure data aggregation scheme based on the homomorphic encryption and a revised version of the homomorphic MAC. Both techniques are lightweight, and they require very few energy consumptions. The proposed scheme can achieve high data transmission efficiency and accurate data aggregation. These features make SDA-HP very suitable to be applied for resource-constrained WSNs. Subsequently, we prove the security performance and give an example to illustrate how SDA-HP can be adopted to handle data aggregation, while protecting data privacy and integrity. Finally, simulations are used to compare our scheme with several known schemes in terms of the computation overhead, communication overhead, and accuracy. The numerical results show that SDA-HP is superior to other approaches in terms of security, complexity, and accuracy.

At present, SDA-HP is applied to the secure aggregation scheme for SUM queries only. Further research will be to design a secure data aggregation scheme which can cover a wide range of exact aggregate queries.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

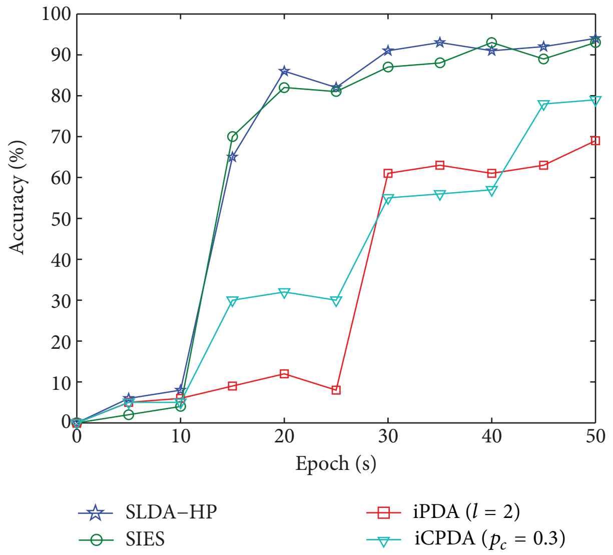


FIGURE 9: Accuracy comparison for SDA-HP, SIES, iPDA (l = 2) and iCPDA (p_c = 0.3).

Acknowledgments

This work is supported by the National Research Project of China under Grant no. 2012A138, National Basic Research Program (973 Program) of China under Grant nos. 2011CB302903, the National Natural Science Foundation of China under Grant nos. 61272084, 61302158, and 61300240, the Key Project of the Natural Science Research of Jiangsu University under Grant no. 11KJA520002, the Key Project of the Natural Science Research of Anhui University under Grant no. KJ2011ZD06, the Research Innovation Program for College Graduates of Jiangsu Province under Grant no. CXZZ12_0477, and the Natural Science Foundation of Chuzhou University under Grant no. 2012kj003Z.

References

[1] S. Ozdemir and Y. Xiao, "Secure data aggregation in wireless sensor networks: a comprehensive overview," *Computer Networks*, vol. 53, no. 12, pp. 2022–2037, 2009.

[2] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 2045–2053, Anchorage, Alaska, USA, May 2007.

[3] T. Jung, X. F. Mao, X. Y. Li et al., "Privacy-preserving data aggregation without secure channel: multivariate polynomial evaluation," in *Proceedings of the 32th IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 2734–2742, Turin, Italy, April 2013.

[4] K. Xing, Z. Wan, P. Hu et al., "Mutual privacy-preserving regression modeling in participatory sensing," in *Proceedings of the 32th IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 3039–3047, Turin, Italy, April 2013.

[5] G. Yang, S. Li, X. Xu, H. Dai, and Z. Yang, "Precision-enhanced and encryption-mixed privacy-preserving data aggregation in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 427275, 12 pages, 2013.

[6] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," *ACM Transactions on Information and System Security*, vol. 11, no. 4, article 18, 2008.

[7] K. B. Frikken, K. Kauffman, and A. Steele, "General secure sensor aggregation in the presence of malicious nodes," in *Proceedings of the 27th IEEE International Conference on Data Engineering (ICDE '11)*, pp. 506–516, Hannover, Germany, April 2011.

[8] 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, Article ID 256852, 9 pages, 2013.

[9] L. Wang, L. Wang, Y. Pan, Z. Zhang, and Y. Yang, "Discrete logarithm based additively homomorphic encryption and secure data aggregation," *Information Sciences*, vol. 181, no. 16, pp. 3308–3322, 2011.

[10] Y. Lin, S. Chang, and H. Sun, "CDAMA: concealed data aggregation scheme for multiple applications in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1471–1483, 2012.

[11] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in *Proceedings of the 7th International Conference on Applied Cryptography and Network Security (Springer ACNS '09)*, pp. 292–305, Paris, France, June 2009.

[12] A. C.-F. Chan, "Symmetric-key homomorphic encryption for encrypted data processing," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, Dresden, Germany, June 2009.

[13] S. Ozdemir and H. Çam, "Integration of false data detection with data aggregation and confidential transmission in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 736–749, 2010.

[14] W. He, H. Nguyen, X. Liu, K. Nahrstedt, and T. Abdelzaher, "iPDA: an integrity-protecting private data aggregation scheme for wireless sensor networks," in *Proceedings of the IEEE Military Communications Conference (IEEE MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.

[15] W. He, X. Liu, H. Nguyen et al., "A cluster-based protocol to enforce integrity and preserve privacy in data aggregation," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (IEEE ICDCS Workshops '09)*, pp. 14–19, Montreal, Canada, June 2009.

[16] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in *Proceedings of the13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 278–287, Alexandria, Va, USA, November 2006.

[17] K. B. Frikken and J. A. Dougherty IV, "An efficient integrity-preserving scheme for hierarchical sensor aggregation," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 68–76, Alexandria, Va, USA, April 2008.

[18] C.-M. Chen, Y.-H. Lin, Y.-C. Lin, and H.-M. Sun, "RCDA: recoverable concealed data aggregation for data integrity in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 727–734, 2012.

[19] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data

in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–36, 2009.

[20] S. Papadopoulos, A. Kiayias, and D. Papadias, "Exact In-Network aggregation with Integrity and Confidentiality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1760–1773, 2012.

[21] M. Garofalakis, J. M. Hellerstein, and P. Maniatis, "Proof sketches: verifiable in-network aggregation," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 996–1005, Istanbul, Turkey, April 2007.

[22] S. Roy, M. Conti, S. Setia et al., "Secure data aggregation in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1040–1052, 2012.

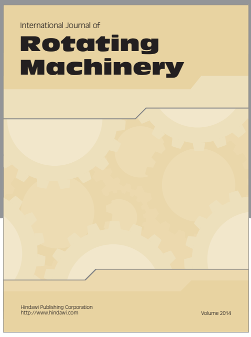
[23] S. Madden, M. J. Franklin, J. M. Hellerstein et al., "TAG: a tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the 5th Operating Systems Design and Implementation Symposium (ACM OSDI '02)*, pp. 131–146, Boston, Mass, USA, December 2002.

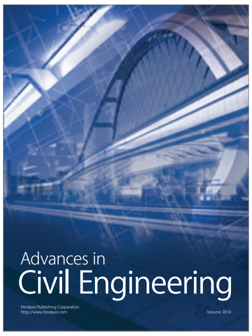
[24] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[25] C. H. Wu, J. H. Hong, and C. W. Wu, "RSA cryptosystem design based on the Chinese remainder theorem," in *Proceedings of the Asia and South Pacific Design Automation Conference (ACM ASP-DAC '01)*, pp. 391–395, Yokohama, Japan, January 2001.

[26] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer, New York, NY, USA, 2002.

[27] M. M. Groat, W. Hey, and S. Forrest, "KIPDA: K-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of The 30th IEEE Conference on Computer Communications (IEEE INFOCOM '11)*, pp. 2024–2032, Shanghai, China, April 2011.





Hindawi

Submit your manuscripts at
http://www.hindawi.com