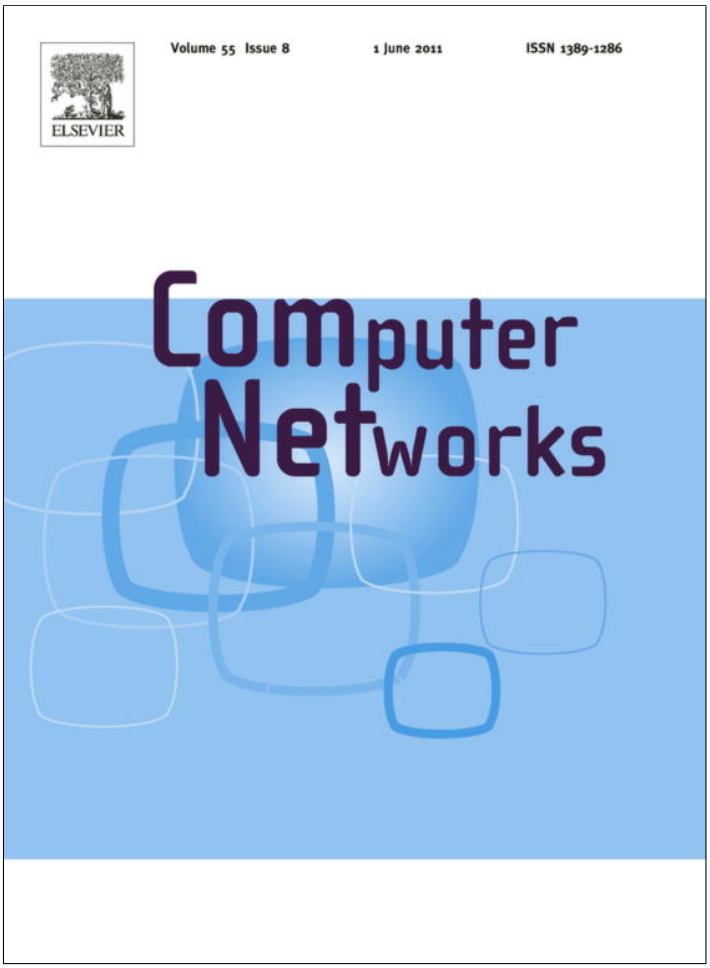
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Computer Networks 55 (2011) 1735–1746

Contents lists available at ScienceDirect

Computer Networks

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

Integrity protecting hierarchical concealed data aggregation for wireless sensor networks

Suat Ozdemir [a,*], Yang Xiao [b]

[a] Computer Engineering Department, Gazi University, Maltepe, Ankara, TR-06570, Turkey
[b] Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, United States

ARTICLE INFO

Article history:
Received 17 January 2010
Received in revised form 16 December 2010
Accepted 10 January 2011
Available online 19 January 2011
Responsible Editor: S. Sicari

Keywords:
Data integrity
Data confidentiality
Concealed data aggregation
Wireless sensor networks

ABSTRACT

In wireless sensor networks, performing data aggregation while preserving data confidentiality and integrity is challenging. Recently, privacy homomorphism-based secure data aggregation schemes have been proposed to seamlessly integrate confidentiality and data aggregation. However, these schemes do not provide data integrity or allow hierarchical data aggregation if more than one encryption key is used in the network. This paper presents a novel integrity protecting hierarchical concealed data aggregation protocol that allows the aggregation of data packets that are encrypted with different encryption keys. In addition, during the decryption of aggregated data, the base station is able to classify the encrypted and aggregated data based on the encryption keys. The proposed data aggregation scheme employs an elliptic curve cryptography-based homomorphic encryption algorithm to offer data integrity and confidentiality along with hierarchical data aggregation.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

A wireless sensor network is composed of a large number of sensor nodes that have strictly limited computation and communication abilities and power resources [1]. In the near future, these networks are predicted to be employed widely in security sensitive applications, including critical area surveillance, office automation, health monitoring, and military tracking. Hence, security is an essential issue in wireless sensor networks, and widespread deployment of these networks could be curtailed by the lack of adequate security [2]. Security in wireless sensor networks mostly involves confidentiality and integrity of the collected data. Confidentiality is usually achieved by preventing outsiders from understanding the transmitted data packets. Integrity means that the receiver node is guaranteed to notice if the received data packet is modified by other nodes. However, compared with conventional desktop computers, implementing security mechanisms that provide confidentiality and integrity is not easy in wireless sensor networks due to the limited processing power, storage, bandwidth, and sensor node energy [2].

In a wireless sensor network, sensor nodes are generally spread over the area to be monitored, where they self-organize into a multi-hop network. Each sensor node gathers data from its sensing region and sends this data to the base station over a multi-hop path. In such a network, high communication costs, limited battery power, and scarce bandwidth resources make it challenging to provide efficient solutions to the data gathering problem [3]. Data aggregation is an important primitive that aims to combine and summarize data packets from several sensor nodes so that communication bandwidth and energy consumption are reduced [4]. However, in terms of security, data aggregation is risky. A sensor node that is compromised by an adversary can either illegally disclose the data it collects from other nodes or report arbitrary values as its aggregation results. Therefore, an adversary can compromise both

* Corresponding author. Tel.: +90 312 582 3133; fax: +90 312 230 6503.
E-mail addresses: suatozdemir@gazi.edu.tr (S. Ozdemir), yangxiao@cs.ua.edu (Y. Xiao).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.01.006

Author's personal copy

the confidentiality and the integrity of the data of a large portion of the wireless sensor network by capturing a number of data aggregators that are positioned close to the base station.

Because both data aggregation and security are critical for wireless sensor networks, achieving secure data aggregation that protects integrity has been an attractive goal for researchers [5–16]. In existing secure data aggregation protocols, data aggregators generally decrypt every message they receive, aggregate the messages according to the corresponding aggregation function, and encrypt the aggregation result before forwarding it. Therefore, while these data aggregation protocols protect data integrity and improve the bandwidth and energy utilization of the network, they negatively affect other performance metrics, such as delay and data confidentiality. Recently, several data aggregation protocols [17–20] that provide data confidentiality without causing delay have been proposed. However, these protocols are not resilient against attacks targeting the data integrity because homomorphic encryption schemes are malleable by design. While data confidentiality is indeed important, preserving the integrity of the sensed data is a more critical goal. For example, if data integrity is not provided in a critical area surveillance application, an adversary can deceive the base station by corrupting the collected data. Therefore, it is crucial to design secure data aggregation schemes that provide both data confidentiality and integrity.

In this paper, we propose an Integrity Protecting Hierarchical Concealed Data Aggregation (IPHCDA) protocol. This novel protocol allows hierarchical aggregation of encrypted sensor data while providing confidentiality and integrity. Moreover, aggregated data can be classified at the base station based on the encryption keys. The proposed protocol employs a privacy homomorphic encryption scheme [21] and message authentication codes (MAC) to achieve hierarchical data aggregation that preserves data confidentiality and integrity. In the network deployment phase, IPHCDA protocol virtually partitions the network into several regions and employs a different public key in each region. The data collected in a region is encrypted using the public key of the region, and the MAC of the aggregated data is computed. The encrypted data of several regions can be hierarchically aggregated into a single piece of data without violating data confidentiality. To preserve data integrity during hierarchical aggregation, the MAC of each region is combined using the XOR function, resulting in a single MAC that is verified by the base station. During the decryption of the aggregated data, the base station is able to classify the aggregated data based on the encryption keys and verify the MAC of the aggregated data, thereby achieving data integrity. This method is particularly useful when the base station needs to analyze the data from a certain region in the network. For example, in a battlefield surveillance application, the base station may need to analyze data from a certain part of the battlefield. In this case, IPHCDA is able to serve this specific information to the base station without violating the data confidentiality, integrity, or energy efficiency requirements of the application. Fig. 1 presents the battlefield surveillance example. The figure also summarizes the motivation behind the IPHCDA protocol. The security analysis shows that the IPHCDA scheme is resilient against general security attacks. In addition, the performance evaluation results indicate that IPHCDA is feasible for resource-constrained sensor nodes.

Our contribution in this paper is to provide a concealed data aggregation technique that protects integrity and allows hierarchical aggregation of data encrypted with different keys. To the best of our knowledge, this work is the first to propose an integrity-protecting concealed data aggregation scheme for a multi-data aggregator model.

The rest of the paper is organized as follows. In Section 2, the current state of secure data aggregation is described. Section 3 explains the system model and preliminaries and offers a network deployment scenario. The IPHCDA protocol is outlined in Section 4. Security anal-

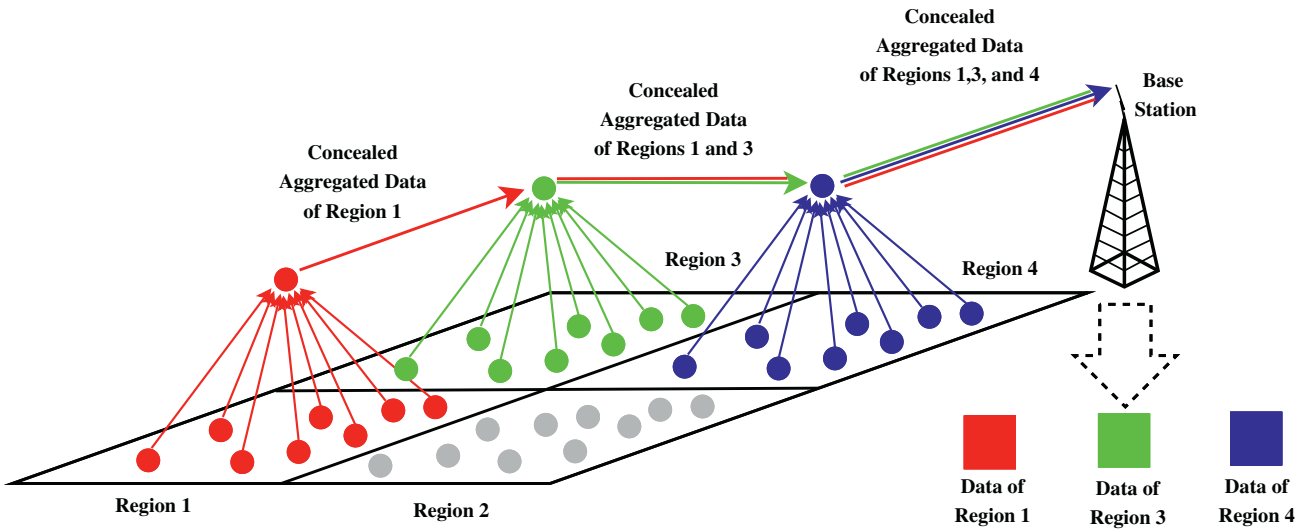


Fig. 1. The motivation behind IPHCDA protocol. A battlefield surveillance network consists of four deployment regions. Each region's data are hierarchically aggregated without violating data confidentiality and integrity. During the decryption of the aggregated data, the base station is able to extract data from each deployment region based on the encryption keys.

Author's personal copy

ysis of IPHCDA is described in Section 5, and performance evaluation is presented in Section 6. Finally, concluding remarks are made in Section 7.

2. Related work

In the wireless sensor network domain, the secure data aggregation has been extensively studied [5–16]. In [5], a security mechanism that detects node misbehaviors, such as dropping or forging messages and transmitting false data, is presented. In [6], random sampling mechanisms and interactive proofs are used to check the correctness of the aggregated data at base station. In [7], sensor nodes first send data aggregators the characteristics of their data to determine which sensor nodes have distinct data; those sensor nodes with distinct data then transmit their encrypted data. In [8], the witness nodes of data aggregators also aggregate data and compute MACs to help verify the correctness of the aggregators' data at the base station. In [9], sensor nodes use the cryptographic algorithms only when cheating activity is detected. The authors of [10] propose that during a normal hop-by-hop aggregation process in a tree topology, more trust is placed on the high-level sensor nodes (i.e., nodes closer to the root) than the low-level nodes. [11] proposes a protocol that makes use of a web of trust to overcome the shortcomings of cryptography-based secure data aggregation solutions.

The schemes proposed in [12] aim to bridge the gap between collaborative data aggregation and data privacy. The authors present two privacy-preserving data aggregation schemes for additive aggregation functions: Cluster-based Private Data Aggregation (CPDA) and Slice-Mix-AggRegaTe (SMART). CPDA leverages the clustering protocol and algebraic properties of polynomials so that the communication overhead is reduced. SMART employs data-slicing techniques and the associative property of addition. In the proposed scheme, each sensor node slices its data into n pieces, and the pieces are then securely distributed to $n-1$ nearest sensor nodes for aggregation. The authors of [13] propose a family of secret perturbation-based schemes that can protect sensor data confidentiality without disrupting additive data aggregation. In the proposed schemes, the base station shares a secret with each sensor node. When a sensor node has a sensory data item to report, it does not report the original data but the sum of the original data and the secret shared with the base station. Compared to existing schemes, the proposed schemes provide confidentiality protection for both raw and aggregated data with lower overhead. In [14], sensor nodes form clusters to perform secret aggregation. To hide the individual sensor readings, the proposed scheme employs twin-keys shared by node pairs within a cluster. The cluster aggregates are sent in clear text to be further aggregated to compute the final aggregate. The proposed scheme ensures that the individual values and the identity of the contributing nodes cannot be derived by any node in the network. The authors of [15] propose several efficient mechanisms for privately querying wireless sensor networks. Two network models are presented. In the first one, access to sensor readings is provided by a single organization. In the second one, any of multiple, mutually distrusting organizations can perform this operation. In [16], PriSense is proposed as a novel solution to privacy-preserving data aggregation in people-centric urban sensing systems. PriSense is based on the concept of data slicing and supports additive and non-additive aggregation functions with accurate aggregation results. Moreover, PriSense provides strong user privacy against a tunable threshold number of colluding users and aggregation servers.

The protocols in [17,18] utilize symmetric and asymmetric privacy homomorphic encryption to allow aggregation of encrypted data. However, in [17], sensor data must be encrypted with a single key to perform concealed data aggregation. Therefore, to hierarchically aggregate data of the whole network, sensor nodes in the network must use a common key for data encryption. Using a single symmetric key is not secure because an adversary can fake the aggregated results through compromising a single sensor node. In addition, symmetric key-based privacy homomorphism is shown to be insecure for chosen plaintext attacks for some specific parameter settings [24]. The scheme proposed in [18] relies on asymmetric key-based privacy homomorphism, but it uses a single public key. Therefore, it is not possible to classify sensor data after it is aggregated. The scheme proposed in [19] allows using different encryption keys in aggregated data. The authors employ an extension of the one-time pad encryption technique using additive operations modulo n. However, several practical issues are not addressed in this paper. First of all, each aggregated data packet is coupled with the list of sensor nodes that failed to contribute to the aggregation, and this fact makes the scheme impractical for large wireless sensor networks. Second, a strong synchronization mechanism must be implemented to perform aggregation correctly. While these protocols provide data confidentiality to data aggregation process, it should be noted that they are not resistant to attacks targeting the data integrity because homomorphic encryptions are malleable by design. IPHCDA, however, integrates homomorphic encryption and MACs to offer data integrity and confidentiality together. In [20], privacy preserving integrity-assured aggregation data aggregation is studied for a single aggregator model. Our work differs from [20] by offering integrity protecting hierarchical concealed data aggregation for the multiple data aggregator model. In addition, our protocol enables the base station to classify aggregated data based on the encryption keys.

3. System model and preliminaries

We consider a large sensor network with densely deployed sensor nodes. Due to the dense deployment, sensor nodes have overlapping sensing regions, and events are detected by multiple sensor nodes. Hence, data aggregation is needed to reduce data transmission. Some sensor nodes are dynamically designated as data aggregators to aggregate data from their neighboring sensor nodes. To balance the energy consumption of sensor nodes, the role of data aggregator is rotated among sensor nodes based on

Author's personal copy

their residual energy levels. Sensor nodes have limited computation and communication capabilities. For example, Mica2 motes [22] have a 4 Mhz 8bit Atmel microprocessor and are equipped with an instruction memory of 128 KB and a RAM of 4 KB. All messages are time-stamped and nonces are used to prevent reply attacks. Sensor nodes encrypt their data prior to data transmission, and data aggregators authenticate their aggregation result by computing the MAC of the aggregated data. Encrypted data are decrypted and validated only at the base station. The base station is interested in the data from a region in the network rather than the data from a single sensor node. Therefore, the network deployment area is divided into several deployment regions as described below.

3.1. Key distribution and network deployment

IPHCDA assumes that the network area is divided into regions and a public/private key pair is assigned to each region. Sensor nodes of a region are given the public key of their respective region. The base station holds the private keys. In addition to the public key, each sensor node i shares a unique MAC key K_m^i with the base station.

To virtually divide the network area into several regions, the sensor network is deployed using a strategy described in [23]. Before the deployment, IPHCDA divides sensor nodes into several groups and assigns a public key to each deployment group. Then, each group is deployed from a certain location over the network area. The network deployment is achieved by dropping the sensor node groups from a plane or a helicopter. Hence, each deployment group covers a part of the network. The idea behind the group based network deployment is that the base station is able to classify the data of a sensor node group based on the public key of that group thereby achieving spatial data gathering.

In the network deployment scenario, we assume that sensor nodes are distributed with a Gaussian (Normal) distribution. The Gaussian distribution allows us to compute the maximum *distance* between two deployment points over the network area. In a Gaussian distribution, the distances between the deployment point of sensor nodes and their final locations are guaranteed to be less than 3σ with probability 0.9987, where σ is the standard deviation of a Gaussian distribution. If each sensor node group covers a circular area with radius 3σ centered at its deployment point, the network area is fully covered. Therefore, to have full coverage of the network area, the distance (d) between two deployment points should not exceed $3\sqrt{2}\sigma$, as shown in Fig. 2.

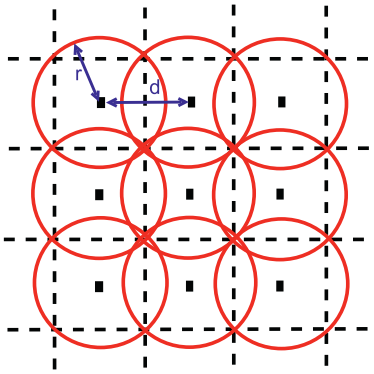


Fig. 2. Determining the positions of the deployment points: Radius (r) of each circle is 3σ by Gaussian distribution; therefore, the maximum distance (d) between any two deployment points cannot exceed $3\sqrt{2}\sigma$ to provide full coverage.

3.2. Privacy homomorphism

A privacy homomorphism is an encryption transformation that allows direct computation on encrypted data. Let *Enc* and *Dec* denote *encryption* and *decryption* processes, respectively. Also, let + denote addition and $\times$ denote multiplication operation over a data set Q. Assume that K_r and K_u are the private and public keys of the base station, respectively. An encryption transformation is accepted to be additively homomorphic if

$$a + b = Dec_{K_r}(Enc_{K_u}(a) + Enc_{K_u}(b)) \quad \text{where} \quad a, b \in Q$$

and it is accepted to be multiplicatively homomorphic if

$$a \times b = Dec_{K_r}(Enc_{K_u}(a) \times Enc_{K_u}(b)) \quad \text{where} \quad a, b \in Q.$$

Because additively and multiplicatively homomorphic cryptographic functions support additive and multiplicative operations on encrypted data, respectively, data aggregators can perform addition- and multiplication-based data aggregation over the encrypted data. Privacy homomorphic encryption can be achieved using symmetric or asymmetric cryptography. Recently, privacy homomorphism based on symmetric key cryptography has been shown to be insecure in chosen plaintext attacks for some specific parameter settings [24]. Therefore, for mission critical networks, asymmetric cryptography-based privacy homomorphism should be used instead of symmetric cryptography-based privacy homomorphism. However, public key-based privacy homomorphism is prohibitively expensive for resource-limited wireless sensor networks.

Realizing that asymmetric cryptography-based privacy homomorphism incurs high computational overhead, the IPHCDA protocol employs the elliptic curve cryptography-based privacy homomorphic encryption scheme proposed in [21]. This scheme allows concealed aggregation of data that are encrypted with different keys. Although the encryption scheme of [21] provides additive and multiplicative homomorphism, IPHCDA protocol only takes advantage of the additive homomorphism property because multiplicative homomorphism is quite expensive. Below, we describe the additive homomorphic encryption process of the IPHCDA protocol using a similar notation to [21].

- *Key generation:* Given a security parameter $\tau \in \mathbb{Z}$ compute $\varphi(\tau)$ to generate the tuple (q_1, q_2, E, n). E is a set of elliptic curve points that form a cyclic group. The set E should be on the order of n where $n = q_1 q_2$. Randomly select two points (u and g) of order n from E. Set $h = u^{q_2}$ where h's order is q_1. Set the public key as $P_u = (n, E, g, h)$ and the private key as $P_r = q_1$.
- *Encryption:* Set an integer T where $T < q_2$, and let the bit length of T be close to the bit length of q_2. The message

Author's personal copy

M space should consist of integers in the set $\{0,1,\ldots,T\}$. To encrypt a message m using public key P_u, pick a random $r \leftarrow \{0,1,\ldots,n-1\}$ and compute the ciphertext $\mathcal{C} = g^m + h^r$ where + is the addition of elliptic curve points and a^b is the scalar multiplication of elliptic curve points a and b. It should be noted that because the encryption process relies on the random number r, the resulting ciphertext is probabilistic; therefore, the scheme is resilient to chosen plaintext attacks [24].

- *Decryption:* To decrypt a ciphertext $\mathcal{C}$ using the private key $P_r = q_1$, observe that $\mathcal{C}^{q_1} = (g^m + h^r)^{q_1} = (g^{q_1})^m$. Let $\hat{g} = g^{q_1}$; then, to recover m, compute the discrete log of C^{q_1} base $\hat{g}$. It should be noted that the message m is between 0 and T; therefore, the decryption operation takes $O(\sqrt{T})$ time using Pollard's lambda method [25].
- *Aggregation:* Two ciphertexts $\mathcal{C}_1 = g^{m_1} + h^{r_1}$ and $\mathcal{C}_2 = g^{m_2} + h^{r_2}$ are aggregated into a ciphertext of $\mathcal{C}'$ as follows:

$$\mathcal{C}' = \mathcal{C}_1 + \mathcal{C}_2 = g^{(m_1+m_2)} + h^{(r_1+r_2)}.$$

For more details, such as proof of homomorphism, interested readers are referred to [21]. Let us offer an example of how this encryption scheme can be employed in wireless sensor networks. To encrypt a message m_i, a sensor node N_i first chooses a random number r_i and computes the ciphertext $\mathcal{C}_i = g^{m_i} + h^{r_i}$ using the public key (n,E,g,h). Similarly, to encrypt a message m_j, a sensor node N_j first selects a random number r_j and computes the ciphertext $\mathcal{C}_j = g^{m_j} + h^{r_j}$ using the public key (n,E,g,h). Assume that a data aggregator aggregates $\mathcal{C}_i$ and $\mathcal{C}_j$ into $\mathcal{C}_{agg}$ and sends it to the base station. Then, the base station computes the aggregated message by calculating the discrete logarithm of $\mathcal{C}_{agg}^{q_1}$ to the base $\hat{g}$ where q_1 is the private key and $\hat{g} = g^{q_1}$.

4. IPHCDA: integrity protecting hierarchical concealed data aggregation

In this section, we first describe the IPHCDA protocol's modified homomorphic encryption scheme that allows aggregation of data of k deployment groups where $k > 1$. Then, we show how IPHCDA provides integrity protection to aggregated data. Finally, we present a concrete example of IPHCDA protocol for a wireless sensor network that consists of two deployment groups.

4.1. Hierarchical concealed data aggregation

In the previous section, we explained the homomorphic encryption scheme of [21]. If the sensor network uses a single public–private key pair, this encryption scheme can be used in the IPHCDA protocol directly. However, the IPHCDA protocol aims to hierarchically aggregate data of multiple sensor node groups (i.e., deployment regions) that use different public–private key pairs. Hence, the homomorphic encryption scheme of [21] should be modified in the following way:

- *Key generation:* Given a security parameter $\tau \in \mathbb{Z}$, compute $\varphi(\tau)$ to generate the tuple $(q_1,q_2,\ldots,q_{k+1},E,n)$. E is a set of elliptic curve points that form a cyclic group. The order of E is n where $n = q_1 q_2 \cdots q_{k+1}$. Next, randomly select $k+1$ points $(u_1,u_2,\ldots,u_{k+1})$ from E where the order of u_i is n for $i = 1$ to $k+1$. Set h as follows:

$$h = u_{k+1}^{\beta} \quad \text{where} \quad \beta = \prod_{i=1}^{k} q_i \quad \text{and} \quad i = 1,\ldots,k.$$

The order of h is q_{k+1}. Now, we need k public keys for k deployment groups; hence, we compute a P value for each deployment group in the following way:

$$P_z = g_z^{\alpha} \quad \text{where} \quad \alpha = \prod_{i=1,i\neq z}^{k+1} q_i \quad \text{and} \quad z = 1,\ldots,k.$$

The public key of the deployment group z is $P_u^z = (n,E,P_z,g,h)$ for $z = 1$ to k, and the private key is $P_r = (q_1,q_2,\ldots,q_{k+1})$.

- *Encryption:* Set $T_z < q_z$, and let the bit length of T_z be approximately close to the bit length of q_z. The message M space of a sensor node that belongs to the deployment group z should consist of integers in the set $\{0,1,\ldots,T_z\}$. To encrypt a message m using public key P_u^z, pick a random $r \leftarrow \{0,1,\ldots,n-1\}$ and compute the ciphertext $\mathcal{C} = P_z^m + h^r$ where + is the addition of elliptic curve points and a^b is the scalar multiplication of elliptic curve points a and b.
- *Aggregation:* Let $\sum m_i$ denote that the aggregated message of the ith deployment group; consequently, k ciphertexts $\mathcal{C}_z = P_z^{m_z} + h^{r_z}$ for $z = 1$ to k are aggregated into a ciphertext of $\mathcal{C}'$ as follows:

$$\mathcal{C}' = \sum_{i=1}^{k} P_i^{\sum m_i} + h^{\sum r_i}.$$

- *Decryption:* During the decryption, the base station is able to separately decrypt the data of each deployment group z from the aggregated ciphertext $\mathcal{C}'$. Let $\widehat{g_z}$ be

$$\widehat{g_z} = g_z^{\alpha} \quad \text{where} \quad \alpha = \prod_{i=1,i\neq z}^{k+1} q_i \quad \text{and} \quad z = 1,\ldots,k,$$

then the base station can recover the aggregated data $\sum_{i=z} m_i$ of each deployment group z by computing the discrete log of $(\mathcal{C}')^{\alpha}$ base $\hat{g}_z$. Therefore, the decrypted data of deployment group z is

$$\sum_{i=z} m_i = log_{\widehat{g_i}}(\mathcal{C}')^{\alpha} \quad \text{where} \quad \widehat{g_i} = g_z^{\alpha},$$

$$\alpha = \prod_{i=1,i\neq z}^{k+1} q_i, \quad \text{and} \quad z = 1,\ldots,k.$$

4.2. Integrity protection of aggregated data

By falsifying the aggregated data, an adversary can manipulate the measurements of a large portion of the wireless sensor network. Therefore, in addition to data privacy, IPHCDA aims to protect the integrity of the aggregated data in the following way. Each sensor node encrypts its data using the public key of the region in which it resides and sends it to the data aggregator of the region. The data

Author's personal copy

aggregator DA_i receives encrypted data from the sensor nodes in the ith region and aggregates ($\mathcal{C}'_i$) as described above. DA_i also computes the MAC of the aggregated data $\mathcal{C}'_i$ ($MAC(\mathcal{C}'_i)$) using the unique symmetric key that it shares with the base station. All DA_is, where $1 < i \leqslant k$ and k is the number of deployment regions; forward their aggregated data and MAC pair ($\mathcal{C}'_i, MAC(\mathcal{C}'_i)$) to the base station over the aggregation tree as shown in Fig. 1. On the aggregation tree, when a data aggregator DA_t receives $\mathcal{C}'_i$, $MAC(\mathcal{C}'_i)$ and $\mathcal{C}'_j$, $MAC\left(\mathcal{C}'_j\right)$ from data aggregators DA_i and DA_j, it aggregates $\mathcal{C}'_i$ and $\mathcal{C}'_j$ and XORs $MAC(\mathcal{C}'_i)$ and $MAC\left(\mathcal{C}'_j\right)$. Therefore, the base station receives the aggregation of all $\mathcal{C}'_i$s and a single $MAC(\mathcal{C}')$, which is an XOR of all $MAC(\mathcal{C}'_i)$s where $1 < i \leqslant k$ and k is the number of deployment regions. When the base station receives the aggregated data, it obtains the data of each DA_i and computes $MAC(\mathcal{C}^*_i)$s for each DA_i. To validate the aggregated data, the base station computes $MAC(\mathcal{C}^*)$ by XORing all $MAC(\mathcal{C}^*_i)$ and compares $MAC(\mathcal{C}^*)$ and $MAC(\mathcal{C}')$. It should be noted that IPHCDA can provide data integrity protection to individual sensor readings as well. However, in that case, the base station would need the list of all sensor nodes that contributed to the aggregated data, thereby incurring too much communication overhead.

4.3. An illustrative example

In this section, we present an example to show how the IPHCDA protocol achieves integrity protection and hierarchical concealed aggregation of multiple deployment groups' data. For the sake of simplicity, let us assume that the network consists of four deployment groups and only two groups send data to the base station. Group 1 has the public key $P^1_u = (n, E, P_1, g, h)$, and group 2 has the public key $P^2_u = (n, E, P_2, g, h)$. As shown in Fig. 3, each group has two sensor nodes and a data aggregator. Group 1 has sensor nodes SN^A_1, SN^B_1 and data aggregator DA_1. Similarly, group 2 has sensor nodes SN^A_2, SN^B_2 and data aggregator DA_2. Furthermore, assume that DA_1 and DA_2 share a symmetric keys K^s_1 and K^s_2 with the base station, respectively. There is also another data aggregator DA_3 in group 3 that aggregates and transmits data from DA_1 and DA_2 to the base station. To keep the example simple, the order of P_1, P_2, and h are set to small numbers in the following way:

- Order of P_1 and value of q_1 is 11
- Order of P_2 and value of q_2 is 13
- Order of h and value of q_3 is 17
- Order of $n = q_1q_2q_3$ is 2431

Sensor nodes in groups 1 and 2 encrypt and send their data in the following way (note that r values are randomly generated by sensor nodes):

- SN^A_1 generates message $M^A_1 = 1$ and encrypts it as $\mathcal{C}^A_1 = P^1_1 + h^4$
- SN^B_1 generates message $M^B_1 = 3$ and encrypts it as $\mathcal{C}^B_1 = P^3_1 + h^6$
- SN^A_2 generates message $M^A_2 = 4$ and encrypts it as $\mathcal{C}^A_2 = P^4_2 + h^2$
- SN^B_2 generates message $M^B_2 = 2$ and encrypts it as $\mathcal{C}^B_2 = P^2_2 + h^7$

Sensor nodes send their messages to data aggregators. Data aggregator DA_1 aggregates $\mathcal{C}^A_1$ and $\mathcal{C}^B_1$ as $\mathcal{C}_1 = P^4_1 + h^{10}$ and computes $MAC(\mathcal{C}_1)$ using K^s_1. Similarly, data aggregator DA_2 aggregates $\mathcal{C}^A_2$ and $\mathcal{C}^B_2$ as $\mathcal{C}_2 = P^6_2 + h^9$ and computes $MAC(\mathcal{C}_2)$ using K^s_2. DA_1 and DA_2 forward their aggregated Data, and MAC pairs to DA_3. DA_3 aggregates $\mathcal{C}_1$ and $\mathcal{C}_2$ as $\mathcal{C} = P^4_1 + P^6_2 + h^{19}$. Because the order of h is 17, $h^{17} = \infty$, and ∞ is the additive unit element in elliptic curve arithmetic, we can write $\mathcal{C} = P^4_1 + P^6_2 + h^2$. In addition, DA_3 performs the following XOR operation $MAC(\mathcal{C}) = MAC(\mathcal{C}_1) \oplus MAC(\mathcal{C}_2)$ and sends $\mathcal{C}, MAC(\mathcal{C})$ to the base station.

To obtain the data of group 1, the base station first computes $\mathcal{C}^{q_2q_3} = (P^4_1 + P^6_2 + h^2)^{221}$. Because a^b donates scalar multiplication of elliptic curve points, $\mathcal{C}^{q_2q_3}$ equals $P^{884}_1 + P^{1326}_2 + h^{442}$. Note that $h^{17} = \infty$, $P^{11}_1 = \infty$, and $P^{13}_2 = \infty$. Thus, using elliptic curve arithmetic, we can write $\mathcal{C}^{q_2q_3} = P^4_1$. Finally, the base station obtains the data of group 1 by computing the discrete logarithm of $\mathcal{C}^{q_2q_3} = P^4_1$ to the base $\widehat{g_1}$ where $\widehat{g_1} = g^{q_2q_3}_1$. Once the base station obtains data from DA_1 and DA_2, it computes their MACs $MAC(\mathcal{C}^*_1)$ and $MAC(\mathcal{C}^*_2)$, respectively. Then, the base station XORs the computed MACs as $MAC(\mathcal{C}^*) = MAC(\mathcal{C}^*_1) \oplus MAC(\mathcal{C}^*_2)$. If the resulting $MAC(\mathcal{C}^*)$ and the original $MAC(\mathcal{C})$ matches, then the base station validates the integrity of the aggregated data.

5. Security analysis

In this section, we show IPHCDA's resistance to attacks that are described in [26], where authors summarize all possible attacks against any concealed data aggregation scheme. Due to space constraints, the details of the secu-

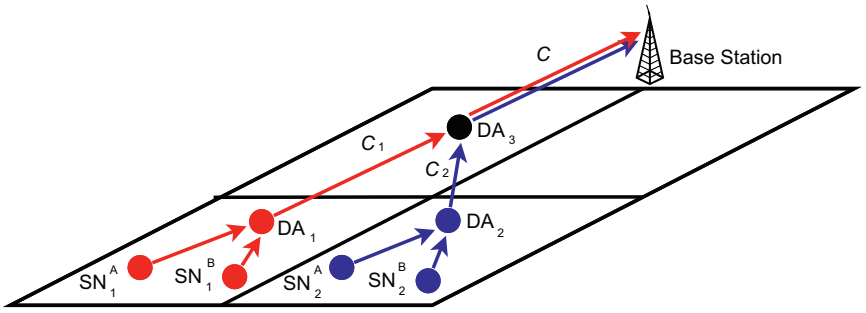


Fig. 3. Example of IPHCDA protocol. For the sake of simplicity, only two groups send data to the base station. DA_3 aggregates data of DA_1 and DA_2.

Author's personal copy

rity of encryption algorithm are not included here, but interested readers are referred to [21].

5.1. Ciphertext analysis

In a wireless environment, the most basic attack is the analysis of encrypted packets in which the adversary tries to obtain information only by interpreting ciphertexts. Because IPHCDA's elliptic curve cryptography-based encryption depends on the factorization of large integers, it is robust to ciphertext analysis.

5.2. Known plaintext attack

In this kind of attack, the adversary wants to determine secret information with a known plaintext and corresponding ciphertext. At the end of this attack, the adversary is able to either reveal the secret key or gather some information that he/she can use to deduce malicious ciphertexts. Because IPHCDA's encryption process relies on random numbers, the resulting ciphertext is probabilistic. Therefore, the IPHCDA scheme is robust to known plaintext attacks.

5.3. Replay attacks

In a replay attack, previously sent valid packets are transmitted later to achieve a malicious effect. The encryption scheme of IPHCDA does not offer any protection against replay attacks. However, the IPHCDA protocol prevents this attack by time stamping all data packets.

5.4. Malleability

Adversaries may want to exhaust sensor nodes by sending them randomly generated meaningless ciphertexts that are syntactically correct. This attack only slows down the operation of the network. Adversaries can also change the content of the legitimate ciphertexts in this attack, thus falsifying the network data. Hence, concealed data aggregation schemes should not let adversaries be able to change the contents of ciphertexts. IPHCDA employs MACs to prevent attacks targeting data integrity. Therefore, an attacker can successfully change a ciphertext if and only if he/she can forge a valid MAC tag for the ciphertext. Considering that IPHCDA uses an unforgeable MAC protocol, such as HMAC [27], attackers cannot successfully forge aggregated data packets.

5.5. Unauthorized aggregation

In a homomorphic data aggregation protocol, the encryption key and the MAC key must be known by the adversary to achieve unauthorized aggregation. Hence, in IPHCDA, an adversary must compromise at least one data aggregator so that he/she can obtain the encryption and MAC keys and successfully perform unauthorized aggregation. The adversary cannot achieve unauthorized aggregation by compromising a regular sensor node because there is only one data aggregator in each deployment region, and the base station would notice the compromised node. In some of the previous privacy homomorphism-based concealed data aggregation schemes, adversaries are able to perform unauthorized aggregation without any additional information [17].

5.6. Forge packets

In any public key-based encryption scheme, there is no need to forge data packets because it easy to generate proper ciphertexts using the public key. Therefore, to protect data integrity, public key-based schemes must be used with additional protection. As explained in Section 5.4, IPHCDA use an unforgeable MAC protocol (such as HMAC [27]) to prevent aggregated data from forgery attacks.

5.7. Physical attacks

Node compromise attacks target sensor node hardware to execute or support an attack. In IPHCDA, if an adversary compromises a data aggregator, it can perform unauthorized aggregation and send false aggregation results to the base station. However, due to the asymmetric public key approach, an adversary cannot gain any additional information regarding the data aggregated. Hence, in IPHCDA, physical attacks can affect the data integrity but not the data confidentiality.

6. Performance evaluation

From the resource consumption point of view, symmetric key-based homomorphic encryption schemes are more efficient than public key-based ones. However, the security of public key-based homomorphic encryption schemes has been shown to be stronger than symmetric key-based ones. Therefore, rather than symmetric key-based schemes, the performance of IPHCDA is compared with two other public key-based homomorphic encryption schemes: namely EC-OU [26] and EC-EG [26].We evaluated the computational overhead, the communication load, and the accuracy of IPHCDA, EC-OU, and EC-EG schemes.

6.1. Computational overhead

Our computational overhead evaluation includes encryption, decryption, and addition operations. We followed the performance evaluation methodology defined in [26] where all computations are converted to and measured in terms the number of base units (1024-bit modular multiplications). It should be noted that IPHCDA, EC-OU, and EC-EG schemes are built upon different mathematical foundations. Hence, a base unit of measurement must be used to achieve a fair comparison. Similar to [26], the measurement unit chosen in our computational overhead comparison is the 1024-bit modular multiplication. In addition, we assume that the number of deployment regions in IPHCDA is 4. For an elliptic curve computation over a finite field F_p, the number of $|p|$-bit modular multiplications is first counted, and it is then converted to 1024-bit modular multiplications. The results are presented in Fig. 4 (a), (b), and (c). The results show that due to its smaller modulus

Author's personal copy

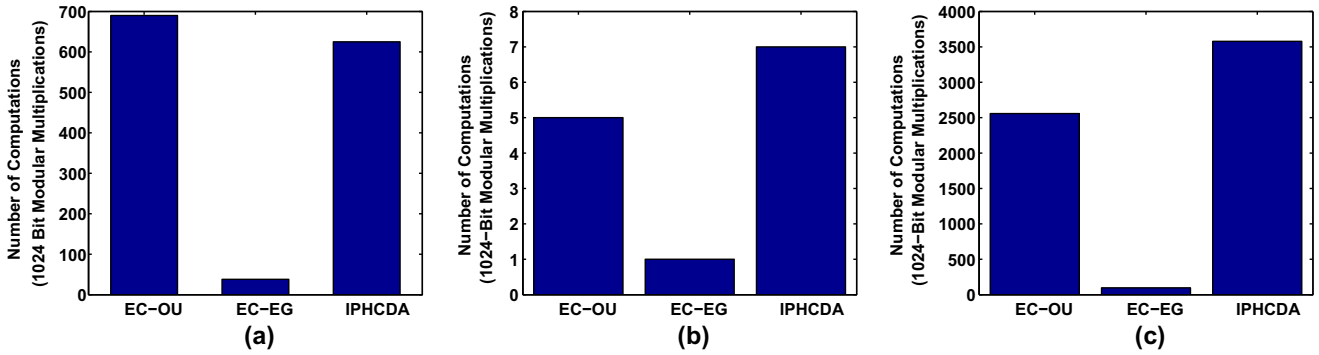


Fig. 4. Performance comparison of EC-OU, EC-EG, and IPHCDA. (a) Encryption cost. (b) Addition cost. (c) Decryption cost.

operations in computation [26], EC-EG has the lowest computational overhead, and the performance of IPHCDA and EC-OU are close to each other. However, neither EC-OU nor EC-EG can support hierarchical concealed data aggregation if different keys are used in the network.

6.2. Communication cost

To evaluate the communication cost, we first compute the ciphertext size of each scheme. For IPHCDA, the number of deployment regions affects the size of $|n|$ (in bits), which is on the order of elliptic curve points. Note that n is composed of $(k+1)$ q-bit prime numbers and $n = q_1 \times q_2 \times q_3 \times \cdots \times q_{k+1}$ where k is the number of deployment regions. Also note that the encryption process involves the addition of elliptic curve points that are on the order of n. It follows that IPHCDA's ciphertext size is $(k+1) \times |q|$. Hence, in IPHCDA scheme, the ciphertext size increases as the number of deployment regions increases. In [26], it is shown that EC-OU's ciphertext size is $3 \times |q| + 2$ ($|q|$ = 341-bit) and EC-EG's ciphertext size is $|q| + 2$ ($|q|$ = 163-bit). Similar to EC-OU, we used 341-bit qs for IPHCDA ($|q|$ = 341-bit) in our evaluation. The comparison of ciphertext sizes are depicted in Fig. 5. The results show that if the number of deployment regions is more than 3, EC-EG and EC-OU outperform IPHCDA. However, neither EC-OU nor EC-EG can support IPHCDA's unique properties, such as hierarchical data aggregation and the separation of aggregated data at the base station. It can also be observed from the Fig. 5 that there is a tradeoff for IPHCDA between security, number of deployment regions and ciphertext size. To achieve stronger security (i.e., longer $|q|$) and low communication overhead, we must reduce the number of deployment regions. Similarly, to have a large number of deployment regions and low communication overhead, we must reduce the size of $|q|$, thereby reducing the security level. Hence, the selection of $|q|$ and the number of deployment regions depends on the level of security required by the application.

6.3. Simulation results

To show the benefit of IPHCDA's hierarchical data aggregation property, IPHCDA, EC-OU, and EC-EG schemes are simulated using TinyOS 2.0 Simulator (TOSSIM) [28]. In three simulation scenarios, 120, 150 and 200 sensor nodes

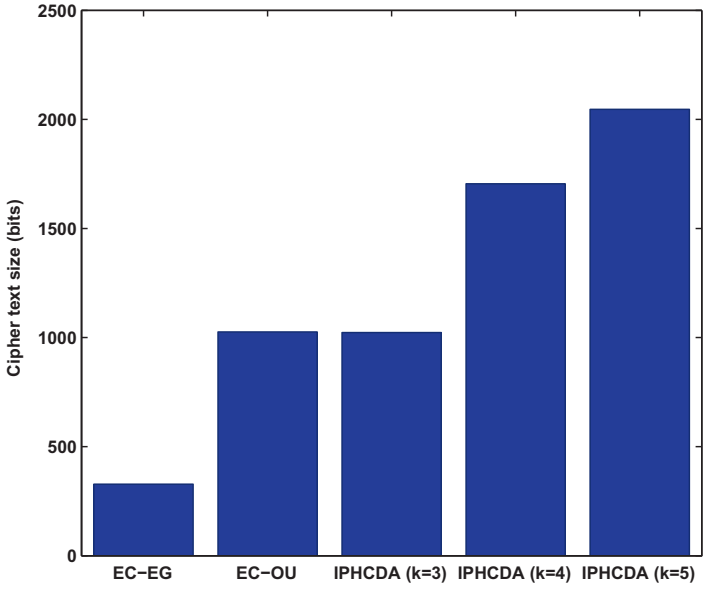


Fig. 5. The ciphertext sizes of IPHCDA, EC-OU, and EC-EG. For IPHCDA, size of $|q|$ is selected as 341-bit.

Author's personal copy

are placed in uniformly distributed random locations within a 160 m × 160 m rectangular area where the base station is located on one corner. The simulation area is vertically divided into 3, 4, and 5 deployment regions where each region has a data aggregator. A static beacon-based ad hoc routing algorithm is used to transfer data from data aggregators to the base station. Due to the poor radio conditions of wireless sensor networks, a retransmission mechanism is implemented, and the retransmission limit is set to 5. The default bit error rate is 5%, which can be accepted as a poor radio condition. The medium access scheme is set as CSMA using the default TinyOS 2.0 CC2420 stack, which has 4 bits per symbol and 64 K symbols per second, for 256 Kbps. Due to the high bit error rates and limitations of TinyOS protocol stack, the maximum MAC layer frame size is set as 39-byte, as depicted in Fig. 6. In the previous section, it is shown that IPHCDA's aggregated ciphertext can be up to 300-bytes. The Fig. 6 also shows how ciphertexts are put into the application layer packets. Clearly, a single MAC layer frame size is not sufficient for this application data packet size. Moreover, it is not practical to send such large data packets over a wireless medium due to high bit error rates. Therefore, in IPHCDA protocol, before data transmission, application layer data packets (both chiphertext and message authentication code) are divided into smaller blocks that can fit into medium access layer frames. For example, if $k = 5$ and ciphertext size is 300, each application layer packet of IPHCDA is divided into 10 blocks where each block is less than 30 bytes and assigned a sequence number. Upon receiving these small data blocks, the base station reconstructs the application data packet using the assigned sequence numbers.

To evaluate the communication overhead, sensor nodes are deployed in a 160 m × 160 m network area in three different simulation scenarios: 120, 150 and 200. The total data transmission amount in the network is measured for IPHCDA, EC-OU, and EC-EG schemes. The network area size is the same for all of the scenarios; hence, the node density of the network is increased as the number of sensor nodes increases. Each simulation run lasts for 600 s, and each result is averaged over 10 simulation runs. In case of IPHCDA, data aggregators are allowed to aggregate other data aggregators' forwarded data, thereby achieving hierarchical data aggregation. In EC-EG and EC-OU cases, data are aggregated only in the deployment region without hierarchical data aggregation. In the first scenario, 70 sensor nodes out of 120 nodes were randomly chosen to send 100 data packets to their data aggregators. In the second scenario, 150 nodes were used, of which 100 sensor nodes were randomly selected to send 100 data packets to their data aggregators. Similarly, in the third scenario, 125 sensor nodes were randomly chosen from 200 sensor nodes to send 100 data packets to their data aggregators. The results are presented in Fig. 7. The simulation results show that although ciphertext size of IPHCDA is greater than EC-OU and EC-EG, hierarchical data aggregation reduces the total amount of data transmission in the network. We can see from Fig. 7 (a) that when the number of deployment regions is 3 or 4 IPHCDA should be used instead of EC-OU or EC-EG. If the number of deployment groups is greater than 4, then IPHCDA can be used only in cases in which the base station needs to analyze each deployment region's data separately. However, the results also show that increasing the number of data-transmitting nodes increases the data aggregation efficiency of IPHCDA compared to EC-OU and EC-EG. The reason behind this improvement can be explained in the following way. As the density of the network increases, each data aggregator is able to aggregate data of more sensor nodes, thereby improving the efficiency of hierarchical data aggregation. Therefore, the decision of employing IPHCDA in a network depends on the network size, the number of deployment regions of the network, and the base station needs.

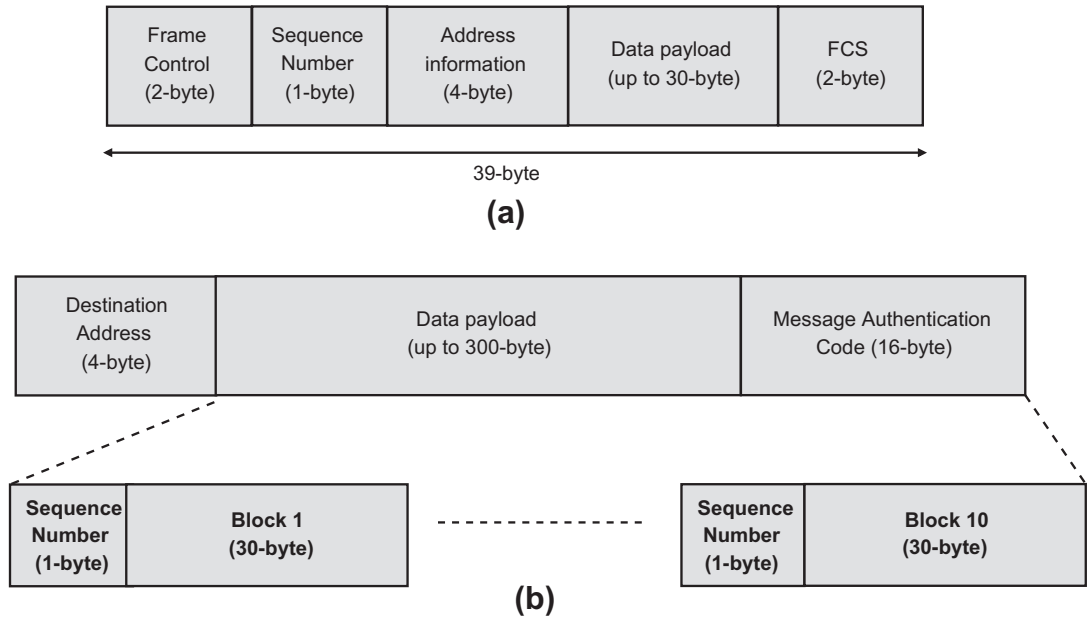


Fig. 6. (a) The medium access layer frame format of IPHCDA. (b) The application packet format of IPHCDA ($k = 5$). To comply with the MAC layer, the packet is divided into 10 packets. Each packet is less than 30 bytes and assigned a sequence number.

Author's personal copy

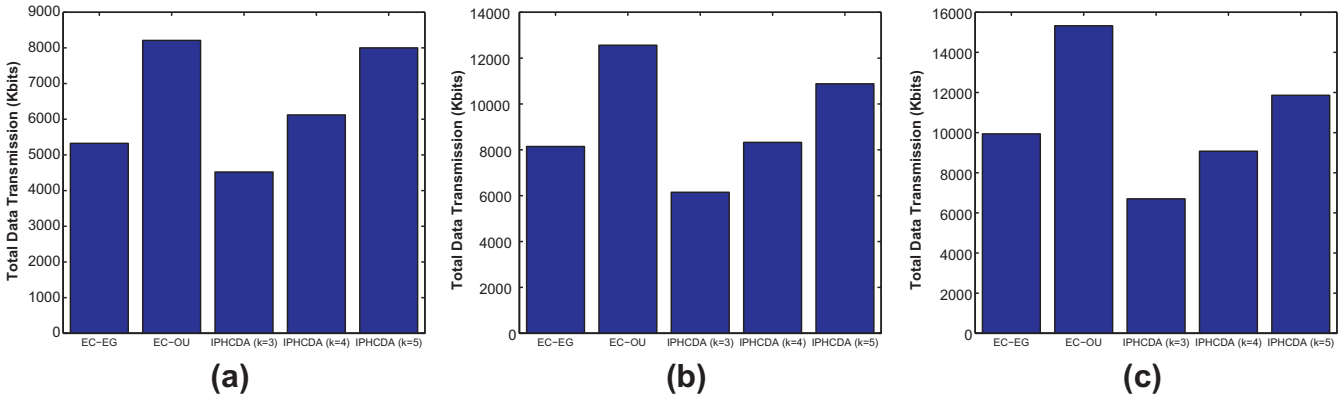


Fig. 7. The total amount of data transmission in the network for different network sizes: (a) 120 sensor nodes, (b) 150 sensor nodes, (c) 200 sensor nodes.

Another factor that affects the total data transmission amount in the network is the bit error rate. In [29], a real-world wireless sensor network experimental study shows that bit errors occur in bursts and there are usually 200 or more consecutive error-free bits between two consecutive burst errors. We followed the same loss model in [29] and evaluated the effect of the bit error rate on the total data transmission amount of the network using IPHCDA, EC-OU and EC-EG schemes. We used a network of 120 sensor nodes and varied the bit error rate from 3% to 10%. The results are presented in Fig. 8, which shows that increasing the bit error rate also increases the amount of data transmission in the network. This is due to the employment of the retransmission mechanism under a high number of frame errors.

6.4. Accuracy

Traditional data aggregation algorithms, such as compression-based data aggregation, may result in alterations in collected data. Therefore, preserving data accuracy is an important issue for traditional data aggregation schemes. IPHCDA, EC-OU, and EC-EG, however, provide additive homomorphic capabilities through the summation of ciphertexts, and they do not change data during data aggregation. Therefore, these additive homomorphic schemes preserve data accuracy after data aggregation. For these schemes, the only factors that affect the accuracy of aggregated data are lost/delayed data packets and computational errors. In the ideal case, when there is no data loss or computational error in the network, all of these schemes achieve 100% data accuracy. However, due to noisy communication channels, delays, and collisions, packet losses frequently occur in wireless sensor networks. Hence, the data accuracy is negatively affected. We evaluated the data accuracy of IPHCDA, EC-OU, and EC-EG schemes. Following the metric proposed in [12], the data accuracy is defined as the ratio between the real sum of raw sensor data and the aggregated sum by the data aggregation scheme in use. The accuracy value of 1 represents the perfect case. In the simulation, 120 sensor nodes and 5% it error rate are used. The accuracy of IPHCDA, EC-OU, and EC-EG with respect to different time intervals between packet transmissions is measured. The results are presented in Fig. 9 (a). The results show that IPHCDA, EC-OU, and EC-EG perform equally in terms of aggregation

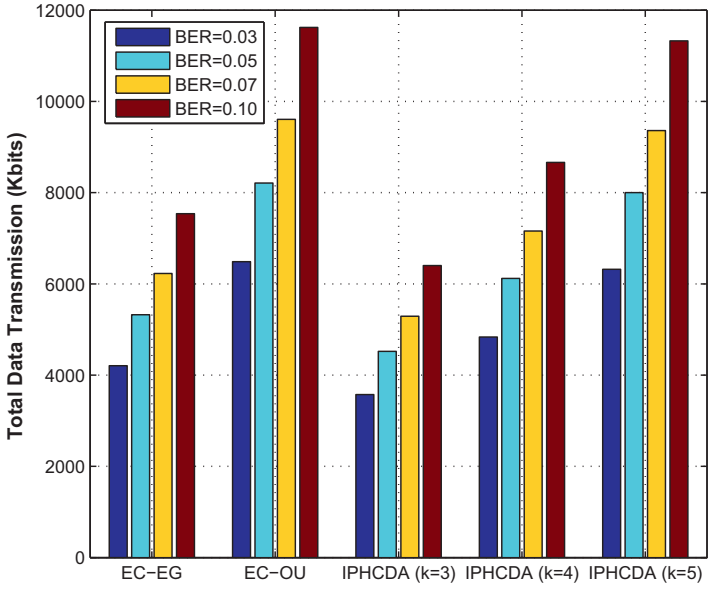


Fig. 8. The effect of BER on the total amount of data transmission in the network of 120 sensor nodes.

Author's personal copy

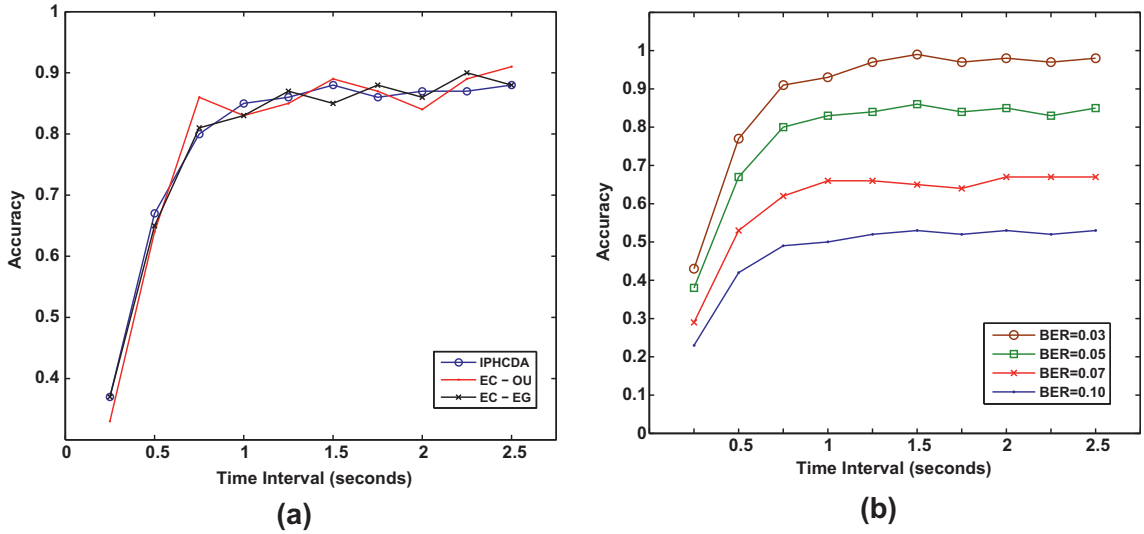


Fig. 9. (a) The accuracy of IPHCDA, EC-OU, and EC-EG with respect to different time intervals between packet transmissions. (b) The effect of BER on data accuracy of IPHCDA.

accuracy. It can also be seen from the results that the accuracy increases as the time intervals between packet transmissions increases. This result is due to the reduction in data collusions and congestion in data aggregators. We have also evaluated the effect of bit error rate on the data aggregation accuracy. The results are presented in Fig. 9 (b). Using 120 sensor nodes, the bit error rate varies from 3% to 10%. The results indicate that increasing the bit error rate reduces the data aggregation accuracy of all data aggregation schemes. This result is due to the fact that higher bit error rates result in higher packet losses.

7. Conclusion

This paper presented the Integrity Protecting Hierarchical Concealed Data Aggregation (IPHCDA) protocol. The proposed scheme is based on a homomorphic encryption algorithm, and it allows the aggregation of data packets that are encrypted with different keys. During the decryption of the aggregated data, the base station is able to classify the encrypted and aggregated data based on the encryption keys. This property is particularly useful in applications where the base station needs to obtain data from a certain part of the network. In addition, the proposed protocol provides integrity protection to aggregated data. To the best of our knowledge, IPHCDA is the first data aggregation scheme that can provide both data confidentiality and integrity in a multi-data aggregator sensor network model. Simulation results show IPHCDA's applicability to wireless sensor networks and that its data aggregation efficiency are better than other privacy homomorphic data aggregation schemes.

Acknowledgement

Prof. Yang Xiao's work was supported in part by the National Science Foundation (NSF) under grants CCF-0829827, CNS-0716211, and CNS-0737325.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (8) (2002) 102–114.
[2] E. Shi, A. Perrig, Designing secure sensor networks, Wireless Communications Magazine 11 (6) (2004) 38–43.
[3] K. Akkaya, M. Demirbas, R.S. Aygun, The impact of data aggregation on the performance of wireless sensor networks, Wiley Wireless Communications and Mobile Computing (WCMC) Journal 8 (2008) 171–193.
[4] S. Ozdemir, Y. Xiao, Secure data aggregation in wireless sensor networks: a comprehensive overview, Computer Networks 53 (12) (2009) 2022–2037.
[5] L. Hu, D. Evans, Secure aggregation for wireless networks, in: Proceedings of Workshop on Security and Assurance in Ad hoc Networks, 2003, pp. 384–392.
[6] B. Przydatek, D. Song, A. Perrig, SIA: secure information aggregation in sensor networks, in: Proceedings of SenSys'03, 2003, pp. 255–265.
[7] H. Çam, S. Ozdemir, P. Nair, D. Muthuavinashiappan, H.O. Sanli, Energy-efficient and secure pattern based data aggregation for wireless sensor networks, Computer Communications 29 (4) (2006) 446–455.
[8] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A witness-based approach for data fusion assurance in wireless sensor networks, in: Proceedings of GLOBECOM'03, 2003, pp. 1435–1439.
[9] K. Wu, D. Dreef, B. Sun, Y. Xiao, Secure data aggregation without persistent cryptographic operations in wireless sensor networks, Ad Hoc Networks 5 (1) (2007) 100–111.
[10] Y. Yang, X. Wang, S. Zhu, G. Cao, SDAP: a secure hop-by-hop data aggregation protocol for sensor networks, ACM Transactions on Information Systems and Security 11 (4) (2008) 1–43.
[11] S. Ozdemir, Functional reputation based reliable data aggregation and transmission for wireless sensor networks, Computer Communications 31 (17) (2008) 3941–3953.
[12] W.B. He, X. Liu, H. Nguyen, K. Nahrstedt, T. Abdelzaher, PDA: privacy-preserving data aggregation in wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2007, pp. 2045–2053.
[13] T. Feng, C. Wang, W. Zhang, L. Ruan, Confidentiality protection for distributed sensor data aggregation, in: Proceedings of IEEE INFOCOM, 2008, pp.56–60.
[14] M. Conti, L. Zhang, S. Roy, R. Di Pietro, S. Jajodia, L.V. Mancini, Privacy-preserving robust data aggregation in wireless sensor networks, Security and Communication Networks (Wiley) 2 (2009) 195–213.
[15] B. Carbunar, Y. Yu, W. Shi, M. Pearce, V. Vasudevan, Query privacy in wireless sensor networks, ACM Transactions on Sensor Networks 6 (2) (2010).
[16] J. Shi, R. Zhang, Y. Liu, Y. Zhang, Prisense: privacy-preserving data aggregation in people-centric urban sensing systems, in: Proceedings of IEEE INFOCOM, 2010, pp.1–9.

Author's personal copy

[17] D. Westhoff, J. Girao, M. Acharya, Concealed data aggregation for reverse multicast traffic in sensor networks: encryption, key distribution and routing adaptation, IEEE Transactions on Mobile Computing 5 (10) (2006) 1417–1431.
[18] S. Ozdemir, Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism, in: Proceedings of ICPS'07: IEEE International Conference on Pervasive Services, 2007, pp. 165–168.
[19] C. Castelluccia, E. Mykletun, G. Tsudik, Efficient aggregation of encrypted data in wireless sensor networks, in: Proceedings of Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005, pp.109–117.
[20] G. Taban, V.D. Gligor, Privacy-preserving integrity-assured data aggregation in sensor networks, in: IEEE International Conference on Computational Science and Engineering, vol. 3, 2009, pp. 168–175.
[21] D. Boneh, Eu-Jin God, K. Nissim, Evaluating 2-DNF formulas on cipertexts, in: Proceedings of Theory of Cryptography Conference, LNCS, vol. 3374, 2005, pp. 325–321.
[22] Crossbow Technologies Inc., <http://www.xbow.com>.
[23] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A key predistribution scheme for sensor networks using deployment knowledge, IEEE Transactions on Dependable and Secure Computing 03 (1) (2006) 62–77.
[24] D. Wagner, Cryptanalysis of an algebraic privacy homomorphis, in: Proceedings of Sixth Information Security Conference, 2003, pp. 234–239.
[25] A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997. pp. 128.
[26] S. Peter, D. Westhoff, C. Castelluccia, A survey on the encryption of convergecast-traffic with in-network processing, IEEE Transactions on Dependable and Secure Computing 99, vol. 2, 2010.
[27] M. Bellare, R. Canetti, H. Krawczyk, Keying hash functions for message authentication, in: Crypto 1996, 1996, pp. 1–15.
[28] TinyOS Simulator, <http://www.tinyos.net>, 2010.
[29] R.K. Ganti, P. Jayachandran, H.Luo, T.F. Abdelzaher, Datalink streaming in wireless sensor networks, in: Proceedings of SenSys'06, 2006, pp. 209–222.

Suat Ozdemir has been with the Computer Engineering Department at Gazi University, Ankara, Turkey since March 2007. He received his MSc degree in Computer Science from Syracuse University and PhD degree in Computer Science from Arizona State University. Dr. Ozdemir's research areas mainly include sensor networks, wireless networks, network security, and data mining. Ozdemir is a member of IEEE and currently serving as editor/TPC member/reviewer for various leading IEEE and ACM journals and conferences.

Yang Xiao worked in industry as a MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement work before he joined Department of Computer Science at The University of Memphis in 2002. He is currently with Department of Computer Science at The University of Alabama. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He is an IEEE Senior Member. He is a member of American Telemedicine Association. He currently serves as Editor-in-Chief for International Journal of Security and Networks (IJSN), International Journal of Sensor Networks (IJSNet), and International Journal of Telemedicine and Applications (IJTA). He serves as a referee/reviewer for many funding agencies, as well as a panelist for NSF and a member of Canada Foundation for Innovation (CFI)'s Telecommunications expert committee. He serves on TPC for more than 100 conferences such as INFOCOM, ICDCS, MOBIHOC, ICC, GLOBECOM, WCNC, etc. He serves as an associate editor for several journals, e.g., IEEE Transactions on Vehicular Technology. His research areas are security, telemedicine, sensor networks, and wireless networks. He has published more than 300 papers in major journals, refereed conference proceedings, book chapters related to these research areas. Dr. Xiao's research has been supported by the US National Science Foundation (NSF), U.S. Army Research, Fleet & Industrial Supply Center San Diego (FISCSD), and The University of Alabama's Research Grants Committee.